ITERATION 7

OUR CLIENT:

EPI-USE

groupelephant.com
BEYOND CORPORATE PURPOSE

TEAM 35

CANVAS

NGO PROJECT PORTFOLIO MANAGEMENT SYSTEM

This document defines the logical narratives of all the use cases identified for the proposed Integrated NGO Management System for EPI-USE and groupelephant.com.



Amahle Khoza

u22539702@tuks.co.za

+27 71 023 1872



Stephan Bredenhann

u23667436@tuks.co.za

+27 71 566 0997



Christopher Bailey

u24661092@tuks.co.za

+27 71 365 9963



Nicholas Bailey

u24671691@tuks.co.za

+27 71 610 6952



Alexander Steven

u22496786@tuks.co.za

+27 72 370 0225

Table of Contents

Table of Tables 6
1. Document Introduction 9
2. Logical Use Case Narratives 10
2.1 User Subsystem Use Case Narratives 10
2.1.1 User Login 10
2.1.2 User Logout 13
2.1.3 Reset Password 15
2.1.4 Create User 18
2.1.5 Update User 22
2.1.6 View User 25
2.1.7 Reactivate User 27
2.1.8 Deactivate User 29
2.1.9 View FAQ 31
2.2 Admin Subsystem Use Case Narratives 33
2.2.1 Create User Profile 33
2.2.2 Update User Profile 35
2.2.3 View User Profile 37
2.2.4 Deactivate User Profile 39
2.2.5 Add FAQ 41
2.2.6 Update FAQ 43
2.2.7 Remove FAQ 45
2.2.8 Add FAQ Type 47
2.2.9 Update FAQ Type 49
2.2.10 Remove FAQ Type 51
2.2.11 Create Role 53
2.2.12 View Role 55
2.2.13 Update Role 57
2.2.14 Reactivate Role 59
2.2.15 Deactivate Role 61
2.2.16 Update Role Permissions 63
2.2.17 View Role Permission 65
2.3 Inventory Subsystem Use Case Narratives 67
2.3.1 Create Procurement 67
2.3.2 View Procurement 70
2.3.3 View Inventory 72
2.3.4 Write-Off Inventory 75
2.3.5 Allocate Inventory 77
2.3.6 Create Inventory Item Type 79
2.3.7 View Inventory Item Types 82
2.3.8 Update Inventory Item Type 84
2.3.9 Deactivate Inventory Item Type 87

2.3.10 Reactivate Inventory Item Type 89
2.4 Group Subsystem Use Case Narratives 91
2.4.1 Create Group 91
2.4.2 View Groups 94
2.4.3 Update Group 97
2.4.4 Deactivate Group 100
2.4.5 Create Skill 102
2.4.6 View Skills 105
2.4.7 View Skill 107
2.4.8 Update Skill 109
2.4.9 Deactivate Skill 111
2.5 Warehouse Subsystem Use Case Narratives 113
2.5.1 Create Warehouse 113
2.5.2 View Warehouse 115
2.5.3 Update Warehouse 117
2.5.4 Reactivate Warehouse 119
2.5.5 Deactivate Warehouse 121
2.5.6 Inventory Stock Take 123
2.6 Vendor Subsystem Use Case Narratives 126
2.6.1 Create Vendor 126
2.6.2 View Vendor 129
2.6.3 View Vendors 131
2.6.4 Edit Vendor 133
2.6.5 Reactivate Vendor 136
2.6.6 Deactivate Vendor 138
2.6.7 Create Vendor Type 140
2.6.8 Update Vendor Type 142
2.6.9 View Vendor Types 144
2.6.10 Reactivate Vendor Type 146
2.6.11 Deactivate Vendor Type 148
2.7 Project Management Subsystem Use Case Narratives 150
2.7.1 Create Project 150
2.7.2 View Project 153
2.7.3 Update Project 155
2.7.4 Deactivate Project 158
2.7.5 Create Project Type 160
2.7.6 View Project Type 162
2.7.7 Update Project Type 164
2.7.8 Deactivate Project Type 166
2.7.9 Add Indemnity Form 168
2.7.10 Update Indemnity Form 170
2.7.11 View Indemnity Form 172
2.7.12 Remove Indemnity Form 174
2.7.13 Capture Indemnity Form - 176

2.7.14 Capture Project Feedback - 178
2.7.15 View Project Feedback - 180
2.7.16 Create Impact Feedback Form 182
2.7.17 View Impact Feedback Form 184
2.7.18 Update Impact Feedback Form 186
2.7.19 Deactivate Impact Feedback Form 188
2.7.20 Capture Impact Feedback 190
2.7.21 View Impact Feedback 192
2.7.22 Add Photo 194
2.7.23 View Photo 196
2.7.24 Delete Photo 198
2.7.25 Add Photo Album 200
2.7.26 View Photo Album 202
2.7.27 Update Photo Album 204
2.7.28 Remove Photo Album 206

2.8 K9 Subsystem Use Case Narratives 208

2.8.1 Add Dog 208
2.8.2 View Dog 211
2.8.3 View Dogs 213
2.8.4 Update Dog 215
2.8.5 Deactivate Dog 218
2.8.6 Assign User to Dog 220
2.8.7 View User-Dog Assignment 223
2.8.8 Update User-Dog Assignment 225
2.8.9 Reactivate User-Dog Assignment 228
2.8.10 Deactivate User-Dog Assignment 230
2.8.11 Add K9 Role 232
2.8.12 Update K9 Role 235
2.8.13 View K9 Role 238
2.8.14 Reactivate K9 Role 240
2.8.15 Deactivate K9 Role 242
2.8.16 View K9 Ancestry 244

2.9 Rhino Subsystem Use Case Narratives 246

2.9.1 Add Rhino 246
2.9.2 View Rhino 249
2.9.3 View Rhinos 251
2.9.4 Update Rhino 253
2.9.5 Deactivate Rhino 256

2.10 Elephant Subsystem Use Case Narratives 258

2.10.1 Add Elephant 258
2.10.2 View Elephant 261
2.10.3 View Elephants 264
2.10.4 Update Elephant 266
2.10.5 Deactivate Elephant 269

2.11 Messages Subsystem Use Case Narratives 271
2.11.1 Create Message 271
2.11.2 View Messages 273
2.12 Report Subsystem Use Case Narratives 275
2.12.1 Generate Custom Report 275
2.12.2 View Custom Report 278
2.12.3 Update Custom Report 280
2.12.4 Remove Custom Report 283
2.12.5 View Bike Impact Report 285
2.12.6 View Budget Report 287
2.12.7 View Rhino Report 289
2.12.8 View Elephant Report 291
2.12.9 View K9 Report 293
2.12.10 View Machine Learning Dashboard 296
2.13 System Subsystem Use Case Narratives 298
2.13.1 Backup System 298
2.13.2 Restore System 300
2.13.3 View Audit Log 303
2.14 Budget Subsystem Use Case Narratives 305
2.14.1 View Project Budget 305
2.14.2 View Project Budgets 307
2.14.3 Log An Expense 309
2.14.4 Update An Expense 312
2.14.5 View Expense 315
2.14.6 Remove Expense 317
3. Conclusion 319
4. Sign-off by Client 320
5. Sign-off by Team 321

Table of Tables

Table 1: User Login - Logical Use Case Narrative 12
Table 2: User Logout - Logical Use Case Narrative 14
Table 3: Reset Password - Logical Use Case Narrative 17
Table 4: Create User - Logical Use Case Narrative 21
Table 5: Update User - Logical Use Case Narrative 24
Table 6: View User - Logical Use Case Narrative 26
Table 7:Reactivate User - Logical Use Case Narrative 28
Table 8: Deactivate User - Logical Use Case Narrative 30
Table 9: View FAQ - Logical Use Case Narrative 32
Table 10: Create User Profile - Logical Use Case Narrative 34
Table 11: Update User Profile - Logical Use Case Narrative 36
Table 12: View User Profile - Logical Use Case Narrative 38
Table 13: Deactivate User Profile - Logical Use Case Narrative 40
Table 14: Add FAQ – Logical Use Case Narrative 42
Table 15: Update FAQ – Logical Use Case Narrative 44
Table 16: Remove FAQ – Logical Use Case Narrative 46
Table 17: Add FAQ Type - Logical Use Case Narrative 48
Table 18: Update FAQ Type - Logical Use Case Narrative 50
Table 19: Remove FAQ Type - Logical Use Case Narrative 52
Table 20: Create Role - Logical Use Case Narrative 54
Table 21: View Role - Logical Use Case Narrative 56
Table 22: Update Role - Logical Use Case Narrative 58
Table 23: Reactivate Role - Logical Use Case Narrative 60
Table 24: Deactivate Role - Logical Use Case Narrative 62
Table 25: Create Role Permission - Logical Use Case Narrative 64
Table 26: View Role Permissions - Logical Use Case Narrative 66
Table 27: Create Procurement - Logical Use Case Narrative 69
Table 28: View Procurement - Logical Use Case Narrative 71
Table 29: View Inventory - Logical Use Case Narrative 74
Table 30: Write-Off Inventory - Logical Use Case Narrative 76
Table 31: Allocate Inventory - Logical Use Case Narrative 78
Table 32: Create Inventory Item Type - Logical Use Case Narrative 81
Table 33: View Inventory Item Types - Logical Use Case Narrative 83
Table 34: Update Inventory Item Type - Logical Use Case Narrative 86
Table 35: Deactivate Inventory Item Type - Logical Use Case Narrative 88
Table 36: Reactivate Inventory Item Type - Logical Use Case Narrative 90
Table 37: Create Group - Logical Use Case Narrative 93
Table 38: View Groups - Logical Use Case Narrative 96
Table 39: Update Group - Logical Use Case Narrative 99
Table 40: Deactivate Group - Logical Use Case Narrative 101
Table 41: Create Skill - Logical Use Case Narrative 104
Table 42: View Skills - Logical Use Case Narrative 106
Table 43: View Skill - Logical Use Case Narrative 108
Table 44: Update Skill - Logical Use Case Narrative 110
Table 45: Deactivate Skill - Logical Use Case Narrative 112

Table 46: Create Warehouse - Logical Use Case Narrative 114
Table 47: View Warehouse - Logical Use Case Narrative 116
Table 48: Update Warehouse - Logical Use Case Narrative 118
Table 49: Reactivate Warehouse -Logical Use Case Narrative 120
Table 50: Deactivate Warehouse - Logical Use Case Narrative 122
Table 51: Inventory Stock Take - Logical Use Case Narrative 125
Table 52: Create Vendor - Logical Use Case Narrative 128
Table 53: View Vendor - Logical Use Case Narrative 130
Table 54: View Vendors - Logical Use Case Narrative 132
Table 55: Edit Vendor - Logical Use Case Narrative 135
Table 56: Reactivate Vendor - Logical Use Case Narrative 137
Table 57: Deactivate Vendor - Logical Use Case Narrative 139
Table 58: Create Vendor Type - Logical Use Case Narrative 141
Table 59: Update Vendor Types - Logical Use Case Narrative 143
Table 60: View Vendor Types - Logical Use Case Narrative 145
Table 61: Reactivate Vendor Type - Logical Use Case Narrative 147
Table 62 :Deactivate Vendor Type - Logical Use Case Narrative 149
Table 63: Create Project - Logical Use Case Narrative 152
Table 64: View Project - Logical Use Case Narrative 154
Table 65: Update Project - Logical Use Case Narrative 157
Table 66: Deactivate Project - Logical Use Case Narrative 159
Table 67: Create Project Type - Logical Use Case Narrative 161
Table 68: View Project Type - Logical Use Case Narrative 163
Table 69: Update Project Type - Logical Use Case Narrative 165
Table 70: Deactivate Project Type - Logical Use Case Narrative 167
Table 71: Add Indemnity Form - Logical Use Case Narrative 169
Table 72: Update Indemnity Form - Logical Use Case Narrative 171
Table 73: View Indemnity Form - Logical Use Case Narrative 173
Table 74: Remove Indemnity Form - Logical Use Case Narrative 175
Table 75: Remove Indemnity Form - Logical Use Case Narrative 177
Table 76: Capture Project Feedback - Logical Use Case Narrative 179
Table 77: View Project Feedback - Logical Use Case Narrative 181
Table 78: Create Impact Feedback Form - Logical Use Case Narrative 183
Table 79: View Impact Feedback Forms - Logical Use Case Narrative 185
Table 80: Update Impact Feedback Form - Logical Use Case Narrative 187
Table 81: Deactivate Impact Feedback Form - Logical Use Case Narrative 189
Table 82: Capture Impact Feedback - Logical Use Case Narrative 191
Table 83: View Impact Feedback - Logical Use Case Narrative 193
Table 84: Add Photo - Logical Use Case Narrative 195
Table 85: View Photo - Logical Use Case Narrative 197
Table 86: Delete Photo - Logical Use Case Narrative 199
Table 84: Add Photo Album - Logical Use Case Narrative 201
Table 85: View Photo Album - Logical Use Case Narrative 203
Table 86: Update Photo Album - Logical Use Case Narrative 205
Table 86: Remove Photo Album - Logical Use Case Narrative 207
Table 87: Add Dog - Logical Use Case Narrative 210
Table 88: View Dog - Logical Use Case Narrative 212

Table 89: View Dogs - Logical Use Case Narrative 214
Table 90: 8.4 Update Dog - Logical Use Case Narrative 217
Table 91: Deactivate Dog - Logical Use Case Narrative 219
Table 92: Assign User to Dog - Logical Use Case Narrative 222
Table 93: View User-Dog Assignment - Logical Use Case Narrative 224
Table 94: Update User-Dog Assignment - Logical Use Case Narrative 227
Table 95: Reactivate User-Dog Assignment - Logical Use Case Narrative 229
Table 96: Deactivate User-Dog Assignment - Logical Use Case Narrative 231
Table 97: Add K9 Role - Logical Use Case Narrative 234
Table 98: Update K9 Role - Logical Use Case Narrative 237
Table 99: View K9 Roles - Logical Use Case Narrative 239
Table 100: Reactivate K9 Role - Logical Use Case Narrative 241
Table 101: Deactivate K9 Role - Logical Use Case Narrative 243
Table 102: View K9 Ancestry - Logical Use Case Narrative 245
Table 103: Add Rhino - Logical Use Case Narrative 248
Table 104: View Rhino - Logical Use Case Narrative 250
Table 105: View Rhino - Logical Use Case Narrative 252
Table 106: Update Rhino - Logical Use Case Narrative 255
Table 107: 9.5 Deactivate Rhino - Logical Use Case Narrative 257
Table 108: Add Elephant - Logical Use Case Narrative 260
Table 109: View Elephant - Logical Use Case Narrative 263
Table 110: View Elephants - Logical Use Case Narrative 265
Table 111: Update Elephant - Logical Use Case Narrative 268
Table 112: Deactivate Elephant - Logical Use Case Narrative 270
Table 113: Create Message - Logical Use Case Narrative 272
Table 114: View Messages - Logical Use Case Narrative 274
Table 115: Generate Custom Report - Logical Use Case Narrative 277
Table 116: View Custom Report - Logical Use Case Narrative 279
Table 117: Update Custom Report - Logical Use Case Narrative 282
Table 118: Remove Custom Report - Logical Use Case Narrative 284
Table 119: View Bike Impact Report - Logical Use Case Narrative 286
Table 120: View Budget Report - Logical Use Case Narrative 288
Table 121: View Rhino Report - Logical Use Case Narrative 290
Table 122: View Elephant Report - Logical Use Case Narrative 292
Table 123: View K9 Report - Logical Use Case Narrative 295
Table 124: View Machine Learning Dashboard - Logical Use Case Narrative 297
Table 125: Backup System - Logical Use Case Narrative 299
Table 126: Restore System - Logical Use Case Narrative 302
Table 127: View Audit Log - Logical Use Case Narrative 304
Table 128: View Project Budget - Logical Use Case Narrative 306
Table 129: View Project Budgets - Logical Use Case Narrative 308
Table 130: Log An Expense – Logical Use Case Narrative 311
Table 131: Update An Expense - Logical Use Case Narrative 314
Table 132: View Expense - Logical Use Case Narrative 316
Table 133: Remove Expense - Logical Use Case Narrative 318

1. Document Introduction

This document outlines the logical narratives of all the use cases defined for the proposed Integrated NGO Management System for EPI-USE and groupelephant.com.

2. Logical Use Case Narratives

2.1 User Subsystem Use Case Narratives

2.1.1 User Login

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 28 March 2026 **Version: 1.0**
Use case name:	User Login	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.1	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None.	
Other participating actors:	• None.	
Other interested stakeholders:	Administrator	
Description:	A user logs into the system using their credentials (email & password) to access a role-specific dashboard and gain access to all parts of the system they have permission to access.	
Pre-condition:	The user must already have a pre-existing account on the system	
Trigger:	The super administrator requests to create a backup of the database.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user enters their email address and password into the respective fields on the login screen and selects “Login” .	**Step 2:** The system retrieves the record associated with the provided email address form the User table. **User:** • UserID [PK] • UserName • UserSurname • UserDateOfBirth

INTEGRATED NGO MANAGEMENT SYSTEM		
		• UserEmail • UserPassword • UserStatus • UserAllergies • ProfilePicture • ActiveStatus • UserShirtSize [ALT]
		Step 3: The system hashes the password provided and compares it to the hashed password stored in the User table.
		Step 4: Upon finding a successful match, the system retrieves the user's role from the UserRole table. [ALT]
		Step 5: The system redirects she user to the appropriate dashboard.
		Step 6: Save login action to the Audit Log. **AuditLog:** • AuditLogID [PK] • UserID [FK] • Timestamp • LogEntry [ALT]
Alternate courses:	**[ALT] Step 2:** No record is found with a matching email address; the system suggests that the user sign's up (Use Case 1.4).	
	[ALT] Step 4: The hashed passwords do not match, and an error message is displayed. The user is redirected back to the login screen.	
	[ALT] Step 6: User cancels the save login action	

INTEGRATED NGO MANAGEMENT SYSTEM	
Conclusion:	User is successfully authenticated and granted access.
Post-condition:	An active session is created for the user.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None.
Open issues:	• None.

Table 1: User Login - Logical Use Case Narrative

2.1.2 User Logout

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 28 March 2026 **Version: 1.0**
Use case name:	User Logout	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.2	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None.	
Other participating actors:	• None.	
Other interested stakeholders:	Administrator	
Description:	A user logs out to terminate their session, preventing unauthorized physical access via their device.	
Pre-condition:	The user must be currently logged into the system.	
Trigger:	The user clicks "Logout".	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects Logout.	**Step 2:** The system terminates the current user session.
		Step 3: The system redirects the user to the login screen.
Alternate courses:	None	
Conclusion:	Session is terminated successfully.	
Post-condition:	Access to the system is revoked until the user logs in.	
Business rules:	• Secure Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Assumptions:	• None.
Open issues:	• None.

Table 2: User Logout - Logical Use Case Narrative

2.1.3 Reset Password

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 28 March 2026 **Version: 1.0**
Use case name:	Reset Password	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.3	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None.	
Other participating actors:	• Email Service	
Other interested stakeholders:	Administrator	
Description:	A user resets their password by receiving a link via email and entering a new password in the case that they forgot the password or believe that an unauthorized actor has access to their credentials.	
Pre-condition:	User must have an existing account.	
Trigger:	User clicks "Forgot Password" on the login page.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: User enters the email address associated to their profile on the password reset screen and selects “Reset Password”.	**Step 2:** The system searches for the email address in the User table to ensure that an account is associated with it. **User:** • UserID [PK] • UserName • UserSurname • UserDateOfBirth • UserEmail • UserPassword • UserStatus • UserAllergies

INTEGRATED NGO MANAGEMENT SYSTEM		
		• ProfilePicture • ActiveStatus • UserShirtSize [ALT]
		Step 3: Once the email is confirmed, the system informs the user to check their email address.
		Step 4: The system sends a reset password link to the confirmed email address.
	Step 5: The user opens the link they received and enters their desired new password in the provided field.	**Step 6:** The system hashes the new password and compares it to the Password field in the user table to ensure the old and new passwords do not match.
		Step 7: The system updates the Password field for the specific user in the User table. [ALT]
		Step 8: The system redirects the user to the Login screen.
		Step 9: Save password reset action to the Audit Log. **AuditLog:** • AuditLogID [PK] • UserID [FK] • Timestamp • LogEntry
Alternate courses:	**[ALT] Step 2:** The email address provided does not exist in the User table, the system suggests that the user Creates a profile (Use Case 1.4).	
	[ALT] Step 7: The password matches the previous password; the system displays an error message and prompts the user to provide another new password.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Conclusion:	Password successfully updated.
Post-condition:	The old password hash is replaced by the new hash.
Business rules:	• Ability to recover Account • Secure Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None.
Open issues:	• None.

Table 3: Reset Password - Logical Use Case Narrative

2.1.4 Create User

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1**
Use case name:	Create User	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.4	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a user signs up and creates a profile. The user will select to sign up for an account and enter their details into the relevant fields. Once done, the system will allow the user to login. On the first login, the user will complete the onboarding details screens. The use case ends once the new user is registered on the system.	
Pre-condition:	User does not already have an account.	
Trigger:	The user requests to create an account.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The user requests to create an account.	
	Step 2: The user selects to sign up.	Step 3: The system requests the user's details including: • First name • Surname • Phone number • Date of birth • Email • Password • Confirm Password

INTEGRATION NGO MANAGEMENT SYSTEM		
	Step 4: The user enters their details and selects to sign up. [ALT]	Step 5: The system validates the details entered. [ALT]
		Step 6: The system saves the new user in the following table: • User [ALT]
		Step 7: The system invokes use case 1.1 (User Login).
	Step 8: The user enters their login details.	Step 9: The system requests the user's details for the first onboarding step, requesting the following information: • Profile photo • Shirt size
	Step 10: The user selects their profile photo by either uploading a photo or selecting one of the available avatars as well as selects a shirt size.	Step 11: The system validates all required fields are completed. [ALT]
		Step 12: The system saves the user's details in the following table: • User [ALT]
		Step 13: The system requests the user's skills on the second onboarding step.
	Step 14: The user selects their skills from the available options and selects to continue.	Step 15: The system validates at least one skill is selected. [ALT]
		Step 16: The system saves the user's skills in the following table: • UserSkill [ALT]
		Step 17: The system requests the user's status in the last onboarding step.

INTEGRATION NGO MANAGEMENT SYSTEM		
	Step 18: The user selects their status and selects finish.	Step 19: The system validates that a status is selected. [ALT]
		Step 20: The system saves the user's status in the following table: • User [ALT]
		Step 21: The system saves the transaction in the AuditLog.
Alternate courses:	**[ALT] Step 4:** The user already has an account and selects to sign in. Use case terminates.	
	[ALT] Step 5: The validation is unsuccessful. Go back to step 4.	
	[ALT] Step 6: The system fails to create the user into the relevant entities and displays an error message.	
	[ALT] Step 11: The validation is unsuccessful. Go back to step 10.	
	[ALT] Step 12: The system fails to save the user's details in the relevant entities and displays an error message.	
	[ALT] Step 15: The validation is unsuccessful. Go back to step 14.	
	[ALT] Step 16: The system fails to save the user's details in the relevant entities and displays an error message.	
	[ALT] Step 19: The validation is unsuccessful. Go back to step 18.	
	[ALT] Step 20: The system fails to save the user's details in the relevant entities and displays an error message.	
Conclusion:	The user's profile is successfully created on the system.	
Post-condition:	• The user's profile is created on the system. • The user can be assigned to projects.	
Business rules:	• Valid email address is required.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature. • Passwords are hashed before storage in the database.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Assumptions:	• None
Open issues:	• None

Table 4: Create User - Logical Use Case Narrative

2.1.5 Update User

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1**
Use case name:	Update User	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.5	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a user updates their profile. The user will navigate to the profile icon in the navbar and select my profile. The system will display the user's current details on the system and allow the user to update their details. The user will update the details they wish to and then select save. The system will validate the details entered before updating the user's details on the system. The use case ends once the user's details are successfully updated on the system.	
Pre-condition:	User is logged into their account.	
Trigger:	User requests to update their details.	
Typical Course of Events:	Actor Action	System Response
	Step 1: User requests to update their details.	Step 2: The system invokes use case 1.6 (View User)
	Step 3: The user selects to update profile.	Step 4: The system populates the input fields with the user's current details including: • Profile Photo • Name • Surname • Email • Date of Birth

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Phone Number • Status • Shirt Size • Session Length • Skills
	Step 5: The user enters their updated details and selects to save the updated details. [ALT]	Step 6: The system validates the updated details entered. [ALT]
		Step 7: The system saves the updated details in the following tables: • User • UserSkill [ALT]
		Step 8: The system saves the transaction to the AuditLog table.
		Step 9: The system displays a success toaster informing the user that their details were updated successfully.
Alternate courses:	**[ALT] Step 5:** The user cancels their request to update their details. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 7: The system fails to update the user in the relevant entities and displays an error message.	
Conclusion:	The system displays a success toaster informing the user that their details were updated successfully.	
Post-condition:	The user's details are updated.	
Business rules:	• None	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	

INTEGRATION NGO MANAGEMENT SYSTEM	
Open issues:	• None

Table 5: Update User - Logical Use Case Narrative

2.1.6 View User

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1**
Use case name:	View User	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.6	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a user views their profile. The user will navigate to the profile icon in the navbar and select my profile. The system will retrieve and display the user's current details on the system. The use case ends once the user can view their profile.	
Pre-condition:	User is logged into their account.	
Trigger:	The user requests to view their profile.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The user requests to view their profile.	
	Step 2: The user selects the profile icon and then my profile.	Step 3: The system retrieves the user's details from the following tables: • User • UserSkill • Skill [ALT]

INTEGRATION NGO MANAGEMENT SYSTEM		
		Step 4: The system displays the user's details in a card format with the following information: • Profile Photo • Email • Name • Surname • Email • Phone number • Date of birth • Shirt size • Status • Session length • My skills
Alternate courses:	**[ALT] Step 3:** The system fails to retrieve the user's details from the relevant entities and displays an error toaster.	
Conclusion:	The user can view their profile.	
Post-condition:	The user can view their profile.	
Business rules:	• None	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 6: View User - Logical Use Case Narrative

2.1.7 Reactivate User

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1**
Use case name:	Reactivate User	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.7	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a user reactivates their account. The user will login into their account before being asked if they would like to reactivate their account. Once reactivated the system will display a success popup. The use case ends once the user's account is successfully reactivated on the system.	
Pre-condition:	User's account is deactivated.	
Trigger:	The user requests to reactivate their account on the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The user requests to reactivate their account on the system.	Step 2: The system invokes use case 1.1 (User Login).
	Step 3: The user enters their login details.	Step 4: The system asks if the user would like to reactivate their account.
	Step 5: The user confirms that they would like to reactivate their account. [ALT]	Step 6: The system sets the active status of the user to reactivated in the User table. [ALT]
		Step 7: The system saves the transaction to the AuditLog table.

INTEGRATION NGO MANAGEMENT SYSTEM		
		Step 8: The system displays a welcome back popup informing the user that their account has been successfully reactivated.
Alternate courses:	**[ALT] Step 5:** The user does not provide confirmation. Use case terminates.	
	[ALT] Step 6: The system fails to update the active status of the user in the relevant entities and displays an error message.	
Conclusion:	The system displays a welcome back popup informing the user that their account has been successfully reactivated.	
Post-condition:	• The user's account is reactivated on the system. • User can be assigned to projects.	
Business rules:	• None	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 7:Reactivate User - Logical Use Case Narrative

2.1.8 Deactivate User

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	Date: 8 August 2026 Version: 1
Use case name:	Deactivate User	USE CASE TYPE Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.8	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a user deactivates their account. The user will navigate to the profile icon in the navbar and select my profile. The system will display the user's current details on the system and allow the user to deactivate their account. Once deactivate the user is logged out of the system. The use case ends once the users account is successfully deactivated on the system.	
Pre-condition:	User's account is logged into their account. User's account is active.	
Trigger:	The user requests to deactivate their account on the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The user requests to deactivate their account on the system.	
	Step 2: The user selects the profile icon and then selects my profile.	Step 3: The system invokes use case 1.6 (View User).
	Step 4: The user selects to deactivate their account.	Step 5: The system requests confirmation.
	Step 6: The user confirms to deactivate their account. [ALT]	Step 7: The system sets the active status of the user to deactivated and the user status to "Not available for

INTEGRATION NGO MANAGEMENT SYSTEM		
		projects" in the User table. [ALT]
		Step 8: The system saves the transaction to the AuditLog table.
		Step 9: The system invokes use case 1.2 (User Logout).
Alternate courses:	**[ALT] Step 6:** The user does not provide confirmation. Use case terminates.	
	[ALT] Step 7: The system fails to update the active status and user status of the user in the relevant entities and displays an error toaster.	
Conclusion:	The user's account is deactivated, and the user is logged out.	
Post-condition:	• The user's account is deactivated on the system. • User cannot be assigned to projects.	
Business rules:	• None	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 8: Deactivate User - Logical Use Case Narrative

2.1.9 View FAQ

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date: 28 March 2026** **Version: 2.0**
Use case name:	View FAQ	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	1.9	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None.	
Other participating actors:	• None.	
Other interested stakeholders:	Administrator	
Description:	A user accesses a context-aware help section to resolve questions about the system or specific subsystems.	
Pre-condition:	The user must be logged in.	
Trigger:	The user clicks on "Help".	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: User clicks "Help" while in a specific subsystem.	**Step 2**: The system identifies the current application context. [ALT]
	.	Step 3: The system retrieves the FAQ table for entries matching the specific FAQType FAQ: • FAQID • FAQTypeID • Question • Answer • DateCreated
		Step 4: The system displays the relevant FAQ questions and answers to the user.

INTEGRATED NGO MANAGEMENT SYSTEM	
Alternate courses:	**[ALT] Step 2:** If the user clicks "Help" on the homepage, the system displays the full FAQ list.
Conclusion:	User receives relevant help information.
Post-condition:	None.
Business rules:	• Access to help function.
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None.
Open issues:	• None.

Table 9: View FAQ - Logical Use Case Narrative

2.2 Admin Subsystem Use Case Narratives

2.2.1 Create User Profile

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 9 August 2026** **Version: 1**
Use case name:	Create User Profile	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.1	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin creates a new user profile. The system will display all the users on the system and allow the super admin to create a new user profile. The super admin will enter the user's details and select to create the user. The use case ends once the user's profile is successfully saved on the system.	
Pre-condition:	Super admin is logged into their account.	
Trigger:	The super admin requests to create a user profile.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to create a user profile.	Step 2: The system invokes use case 2.3 (View User Profile)
	Step 3: The super admin selects to create a user profile.	Step 4: The system requests the user's details including: • Name • Surname • Email • Date of Birth • Phone Number

INTEGRATION NGO MANAGEMENT SYSTEM		
	Step 5: The user enters the new user's details and selects to create the new user. [ALT]	Step 6: The system validates the details entered. [ALT]
		Step 7: The system saves the new user in the following tables: • User [ALT]
		Step 8: The system sends an email to the new user using the email entered with a temporary password.
		Step 9: The system saves the transaction to the AuditLog table.
		Step 10: The system displays a success toaster informing the super admin that the new user was created successfully.
Alternate courses:	**[ALT] Step 5:** The user cancels their request to create a user profile. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 7: The system fails to create the user in the relevant entities and displays an error message.	
Conclusion:	The system displays a success toaster informing the super admin that the new user was created successfully.	
Post-condition:	The user can login and complete the onboarding steps.	
Business rules:	• Only super admins or user's with the required permissions can create user profiles.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• Super admin has the correct user details.	
Open issues:	• None	

Table 10: Create User Profile - Logical Use Case Narrative

2.2.2 Update User Profile

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 9 August 2026** **Version: 1**
Use case name:	Update User Profile	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.2	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin updates a user's profile. The system will display all the users on the system and allow the super admin to select a user profile to update. The system will display the user's current details and allow the user to update their details. The use case ends once the user's profile is successfully updated on the system.	
Pre-condition:	Super admin is logged into their account.	
Trigger:	The super admin requests to update a user profile.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to update a user profile.	Step 2: The system invokes use case 2.3 (View User Profile)
	Step 3: The user selects to update a user profile.	Step 4: The system displays the user's current details in editable fields including: • Name • Surname • Email • Date of Birth • Phone Number • Status • Shirt Size

INTEGRATION NGO MANAGEMENT SYSTEM		
	Step 5: The user updates the user's details and selects to save the updated details. [ALT]	Step 6: The system validates the details entered. [ALT]
		Step 7: The system saves the new user in the following tables: • User [ALT]
		Step 8: The system saves the transaction to the AuditLog table.
		Step 9: The system displays a success toaster informing the super admin that the user was updated successfully.
Alternate courses:	**[ALT] Step 5:** The user cancels their request to update a user profile. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 7: The system fails to update the user in the relevant entities and displays an error message.	
Conclusion:	The system displays a success toaster informing the super admin that the user was updated successfully.	
Post-condition:	The user's profile is updated on the system.	
Business rules:	• Only super admins or user's with the required permissions can update user profiles.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• Super admin has the correct updated user details.	
Open issues:	• None	

Table 11: Update User Profile - Logical Use Case Narrative

2.2.3 View User Profile

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 9 August 2026** **Version: 1**
Use case name:	View User Profile	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.3	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes when an admin views the details of all the users within the system. The system displays the users in a table containing the user's name, surname, email, date of birth, shirt size, and status. The use case ends when the admin sees all the users and their details.	
Pre-condition:	The admin must be logged into the system.	
Trigger:	The admin wants to view the users on the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The admin wants to view the users on the system.	
	Step 2: The admin hovers the admin tab and then selects users. [ALT]	**Step 3:** The system retrieves all the current users on the system from the User table.
		Step 4: The system displays the User Management page showing all the current users in a table: • Name • Surname • Email • Date of Birth • Shirt Size

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Status • Role • Actions [ALT]
	Step 5: The admin can view the users.	
Alternate courses:	**ALT-Step 2:** User is on mobile and selects the menu icon. Go to step 3.	
	ALT-Step 4: The system does not find any current users. Terminate use case.	
Conclusion:	All users' details are displayed in a table.	
Post-condition:	The system displays all the current users in a table.	
Business rules:	• Only super admins or users given appropriate permissions can view user's profiles.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 12: View User Profile - Logical Use Case Narrative

2.2.4 Deactivate User Profile

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 9 August 2026** **Version: 1**
Use case name:	Deactivate User	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.4	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a Super Admin deactivates a user in the system. The system allows the Super Admin to select deactivate on a user and will then confirm the deactivation. Once confirmed, the system updates the user's status to inactive in the database. The use case ends when the user's status is changed to inactive.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to deactivate a user on the system.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin wants to deactivate a user on the system.	**Step 2:** The system invokes UC 2.3 (View User Profile) to display all the current users.
	Step 3: The super admin selects deactivate on a specific user.	**Step 4:** The system requests confirmation for the user deactivation.
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system updates the user's status to be inactive in the User table. [ALT]
		Step 7: The system records the transaction to the AuditLog table.

INTEGRATION NGO MANAGEMENT SYSTEM		
		Step 8: The system displays a confirmation toaster.
Alternate courses:	**ALT-Step 5:** The super admin does not confirm. Go back to step 2.	
	ALT-Step 6: The system fails to deactivate the user and displays a toaster message. Terminate use case.	
Conclusion:	The active user has been deactivated, and a confirmation toaster is shown.	
Post-condition:	The system updates the user's status to inactive in the User table.	
Business rules:	• Only super admin or users given appropriate permissions can deactivate a user.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature. •	
Assumptions:	• None	
Open issues:	• None	

Table 13: Deactivate User Profile - Logical Use Case Narrative

2.2.5 Add FAQ

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 21 March 2026** **Version: 1.0**
Use case name:	Add FAQ	**USE CASE TYPE**
Use case id:	2.5	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• User	
Other interested stakeholders:	None	
Description:	This use case describes the process to add a frequently asked question within the system in a controlled and secure manner.	
Pre-condition:	• Admin is authenticated and authorised • Required related data such as the FAQ types exist in the system	
Trigger:	The Admin selects the relevant option being "Add FAQ" in the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The Admin selects the "Add FAQ" function.	**Step 2:** The system retrieves the relevant data from the database.
	Step 3: The Admin inputs the FAQ.	**Step 4:** The System validates data against business rules and relationships [ALT]
	Step 5: The Admin confirms the action to add the FAQ [ALT].	**Step 6:** The System updates the new FAQ record while maintaining referential integrity.
		Step 7: The System confirms successful completion.
Alternate courses:	**[ALT] Step 4:**	

INTEGRATED NGO MANAGEMENT SYSTEM	
	If validation fails, the System displays an error and returns to step 3.
	[ALT] Step 5 User cancels the function to add a FAQ
Conclusion:	The operation to add the FAQ is completed successfully.
Post-condition:	• The system database reflects the completed operation
Business rules:	• Only authorised users may perform this action.
Implementation constraints and specifications:	• Input validation must be applied. • Role-based access control must be enforced.
Assumptions:	• Admin has sufficient knowledge of the system • System is functioning correctly.
Open issues:	• None

Table 14: Add FAQ – Logical Use Case Narrative

2.2.6 Update FAQ

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Update FAQ	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.6	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• User	
Other interested stakeholders:	None	
Description:	This use case allows editing of an existing FAQ while maintaining its link to a valid FAQ Type. The System ensures that the updated FAQ remains valid, uniquely identified, and correctly categorised.	
Pre-condition:	• Admin is authenticated • FAQ exists in the system • At least one FAQ Type exists	
Trigger:	Admin selects an FAQ and clicks “Update”	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects the FAQ to update.	**Step 2:** System retrieves the FAQ details including its faqtype_id.
	Step 3: Admin edits the question, answer, or FAQ Type.	**Step 4:** The System validates that the new FAQ Type exists [ALT]
		Step 5: The System validates that the updated questions does not duplicate another FAQ.
	Step 6: The Admin confirms the update [ALT].	**Step 7:** System updates the FAQ record in the database.
		Step 8: System logs the update action.

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 9: System displays confirmation.
Alternate courses:	**[ALT] Step 4:** Duplicate question detected and the System rejects the update and returns to step 3.	
.	**[ALT] Step 6:** User cancels the function to update an FAQ.	
Conclusion:	The FAQ is successfully updates while maintaining valid categorisation.	
Post-condition:	• Updated FAQ is stored.	
Business rules:	• Questions must be unique • Each FAQ must belong to a valid FAQ Type	
Implementation constraints and specifications:	• Maintain referential integrity	
Assumptions:	• Admin reviews content before updating	
Open issues:	• None	

Table 15: Update FAQ – Logical Use Case Narrative

2.2.7 Remove FAQ

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Remove FAQ	**USE CASE TYPE**
Use case id:	2.7	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• User	
Other interested stakeholders:	None	
Description:	This use case describes an event that will allow the Admin to remove an FAQ from the system.	
Pre-condition:	• Admin is authenticated • The FAQ exists	
Trigger:	Admin selects the "Remove FAQ" function.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects FAQ.	**Step 2:** The System retrieves FAQ.
	Step 3: Admin selects "Remove".	
	Step 4: Admin confirms deletion [ALT].	**Step 5:** System removes FAQ.
		Step 6: The System logs the action.
		Step 7: System confirms successful removal.
Alternate courses:	**[ALT] Step 4:** User cancels FAQ removal function	
Conclusion:	The FAQ is removed from active use.	
Post-condition:	• FAQ is no longer visible to users.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Business rules:	• FAQs should not be removed if they are required for audit.
Implementation constraints and specifications:	• None
Assumptions:	• Removal policy is defined.
Open issues:	• None

Table 16: Remove FAQ – Logical Use Case Narrative

2.2.8 Add FAQ Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 14 August 2026** **Version: 1.0**
Use case name:	Add FAQ Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.8	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• User	
Other interested stakeholders:	None	
Description:	This use case describes an event that will allow the Super Admin to capture and commit a new FAQ classification type to the global lookup directory via a reactive form modal.	
Pre-condition:	• Admin is authenticated	
Trigger:	Admin selects the "Add FAQ Type" function.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects "Add FAQ Type".	Step 2: The System displays the Add FAQ Type overlay modal form.
	Step 3: Admin enters the Type Name (max 100 characters), Description, and Active status.	
	Step 4: Admin submits the form [ALT].	Step 5: System validates the input fields and displays inline feedback for any errors.
		Step 6: System dispatches the payload over HTTPS and commits the new FAQ Type to the lookup directory.

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: System confirms successful addition and closes the modal.
Alternate courses:	**[ALT] Step 4:** Admin cancels the Add FAQ Type function, and the modal is dismissed without saving.	
Conclusion:	The new FAQ Type is added to the system and becomes available for classifying FAQs.	
Post-condition:	• The new FAQ Type is available for selection when creating or editing FAQs.	
Business rules:	• Type Name must not exceed a data allocation size ceiling of 100 characters. • Type Name and Description are mandatory fields.	
Implementation constraints and specifications:	• Angular Reactive Form presented within an overlay modal layout. • Persistent HTTPS connection required to dispatch payloads to host environments. • Validation feedback must be responsive, clean, and intuitive for minimally trained admin users.	
Assumptions:	• The FAQ Type classification structure has been defined and approved.	
Open issues:	• None	

Table 17: Add FAQ Type - Logical Use Case Narrative

2.2.9 Update FAQ Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 14 August 2026** **Version: 1.0**
Use case name:	Update FAQ Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.9	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	None	
Description:	This use case describes an event that will allow the Super Admin to alter the name, description, or status of an existing FAQ Type via a contextual edit modal that auto-populates the current data.	
Pre-condition:	• Admin is authenticated • The FAQ Type exists.	
Trigger:	Admin selects the "Update FAQ Type" function.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects an FAQ Type.	Step 2: The System retrieves the FAQ Type and opens the edit modal, patching current values into the form controls.
	Step 3: Admin edits the Type Name, Description, and/or Active status.	
	Step 4: Admin submits the changes [ALT].	Step 5: System packages the mutated fields into a DTO and dispatches an asynchronous HTTP PUT request to api/FaqType/update/{id}.

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 6: System validates and saves the updated FAQ Type.
		Step 7: System confirms successful update and closes the modal.
Alternate courses:	**[ALT] Step 4:** Admin cancels the Update FAQ Type function, and the modal is dismissed without saving.	
Conclusion:	The FAQ Type is updated and reflects the latest information.	
Post-condition:	• The updated FAQ Type details are reflected wherever the type is referenced.	
Business rules:	• Secure access.	
Implementation constraints and specifications:	• Modifications must be processed asynchronously through DTOs to insulate the database from the network layer.	
Assumptions:	• The existing FAQ Type data retrieved for editing is accurate and current.	
Open issues:	• None	

Table 18: Update FAQ Type - Logical Use Case Narrative

2.2.10 Remove FAQ Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 14 August 2026** **Version: 1.0**
Use case name:	Remove FAQ Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.10	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	None	
Description:	This use case describes an event that will allow the Super Admin to remove an FAQ Type from the system database when it is no longer required, subject to an explicit modal confirmation prompt.	
Pre-condition:	• Admin is authenticated • The FAQ Type exists.	
Trigger:	Admin selects the "Remove FAQ Type" function.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects an FAQ Type row and clicks "Remove".	Step 2: The System displays a modal verification confirmation overlay.
	Step 3: Admin confirms deletion [ALT].	Step 4: System validates database dependency maps for the selected FAQ Type.
		Step 5: System dispatches an asynchronous HTTP DELETE request to api/FaqType/remove/{id}.
		Step 6: System removes the FAQ Type row from disk storage.

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: System confirms successful removal.
Alternate courses:	**[ALT] Step 3:** Admin cancels the deletion and the confirmation modal is dismissed without changes.	
Conclusion:	The FAQ Type is removed from the system.	
Post-condition:	• The FAQ Type is no longer available for selection when creating or editing FAQs.	
Business rules:	• Secure access.	
Implementation constraints and specifications:	• Destructive actions must never process silently; an explicit verification prompt must always interrupt execution.	
Assumptions:	• The FAQ Type has no active dependencies that would be broken by its removal.	
Open issues:	• None	

Table 19: Remove FAQ Type - Logical Use Case Narrative

2.2.11 Create Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Create Role	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.11	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• User	
Other interested stakeholders:	None	
Description:	This use case describes an event where an Admin is required to create a role.	
Pre-condition:	• Admin is authenticated	
Trigger:	Admin selects “Create Role”.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects “Create Role”.	**Step 2:** The system displays the role creation form.
	Step 3: Admin fills in the form.	**Step 4:** The system displays the form with the details filled in by the Admin.
	Step 5: Admin confirms the details.[ALT}	**Step 6:** The System validates the details.
		Step 7: The System successfully creates the role and displays confirmation.
Alternate courses:	**[ALT] Step 5:** The user cancels the create role function.	
Conclusion:	Admin successfully created role.	
Post-condition:	• Role available for use.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Business rules:	• Only authorised users may create roles.
Implementation constraints and specifications:	• None •
Assumptions:	• None

Table 20: Create Role - Logical Use Case Narrative

2.2.12 View Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	View Role	**USE CASE TYPE**
Use case id:	2.12	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case enables the Admin to retrieve details of roles already defined in the system, including role names, descriptions, and associated permissions.	
Pre-condition:	• At least one role exists in the system • Admin is authenticated	
Trigger:	Admin selects "View Role"	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin requests to view roles.	**Step 2:** System retrieves role records from the database.
		Step 3: System retrieves associated permissions for each role [ALT]
		Step 4: System compiles role data.
		Step 5: System displays roles in a structured format.
	Step 6: Admin reviews role details.	
Alternate courses:	**[ALT] Step 3:** Permission retrieval fails and the system displays role without permission and logs an error.	
Conclusion:	Admin successfully reviews role configurations.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	• Role visibility is confirmed.
Business rules:	• Only authorised Admins can view roles.
Implementation constraints and specifications:	• Must support efficient querying of role-permission relationships.
Assumptions:	• Role permission relationships are correctly maintained.

Table 21: View Role - Logical Use Case Narrative

2.2.13 Update Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Update Role	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.13	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to modify existing role details, including role name and description, to better reflect organisational responsibilities or changes in system access policies.	
Pre-condition:	• Admin is authenticated • Role must exist	
Trigger:	Admin selects “Update Role”	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects a role.	**Step 2:** System retrieves the role details.
		Step 3: System displays all the editable fields.
	Step 4: The Admin updates role information.	**Step 5:** System should validate uniqueness of role name [ALT]
	Step 6: Admin confirms new update [ALT].	**Step 7:** System updates role record.
		Step 8: System logs modification.
		Step 9: System confirms success.

INTEGRATED NGO MANAGEMENT SYSTEM	
Alternate courses:	**[ALT] Step 5:** Duplicate role name therefore the system rejects and returns to step 4.
	[ALT] Step 6: The user cancels the update function.
Conclusion:	Role successfully updated.
Post-condition:	• Updated role stored.
Business rules:	• Role names mist be unique.
Implementation constraints and specifications:	• None
Assumptions:	• None

Table 22: Update Role - Logical Use Case Narrative

2.2.14 Reactivate Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Reactivate Role	**USE CASE TYPE**
Use case id:	2.14	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to restore a previously deactivated role, making it available again for assignment to users.	
Pre-condition:	• Admin is authenticated • Role exists and is inactive	
Trigger:	Admin selects "Activate Role"	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects the inactive role.	Step 2: System retrieves role status [ALT]
	Step 3: Admin confirms reactivation [ALT].	Step 4: System updates the role status to active.
		Step 5: System logs actions.
		Step 6: System confirms success.
Alternate courses:	**[ALT] Step 2:** Role already exists so the system prevents the action.	
	[ALT] Step 3: The user cancels the role reactivation function.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Conclusion:	Role successfully reactivated.
Post-condition:	• Role is available for assignment.
Business rules:	• Only inactive roles can be reactivated.
Implementation constraints and specifications:	• None
Assumptions:	• Role is still valid for current system use

Table 23: Reactivate Role - Logical Use Case Narrative

2.2.15 Deactivate Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Deactivate Role	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.15	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to deactivate a role, preventing it from being assigned to users and restricting its use in access control without permanently deleting it.	
Pre-condition:	• Admin is authenticated • The role exists	
Trigger:	Admin selects “Deactivate Role”.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects the role.	**Step 2:** System checks if the role is assigned to users.
		Step 3: System displays impact warming.
	Step 4: Admin confirms deactivation [ALT].	**Step 5:** System updates the tole status to inactive.
		Step 6: System logs action.
		Step 7: The System confirms success.
Alternate courses:	**[ALT] Step 4:** The user cancels the role deactivation function.	
Conclusion:	Role successfully deactivated.	
Post-condition:	• Role cannot be assigned to users.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Business rules:	• Roles assigned to users may require reassignment before deactivation.
Implementation constraints and specifications:	• None
Assumptions:	• None

Table 24: Deactivate Role - Logical Use Case Narrative

2.2.16 Update Role Permissions

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Update Role Permission	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.16	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to modify existing permissions assigned to a role, enabling dynamic adjustment of access rights as system requirements or organisational responsibilities change.	
Pre-condition:	• Admin is authenticated • Role and associated permissions exist	
Trigger:	Admin selects "Update Role Permission".	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects a role.	**Step 2:** System retrieves current permissions.
		Step 3: System displays editable permission matrix.
	Step 4: Admin modifies permissions.	**Step 5:** System validated permission.
	Step 6: Admin confirms the update. [ALT].	**Step 7:** System updates and logs changes.
		System 8: System confirms success.
Alternate courses:	**[ALT} Step 6**: User cancels the update function.	
Conclusion:	Role permissions successfully updated.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	• Updated permissions enforced in the system
Business rules:	• Changes must take effect immediately
Implementation constraints and specifications:	• None
Assumptions:	• None

Table 25: Create Role Permission - Logical Use Case Narrative

2.2.17 View Role Permission

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	View Role Permission	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	2.17	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to view permissions assigned to a role.	
Pre-condition:	• Admin is authenticated • Role must exist in the system	
Trigger:	Admin selects "View Role Permission"	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects role.	**Step 2:** System retrieves role details.
		Step 3: System retrieves associated permissions.
		Step 4: System retrieves associated permissions by subsystem.
	Step 5: Admin reviews permission.	
Alternate courses:	None	
Conclusion:	Admin successfully reviews role permissions.	
Post-condition:	• No changes made	
Business rules:	• Permissions must reflect current system configuration	

INTEGRATED NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• None
Assumptions:	• None

Table 26: View Role Permissions - Logical Use Case Narrative

2.3 Inventory Subsystem Use Case Narratives

2.3.1 Create Procurement

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 4 August 2026** **Version: 1.0**
Use case name:	Create Procurement	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.1	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which an admin creates a procurement after receiving inventory. The system will request the procurement details including the vendor, warehouse, procurement date, project (optional), invoice and the inventory items received. Once entered the admin will select to create the procurement. The system will validate the procurement's details before saving the procurement to the system. The use case ends once the system displays the success toaster informing the admin of the successful procurement creation.	
Pre-condition:	Admin is logged in.	
Trigger:	The admin requests to create a procurement.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The admin requests to create a procurement.	Step 2: The system invokes use case 3.3 (View Inventory).
	Step 3: The admin selects to create a procurement.	Step 4: The system will request the procurement details including:

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Warehouse • Procurement Date • Project (Optional) • Invoice • Inventory Items: ○ Inventory Type ○ Model Name ○ Quantity • Price
	Step 5: The admin enters the procurement details and selects to create the procurement. [ALT]	Step 6: The system validates that all required fields are completed. [ALT]
		Step 7: The system creates the procurement and adds the inventory to the following tables: • Procurement • ProcurementLine • Inventory [ALT]
Alternate courses:	**[ALT] Step 5:** The admin cancels their request to create a procurement. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful; the admin is prompted to enter the details into the empty fields. Go back to step 5.	
	[ALT] Step 7: The system fails to create the procurement into the relevant entities and displays an error message to the admin.	
Conclusion:	The system displays a success toaster informing the admin that the procurement was created successfully.	
Post-condition:	The procurement is created and inventory is added to the system.	
Business rules:	• Ony super admins, admins or users with the required permissions can create a procurement.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature. • Each inventory item is stored individually in the Inventory table.
Assumptions:	• None
Open issues:	• None

Table 27: Create Procurement - Logical Use Case Narrative

2.3.2 View Procurement

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 4 August 2026** **Version: 1**
Use case name:	View Procurement	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.2	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which an admin views a procurement. The system will display the procurement id, date, total items, an invoice link as well as the inventory item details. The use case ends once the admin views the procurement.	
Pre-condition:	Admin is logged in.	
Trigger:	Admin wants to view a procurement	
Typical Course of Events:	Actor Action	System Response
	Step 1: Admin wants to view a procurement.	Step 2: The system invokes use case 3.3 (View Inventory).
	Step 3: The admin selects to view inventory and then the procurement they wish to view.	Step 4: The system retrieves and displays the selected procurement from the following tables: • Procurement • ProcurementLine • InventoryType [ALT]
		Step 5: The system displays the procurement details modal.
Alternate courses:	**[ALT] Step 4:**	

	The system fails to retrieve the procurement's details. The system will display an error toaster message.
Conclusion:	The system displays the procurement details modal to the admin.
Post-condition:	The system displays the procurement details modal to the admin.
Business rules:	• Only super admins, admins or users with the required permissions can view a procurement.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 28: View Procurement - Logical Use Case Narrative

2.3.3 View Inventory

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 4 August 2026** **Version: 1**
Use case name:	View Inventory	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.3	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which an admin views inventory. The admin will navigate to the Inventory tab in the navbar, and the system will retrieve and display all inventory on the system where the admin can filter and search inventory. The use case ends once the admin views inventory.	
Pre-condition:	Admin is logged in.	
Trigger:	Admin requests to view inventory.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Admin requests to view inventory.	
	Step 2: The super admin will navigate to and select the inventory tab. [ALT]	Step 3: The system retrieves all the inventory on the system from the following tables: • Inventory • InventoryType • Warehouse • ProcurementLine [ALT]
		Step 4: The system displays the inventory totals.

	Step 5: The admin selects to see the inventory view. [ALT]	Step 6: The system displays the inventory list.
	Step 7: The admin filters the inventory by status, inventory type, warehouse or by entering the inventory type, model name, warehouse and/or location in the search bar. [ALT]	Step 8: The system filters and displays the inventory matching the filter criteria. [ALT]
Alternate courses:	**[ALT] Step 2:** The admin is using a mobile device such as a phone. The admin will select the menu (hamburger) icon and then the inventory option. Continue to step 3.	
	[ALT] Step 3: The system fails to retrieve the inventory totals. The system will display an error toaster message.	
	[ALT] Step 5: The admin stays on the quantity totals view and filters the inventory items by inventory type and/or status.	
	[ALT] Step 7: The admin does not filter and just views inventory.	
	[ALT] Step 8: No inventory is found matching the filter criteria. The system will inform the admin that no inventory matches the selected filters. Go back to step 7.	
Conclusion:	The system filters and displays the inventory matching the criteria.	
Post-condition:	The inventory is displayed.	
Business rules:	• Only super admins, admins or users with the required permissions can view inventory.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	

Open issues:	• None

Table 29: View Inventory - Logical Use Case Narrative

2.3.4 Write-Off Inventory

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 4 August 2026** **Version: 1**
Use case name:	Write-Off Inventory	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.4	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which an admin writes-off inventory. The system displays the inventory records and allows the admin to select inventory for the write-off process. The system will request the amount of inventory to be written off as well as a reason for the write-off. Once entered the system will validate the details entered before updating the status of the selected inventory items to Written Off. The use case ends when a success toast notification is displayed, confirming that the selected inventory items have been successfully updated.	
Pre-condition:	Admin is logged in.	
Trigger:	The admin requests to write-off inventory.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The admin requests to write-off inventory.	Step 2: The system invokes use case 3.3 (View Inventory).
	Step 3: The admin selects to write off inventory.	Step 4: The system requests the write off details including: • Quantity to write off • Reason for write off

	Step 5: The admin enters the write off details. [ALT]	Step 6: The system validates that all required fields are completed and the quantity to be written off. [ALT]
		Step 7: The system updates the status of the inventory to written off in the following table: • Inventory [ALT]
		Step 8: The system saves the transaction to the AuditLog table.
		Step 9: The system displays a success toaster informing the admin that the inventory has been written off successfully.
Alternate courses:	**[ALT] Step 5:** Admin selects to cancel their write off inventory request. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 7: The system fails to write-off the inventory in the relevant entities and displays an error message to the admin.	
Conclusion:	The system displays a success toaster informing the admin that the inventory has been written off successfully.	
Post-condition:	The inventory is written off.	
Business rules:	• Only super admins, admins or users with the required permissions can write off inventory.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature. • Only available inventory can be written off.	
Assumptions:	• None	
Open issues:	• None	

Table 30: Write-Off Inventory - Logical Use Case Narrative

2.3.5 Allocate Inventory

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 4 August 2026** **Version: 1**
Use case name:	Allocate Inventory	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.5	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which an admin allocates inventory. The system displays the inventory records and allows the admin to select inventory for the allocation process. The system requests the project as well as the quantity of inventory to be allocated. The system validates the details before updating the status of the selected inventory items to allocated. The use case ends when a success toast notification is displayed, confirming that the selected inventory items have been successfully allocated.	
Pre-condition:	Admin is logged in.	
Trigger:	The admin requests to allocate inventory.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The admin requests to allocate inventory.	Step 2: The system invokes use case 3.3 (View Inventory)
	Step 3: The admin selects to allocate inventory.	Step 4: The system requests the allocate details including: • Project • Quantity
	Step 5: The admin selects a project and enters the	Step 6: The system validates that all required fields are

	quantity they wish to allocate. [ALT]	completed and the quantity to be allocated. [ALT]
		Step 7: The system updates the inventory status and project id in the following table: • Inventory [ALT]
		Step 8: The system saves the transaction to the AuditLog table.
		Step 9: The system displays a success toaster informing the admin that the inventory has been allocated successfully.
Alternate courses:	**[ALT] Step 5:** Admin has chosen to cancel their allocate inventory request. Use case terminates	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 7: The system fails to allocate the inventory in the relevant entities and displays an error message to the admin.	
Conclusion:	The system displays a success toaster informing the admin that the inventory has been allocated successfully.	
Post-condition:	The inventory is allocated to a project.	
Business rules:	• Only super admins, admins or users with the required permissions can allocate inventory. • Only available inventory can be allocated. • Inventory can only be allocated to active projects.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 31: Allocate Inventory - Logical Use Case Narrative

2.3.6 Create Inventory Item Type

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 4 August 2026** **Version: 1**
Use case name:	Create Inventory Item Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.6	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin creates an inventory item type. The system will display inventory type records and allow the super admin to create a new type. The system will request the new inventory item types details and allow the super admin to enter the details. Once entered the system validates the details and creates the new inventory item type. The use case ends once the success toaster is displayed and the new inventory item type has successfully been captured on the system.	
Pre-condition:	Super admin is logged in.	
Trigger:	The super admin requests to create a new inventory item type.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to create a new inventory item type.	Step 2: The system invokes use case 3.7 (View Inventory Item Types)
	Step 3: The super admin will select to create a new inventory item type.	Step 4: The system requests the new inventory item type's details including: • Name • Security Level

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Description • Whether it is a depreciating asset or not
	Step 5: The super admin enters the new inventory item type details and selects to create the inventory item type. [ALT]	Step 6: The system validates that all required fields are completed. [ALT]
		Step 7: The system adds the new inventory item type to the following table: • InventoryType [ALT]
		Step 8: The system saves the transaction to the AuditLog table.
		Step 9: The system displays a success toaster informing the super admin that the inventory item type has been created successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects to cancel their inventory item type request. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 7: The system fails to add the inventory item type into the relevant entities and displays an error message to the super admin.	
Conclusion:	The system displays a success toaster informing the super admin that the inventory item type has been created successfully.	
Post-condition:	The inventory item type is created successfully and can be used for new procurements.	
Business rules:	• Only super admins or users with the required permissions can create an inventory item type.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 32: Create Inventory Item Type - Logical Use Case Narrative

2.3.7 View Inventory Item Types

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 4 August 2026** **Version: 1**
Use case name:	View Inventory Item Types	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.7	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin views inventory item types. The super admin will navigate to the Inventory tab in the navbar. The system will display inventory records and allow the super admin to manage inventory types. The system will retrieve and display inventory type records. The use case ends once the inventory item types are displayed.	
Pre-condition:	Super admin is logged in.	
Trigger:	The super admin requests to view inventory item types.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to view inventory item types.	Step 2: The system invokes use case 3.3 (View Inventory).
	Step 3: The super admin will select to manage inventory item types.	Step 4: The system retrieves all the inventory item types on the system from the following table: • InventoryType [ALT]
Alternate courses:	**[ALT] Step 4:**	

INTEGRATION NGO MANAGEMENT SYSTEM	
	The system fails to retrieve the inventory item types. The system will display an error toaster message.
Conclusion:	The super admin can view all the inventory item types.
Post-condition:	All inventory items are displayed.
Business rules:	• Only super admins or users with the required permissions can view inventory item types.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 33: View Inventory Item Types - Logical Use Case Narrative

2.3.8 Update Inventory Item Type

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 6 August 2026** **Version: 1**
Use case name:	Update Inventory Item Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.8	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin updates an inventory item type. The super admin will navigate to and select to manage inventory types and the system will retrieve and display inventory type records. The super admin will select the inventory item type they wish to update and the system will request the updated details. Once entered the system will validate the details. The use case ends once the success toaster is displayed and the inventory item type is updated on the system.	
Pre-condition:	Super Admin is logged into the system. The inventory item type exists and is active.	
Trigger:	The super admin requests to update an inventory item type.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to update an inventory item type.	Step 2: The system invokes use case 3.7 (View Inventory Item Types)
	Step 3: The super admin finds and selects to update an inventory item type.	Step 4: The system retrieves the selected inventory item type's current details from the following table: • InventoryType

INTEGRATION NGO MANAGEMENT SYSTEM		
		[ALT]
		Step 5: The system requests the super admin to enter the inventory item types updated details including: • Name • Description • Security Level • Whether it is a depreciating asset or not
	Step 6: The super admin enters the updated details and selects to save changes. [ALT]	Step 7: The system validates that all required fields are completed. [ALT]
		Step 8: The system saves the updated details in the following table: • InventoryType [ALT]
		Step 9: The system saves the transaction to the AuditLog table.
		Step 10: The system displays a success toaster informing the super admin that the inventory item type was updated successfully.
Alternate courses:	**[ALT] Step 4:** The system fails to retrieve the inventory item type from the relevant entities and displays an error toaster to the super admin.	
	[ALT] Step 6: The super admin selects cancel their update inventory item type request. Use case terminates.	
	[ALT] Step 7: The validation is unsuccessful. Go back to step 6.	
	[ALT] Step 8: The system fails to update the inventory item type in the relevant entities and displays an error message to the super admin.	
Conclusion:	The system displays a success toaster informing the super admin that the inventory item type was updated successfully.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Post-condition:	The inventory item type's details are updated.
Business rules:	• Only super admins or users with the required permissions can update an inventory item type.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 34: Update Inventory Item Type - Logical Use Case Narrative

2.3.9 Deactivate Inventory Item Type

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 6 August 2026** **Version: 1**
Use case name:	Deactivate Inventory Item Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.9	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin deactivates an inventory item type. The super admin will navigate to and select to manage inventory types and the system will retrieve and display inventory type records. The super admin will select the inventory item type they wish to deactivate and the system will request confirmation. The use case ends once the success toaster is displayed and the inventory item type is deactivated on the system.	
Pre-condition:	Super Admin is logged into the system. The inventory item type is active.	
Trigger:	The super admin requests to deactivate an inventory item type.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to deactivate an inventory item type.	Step 2: The system invokes use case 3.7 (View Inventory Item Types)
	Step 3: The super admin finds and selects to deactivate an inventory item type.	Step 4: The system requests confirmation for the inventory item type to be deactivated.
	Step 5: The super admin confirms that the inventory item type can be deactivated. [ALT]	Step 6: The system sets the status of the inventory item type to deactivated in the following table:

INTEGRATION NGO MANAGEMENT SYSTEM		
		• InventoryType [ALT]
		Step 7: The system saves the transaction to the AuditLog table.
		Step 8: The system displays a success toaster informing the super admin that the inventory item type was deactivated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects to cancel their deactivate inventory item type request. Use case terminates.	
	[ALT] Step 6: The system fails to update the status of the inventory item type in the relevant entities and displays an error toaster to the super admin. Use case terminates.	
Conclusion:	The system displays a success toaster informing the super admin that the inventory item type was deactivated successfully.	
Post-condition:	• The inventory item type is deactivated on the system. • The inventory item type can no longer be used when creating a procurement.	
Business rules:	• Only super admins or users with the required permissions can deactivate an inventory item type.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 35: Deactivate Inventory Item Type - Logical Use Case Narrative

2.3.10 Reactivate Inventory Item Type

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 6 August 2026** **Version: 1**
Use case name:	Reactivate Inventory Item Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	3.10	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin reactivates an inventory item type. The super admin will navigate to and select to manage inventory types and the system will retrieve and display inventory type records. The super admin will select the inventory item type they wish to reactivate and the system will request confirmation. The use case ends once the success toaster is displayed and the inventory item type is reactivated on the system.	
Pre-condition:	Super Admin is logged into the system. The inventory item type is inactive.	
Trigger:	The super admin requests to reactivate an inventory item type.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to reactivate an inventory item type.	Step 2: The system invokes use case 3.7 (View Inventory Item Types)
	Step 3: The super admin finds and selects to reactivate an inventory item type.	Step 4: The system requests confirmation for the inventory item type to be reactivated.
	Step 5: The super admin confirms that the inventory item type can be reactivated. [ALT]	Step 6: The system sets the status of the inventory item type to reactivated in the following table:

INTEGRATION NGO MANAGEMENT SYSTEM		
		• InventoryType [ALT]
		Step 7: The system saves the transaction to the AuditLog table.
		Step 8: The system displays a success toaster informing the super admin that the inventory item type was reactivated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects to cancel their reactivate inventory item type request. Use case terminates.	
	[ALT] Step 6: The system fails to update the status of the inventory item type in the relevant entities and displays an error toaster to the super admin. Use case terminates.	
Conclusion:	The system displays a success toaster informing the super admin that the inventory item type was reactivated successfully.	
Post-condition:	• The inventory item type is reactivated on the system. • The inventory item type can be used when creating a procurement.	
Business rules:	• Only super admins or users with the required permissions can reactivate an inventory item type.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 36: Reactivate Inventory Item Type - Logical Use Case Narrative

2.4 Group Subsystem Use Case Narratives

2.4.1 Create Group

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Create Group	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	4.1	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• Users	
Description:	This use case describes how a super admin creates a new group within a selected project. The system allows the user to enter the group's details, assign users, and then the system will validate the input, and confirm the creation before saving the group to the database. The group is then stored, linked to each user, and a confirmation toaster is displayed. The transaction is also recorded in the audit log.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to add a new group to the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to add a new group to the system.	**Step 2:** The system invokes UC 4.2 (View Groups) to display a pop-up showing all the groups' details.
	Step 3: The super admin selects to add a new group to the project.	**Step 4:** The system displays a Create Group modal to provide the following information:

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Group name • Members
	Step 5: The super admin enters the group's details, adds users and selects create. [ALT]	**Step 6:** The system validates the details entered. [ALT]
		Step 7: The system requests confirmation for the new group to be added.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the new group and the assigned users onto the following tables: • Group • UserGroup [ALT]
		Step 10: The system automatically creates a messaging channel for the group members to communicate.
		Step 11: The system records the transaction to the AuditLog table.
		Step 12: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and displays an error message within the Create Group modal. Go back to step 4.	
	[ALT] Step 8: The admin does not confirm the changes. Go back to step 4.	
	[ALT] Step 9: The system fails to save the new group and displays a toaster message. Terminate use case.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Conclusion:	The new group's details have been saved onto the system, and a confirmation toaster is shown.
Post-condition:	The system saves the new group onto the Group table and the assigned users onto the UserGroup table.
Business rules:	• Only super admin or users given appropriate permissions can add a new group.
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 37: Create Group - Logical Use Case Narrative

2.4.2 View Groups

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Groups	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	4.2	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes when a user wants to view the groups for a specific project on the system The system retrieves active projects along with their group count and displays them in a card using the Angular frontend. The user can select a project, and the system retrieves all related groups, including their details and assigned users through the relevant API endpoints and displayed in a pop-up modal. The groups are shown with their names and members.	
Pre-condition:	The user must be logged into the system.	
Trigger:	The user wants to view the groups for a specific project to which they are assigned.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user wants to view the groups for a specific project to which they are assigned.	
	Step 2: The user selects the team tab [ALT]	**Step 3:** The system retrieves the current projects and the number of groups associated with them on the system from the Project and Group table.

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 4: The system displays the projects and the number of groups they have based on user permissions. [ALT]
	Step 5: The user selects a project card.	**Step 6:** The system retrieves all the groups for that project, their details as well as the assigned users from the tables: • Group • UserGroup • User
		Step 7: The system displays a pop-up with all the details. These details include: • Group Names • Members [ALT]
	Step 8: The user can view the groups.	
Alternate courses:	**[ALT] Step 2:** User is on mobile and selects the menu icon. Go to Step 3.	
	[ALT] Step 4: The system does not find any active projects. Terminate use case.	
	[ALT] Step 7: The system does not find any active groups in the selected project. Terminate use case.	
Conclusion:	The system displays all the groups' details.	
Post-condition:	All the related groups' details are displayed.	
Business rules:	• Users can only view groups they are assigned to. • Admins and super admins can view all groups on the system.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	

INTEGRATED NGO MANAGEMENT SYSTEM	
Open issues:	• None

Table 38: View Groups - Logical Use Case Narrative

2.4.3 Update Group

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update Group	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	4.3	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a Super Admin updates an existing group within a selected project. The system retrieves and displays the current group details, including the group name and assigned users. The Super Admin can update the group name and manage group membership by adding or removing users. Once the changes are made, the system validates the updated information before updating it. The use case ends when the system successfully saves the updated group's details to the database.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to update a group's details on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to update a group's details on the system.	**Step 2:** The system invokes UC 4.2 (View Groups) to display a pop-up showing all the groups' details.
	Step 3: The super admin selects edit on a specific group.	**Step 4:** The system displays an Edit Group modal with the selected group's current information: • Group Name

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Members
	Step 5: The super admin updates a group's details and selects save. [ALT]	**Step 6:** The system validates the updated data. [ALT]
		Step 7: The system requests confirmation for the changes to be made to the group.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the changes to the following tables: • Group • UserGroup [ALT]
		Step 10: The system updates the messaging channel.
		Step 11: The system records the transaction to the AuditLog table.
		Step 12: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and displays an error message within the Edit Group modal. Go back to step 4.	
	[ALT] Step 8: The super admin does not confirm the changes. Go back to step 4.	
	[ALT] Step 9: The system fails to save the changes and displays a toaster message. Terminate use case.	
Conclusion:	The updated group's details have been saved onto the system, and a confirmation toaster is shown.	
Post-condition:	The system saves the changes to the Group and UserGroup tables.	
Business rules:	• Only super admin or users given appropriate permissions can edit a group.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 39: Update Group - Logical Use Case Narrative

2.4.4 Deactivate Group

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Deactivate Group	**USE CASE TYPE**
Use case id:	4.4	Business Requirements: □
Priority:	Medium	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a Super Admin deactivates an existing group within the system. The system displays the selected group and allows the Super Admin to select deactivate. The group is then soft deleted by changing the status to inactive, to ensure data integrity and historical tracking. The system will no longer show the group in the active group listings. The use case ends when the group has been successfully deactivated and is no longer visible as an active group within the system.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to deactivate a group on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to deactivate a group on the system.	**Step 2:** The system invokes UC 4.2 (View Groups) to display a pop-up showing all the groups' details.
	Step 3: The super admin selects deactivate on a specific group. [ALT]	**Step 4:** The system requests confirmation for the group deactivation.

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system sets the group's status to be inactive in the Group table. [ALT]
		Step 7: The system records the transaction to the AuditLog table.
		Step 8: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 3:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 5: The super admin does not confirm. Go back to step 2.	
	[ALT] Step 6: The system fails to deactivate the group and displays a toaster message. Terminate use case.	
Conclusion:	The group has been deactivated on the system, and a confirmation toaster is shown.	
Post-condition:	The system sets the group's status to inactive in the Group table and the group is no longer displayed in the active group listings.	
Business rules:	• Only super admin or users given appropriate permissions can deactivate a group.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 40: Deactivate Group - Logical Use Case Narrative

2.4.5 Create Skill

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Create Skill	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	4.5	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a Super Admin creates a new skill within the system. The system allows the user to enter the skill's name and description, validates the input, and requests confirmation before saving the skill. Once confirmed, the system sends the data to the backend API endpoint, where the skill is created in the database. The new skill is then displayed on the Skills page as a card, and a confirmation toaster is shown.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to add a new skill to the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to add a new skill to the system.	**Step 2:** The system invokes UC 4.6 (View Skills) to display all the current skills.
	Step 3: The super admin selects add skill.	**Step 4:** The system displays a Create Skill modal to provide the following information: • Skill • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin enters the new skill's details and selects create. [ALT]	**Step 6:** The system validates the details entered. [ALT]
		Step 7: The system requests confirmation for the skill to be added.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the new skill onto the Skills table. [ALT]
		Step 10: The system records the transaction to the AuditLog table.
		Step 11: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and displays an error message within the Create Skill modal. Go back to step 4.	
	[ALT] Step 8: The super admin does not confirm the changes. Go back to step 4.	
	[ALT] Step 9: The system fails to save new skill and displays a toaster message. Terminate use case.	
Conclusion:	The new skill's details have been saved onto the system, and a confirmation toaster is shown.	
Post-condition:	The system saves the new skill onto the Skills table.	
Business rules:	• Only super admin or users given appropriate permissions can add a new skill.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	

INTEGRATED NGO MANAGEMENT SYSTEM	
Open issues:	• None

Table 41: Create Skill - Logical Use Case Narrative

2.4.6 View Skills

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Skills	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	4.6	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes when an admin views the details of all the skill within the system. The system displays the skills as cards containing the skill's name and description. The use case ends when the admin sees all the skills displays as cards.	
Pre-condition:	The admin must be logged into the system.	
Trigger:	The admin wants to view the skills on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The admin wants to view the skills on the system.	
	Step 2: The admin hovers the admin tab and then selects skills. [ALT]	**Step 3:** The system retrieves all the current skills on the system from the Skills table.
		Step 4: The system displays the skills page showing all the current skills as cards: • Skill • Description [ALT]
	Step 5: The admin can view the skills.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Alternate courses:	**[ALT] Step 2:** User is on mobile and selects the menu icon. Go to step 3.
	[ALT] Step 4: The system does not find any active skills. Terminate use case.
Conclusion:	All skills are displayed in cards.
Post-condition:	The system displays all the current skills in cards.
Business rules:	• Only super admins or users given appropriate permissions can view the skills.
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 42: View Skills - Logical Use Case Narrative

2.4.7 View Skill

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Skill	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	4.7	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes when an admin views the details of a selected skill within the system. The admin will select a skill card, and it will open a modal. From this modal, the user can view full skill information and access actions such as editing or deactivating the skill. The use case ends when the admin can view the skill's details.	
Pre-condition:	The admin must be logged into the system.	
Trigger:	The admin wants to view the specific skill's details on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The admin wants to view the specific skill's details on the system.	**Step 2:** The system invokes UC 4.6 (View Skills) to display all the current skills.
	Step 3: The admin selects a skills card.	**Step 4:** The system retrieves the selected skill details on the system from the Skills table.
		Step 5: The system displays a View Skill modal showing the skill details: • Skill • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 6: The admin can view the skill.	
Alternate courses:	None	
Conclusion:	The system displays a modal showing the skill's details.	
Post-condition:	The skill's details are displayed.	
Business rules:	• Only super admins or users given appropriate permissions can view a skill.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 43: View Skill - Logical Use Case Narrative

2.4.8 Update Skill

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update Skill	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	4.8	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a Super Admin updates an existing skill in the system. The system allows the user to modify the skill's name and description on an edit modal. Once the user confirms the changes, the system validates the input and updates the skill in the database. The updated skill information is then reflected on the skills page.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to update a skill's details on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to update a skill's details on the system.	**Step 2:** The system invokes UC 4.7 (View Skill) to display a pop-up showing the skill's details.
	Step 3: The super admin selects edit.	**Step 4:** The system displays the Edit Skill modal with the selected skill's current information: • Skill • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin updates the skill's details and selects save. [ALT]	**Step 6:** The system validates the details entered. [ALT]
		Step 7: The system requests confirmation for the changes to be made to the skill.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the updated skill's details onto the Skills table. [ALT]
		Step 10: The system records the transaction to the AuditLog table.
		Step 11: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and displays an error message within the Edit Skill modal. Go back to step 4.	
	[ALT] Step 8: The super admin does not confirm. Go back to step 4.	
	[ALT] Step 9: The system fails to save the changes and displays a toaster message. Terminate use case.	
Conclusion:	The updated skill's details have been saved onto the system, and a confirmation toaster is shown.	
Post-condition:	The system saves the updated skill's details onto the Skills table.	
Business rules:	• Only super admin or users given appropriate permissions can update a skill.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 44: Update Skill - Logical Use Case Narrative

2.4.9 Deactivate Skill

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Deactivate Skill	**USE CASE TYPE**
Use case id:	4.9	Business Requirements: □
Priority:	Low	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a Super Admin deactivates an existing skill in the system. The system allows the Super Admin to select deactivate on a skill and will then confirm the deactivation. Once confirmed, the system updates the skill's status to inactive in the database, which will remove it from the active skills list. The use case ends when the skill is no longer visible within the active skill listings.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to deactivate a skill on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to deactivate a skill on the system.	**Step 2:** The system invokes UC 4.7 (View Skill) to display a pop-up showing the skill's details.
	Step 3: The super admin selects deactivate. [ALT]	**Step 4:** The system requests confirmation for the skill deactivation.
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system updates the skill's status to be inactive in the Skills table. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system records the transaction to the AuditLog table.
		Step 8: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 3:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 5: The super admin does not confirm. Go back to step 2.	
	[ALT] Step 6: The system fails to deactivate the skill and displays a toaster message. Terminate use case.	
Conclusion:	The active skill has been deactivated, and a confirmation toaster is shown.	
Post-condition:	The system updates the skill's status to inactive in the Skill table.	
Business rules:	• Only super admin or users given appropriate permissions can deactivate a skill.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 45: Deactivate Skill - Logical Use Case Narrative

2.5 Warehouse Subsystem Use Case Narratives

2.5.1 Create Warehouse

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Create Warehouse	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	5.1	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to register a new warehouse within the system, capturing its identifying and operational details such as warehouse name, physical location, storage capacity, and status. The warehouse serves as a storage entity for inventory items and is critical for tracking stock distribution and allocation.	
Pre-condition:	• Admin is authenticated to manage warehouse • Warehouse does not already exist	
Trigger:	Admin selects “Create Warehouse”	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin initiates warehouse creation.	**Step 2:** System displays warehouse creation form.
	Step 3: Admin captures warehouse details (name, location, capacity, description) [ALT]	**Step 4:** System validates required fields and its uniqueness.
	Step 5: Admin confirms submission.	**Step 6:** System creates the warehouse and confirms successful creation.

INTEGRATED NGO MANAGEMENT SYSTEM	
Alternate courses:	**[ALT] Step 3:** System highlights errors and prevents submission due to missing required fields.
Conclusion:	Warehouse successfully created and is available for use.
Post-condition:	Warehouse stored in system
Business rules:	• Warehouse must be unique per location
Implementation constraints and specifications:	Must integrate with inventory subsystem
Assumptions:	• Admin provides accurate warehouse details.
Open issues:	• None

Table 46: Create Warehouse - Logical Use Case Narrative

2.5.2 View Warehouse

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	View Warehouse	**USE CASE TYPE**
Use case id:	5.2	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to retrieve and view details of a specific warehouse, including its location, operational status, and associated inventory (if integrated).	
Pre-condition:	• Admin is authenticated to manage warehouse • Warehouse should exist	
Trigger:	Admin selects "View Warehouse"	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects the warehouse.	**Step 2:** System retrieves warehouse details [ALT]
		Step 3: System compiles warehouse data.
		Step 4: System displays warehouse information.
Alternate courses:	**[ALT] Step 2:** System does not find the warehouse and displays error message.	
Conclusion:	Warehouse information successfully displayed.	
Post-condition:	No data modified	

INTEGRATED NGO MANAGEMENT SYSTEM	
Business rules:	• Only authorised users may view warehouse data.
Implementation constraints and specifications:	None
Assumptions:	• Warehouse data is up to date.
Open issues:	• None

Table 47: View Warehouse - Logical Use Case Narrative

2.5.3 Update Warehouse

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Update Warehouse	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	5.3	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to modify warehouse details such as name, location, or capacity to reflect operational changes.	
Pre-condition:	• Admin is authenticated to manage warehouse • Warehouse should exist	
Trigger:	Admin selects "Update Warehouse"	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects the warehouse.	**Step 2:** System retrieves warehouse details
		Step 3: System displays editable warehouse data.
	Step 4: Admin updates details.	**Step 5:** System validates updates.
	Step 6: Admin confirms details.	**Step 7:** System updates warehouse and confirms success.
Alternate courses:	None	
Conclusion:	Warehouse successfully updates	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	Updates changes stored
Business rules:	• Only authorised users may update warehouse data.
Implementation constraints and specifications:	None
Assumptions:	• None
Open issues:	• None

Table 48: Update Warehouse - Logical Use Case Narrative

2.5.4 Reactivate Warehouse

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Reactivate Warehouse	**USE CASE TYPE**
Use case id:	5.4	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to restore a previously deactivated warehouse, making it available again for inventory storage and operations.	
Pre-condition:	• Admin is authenticated to manage warehouse • Warehouse should exist and is inactive	
Trigger:	Admin selects "Activate Warehouse"	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects the warehouse.	**Step 2:** System retrieves warehouse details
	Step 3: Admin confirms the activation of the warehouse.	**Step 4:** System updates status to active. .
		Step 5: The System logs the actions and confirms success.
Alternate courses:	None	
Conclusion:	Warehouse successfully reactivated.	
Post-condition:	Warehouse is now available for operations	
Business rules:	• Only inactive warehouses can be reactivated	

INTEGRATED NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	None
Assumptions:	• None
Open issues:	• None

Table 49: Reactivate Warehouse -Logical Use Case Narrative

2.5.5 Deactivate Warehouse

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 22 March 2026** **Version: 1.0**
Use case name:	Deactivate Warehouse	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	5.5	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• System	
Other participating actors:	• None	
Other interested stakeholders:	User	
Description:	This use case allows the Admin to deactivate a warehouse, preventing it from being used for inventory storage while preserving historical data.	
Pre-condition:	• Admin is authenticated to manage warehouse • Warehouse should exist	
Trigger:	Admin selects "Deactivate Warehouse"	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Admin selects the warehouse.	**Step 2:** System checks for existing inventory in the warehouse.
		Step 3: System displays warning if inventory exists.
	Step 4: Admin confirms deactivation.	**Step 5:** System updates warehouse status to inactive.
		Step 6: System logs action and confirms success.
Alternate courses:	None	
Conclusion:	Warehouse successfully deactivated	
Post-condition:	Warehouse is no longer available for allocation	

INTEGRATED NGO MANAGEMENT SYSTEM	
Business rules:	• Only authorised users may deactivate warehouses.
Implementation constraints and specifications:	None
Assumptions:	• None
Open issues:	• None

Table 50: Deactivate Warehouse - Logical Use Case Narrative

2.5.6 Inventory Stock Take

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Inventory Stock Take	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	5.6	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which the super admin performs an inventory stock take. The system will allow the super admin to select a warehouse and view its inventory. The super admin will perform a stock take and enter the stock count. The system will determine the difference and capture the difference to adjust the inventory. The use case ends once inventory has been adjusted and the stock take is completed.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin wants to perform an inventory stock take.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super admin wants to perform an inventory stock take.	
	Step 2: The super admin will navigate to the inventory tab and select stock take.	**Step 3:** The system retrieves all the warehouses from the Warehouse table.
		Step 4: The system displays the "Stock Take" form.

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin selects a warehouse.	**Step 6:** The system retrieves all the inventory items in the selected warehouse using the InventoryTransaction table.
	Step 7: The super admin enters the physical stock count. [ALT]	**Step 8:** The system determines the difference and saves the results and adjusts the inventory in the following tables: • StockTake • StockTakeDetails • InventoryTransactions [ALT]
		Step 9: The system saves the transaction to the AuditLog table.
		Step 10: The system displays a message informing the admin that the inventory has been adjusted successfully.
Alternate courses:	**[ALT] Step 7:** The super admin selects cancel. Use case terminates	
	[ALT] Step 8: The system fails to add the stock take into the relevant entities and displays an error message to the admin. Use case terminates.	
Conclusion:	The system displays a message informing the admin that the inventory has been adjusted successfully.	
Post-condition:	The inventory is adjusted and the stock take is complete is removed.	
Business rules:	• Only a super admin can perform an inventory stock take	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	

INTEGRATED NGO MANAGEMENT SYSTEM	
Open issues:	• None

Table 51: Inventory Stock Take - Logical Use Case Narrative

2.6 Vendor Subsystem Use Case Narratives

2.6.1 Create Vendor

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1.0**
Use case name:	Create Vendor	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.1	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin creates a vendor on the system. The super admin will select to add a vendor and enter the vendor's information. The system will validate the details inputted before informing the super admin that the vendor has been successfully added. The use case ends once the vendor has successfully been added to the system.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin requests to create a vendor.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to create a vendor.	Step 2: The system invokes use case 6.3 (View Vendors)
	Step 3: The super admin selects to add a vendor.	Step 4: The system will request the following information: • Name • Vendor Type • Email • Phone

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Contact Person • Credit Facility • Address • Rating
	Step 5: The super admin will enter the vendor's details and select to add the vendor. [ALT]	Step 6: The system validates that all required fields are completed. [ALT]
		Step 7: The system requests confirmation for the vendor to be added.
	Step 8: The super admin confirms that the vendor can be added. [ALT]	Step 9: The system adds the vendor to the following tables: • Vendor [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a success toaster informing the super admin that the vendor has been added successfully.
Alternate courses:	**[ALT] Step 5:** The super admin cancels their request to create a vendor. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 8: The super admin does not provide confirmation. Go back to step 5.	
	[ALT] Step 9: The system fails to add the vendor into the relevant entities and displays an error toaster to the super admin. Use case terminates.	
Conclusion:	The system displays a success toaster informing the super admin that the vendor has been created successfully.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Post-condition:	The new vendor is successfully captured on the system and can be used when creating a procurement or management within the system.
Business rules:	• Only a super admin or users with the required permissions can create a vendor.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 52: Create Vendor - Logical Use Case Narrative

2.6.2 View Vendor

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1.0**
Use case name:	View Vendor	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.2	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• Admin	
Description:	This use case describes the event in which a super admin wants to view a vendor on the system. The system will retrieve and display all the vendors on the system before the super admin filters the vendor's name and selects a vendor to view. The system will retrieve the selected vendor's details and display them to the super admin. The use case ends once the system displays the selected vendor's details.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin requests to view a vendor.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to view a vendor.	Step 2: The system will invoke use case 6.3 (View Vendors).
	Step 3: The super admin filters the vendors list by active status or by searching by name, email or contact. [ALT]	Step 4: The system filters and displays the vendor matching the criteria. [ALT]
	Step 5: The super admin selects the vendor.	Step 6: The system retrieves the selected vendor's details from the following tables:

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Vendor • VendorType [ALT]
		Step 7: The system displays the vendor's details, displaying the following information: • Name • Type • Email • Phone • Contact Person • Credit Facility • Address • Rating • Active Status
Alternate courses:	**[ALT] Step 3:** The super admin does not filter and selects a displayed vendor.	
	[ALT] Step 4: The system fails to find any vendors matching the filter criteria. Go back to step 3.	
	[ALT] Step 5: The system fails to retrieve the selected vendor's details from the relevant entities and displays an error toaster.	
Conclusion:	Super admin can view a vendor's details.	
Post-condition:	The vendor's details are displayed.	
Business rules:	• Super admins or users with the required permissions can view a vendor.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 53: View Vendor - Logical Use Case Narrative

2.6.3 View Vendors

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1.0**
Use case name:	View Vendors	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.3	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin wants to view all the vendors on the system. The super admin will navigate to the Admin tab and select Vendors. The system will retrieve all the vendors on the system and display them in a list to the super admin. The use case ends once all the vendors are displayed.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin requests to view all the vendors.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to view all the vendors.	
	Step 2: The super admin will navigate to the Admin tab and select Vendors.	Step 3: The system retrieves all the vendors on the system from the following tables: • Vendor • VendorType [ALT]
		Step 5: The system displays the vendors in a list displaying the following details for each vendor:

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Name • Type • Email • Phone • Contact • Rating • Status
Alternate courses:	**[ALT] Step 3:** The system fails the retrieve the vendors from the relevant entities and displays an error toaster.	
Conclusion:	The system displays the vendors for the super admin to view.	
Post-condition:	The vendor's details are displayed.	
Business rules:	• Super admins or users with the required permission can view vendors.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 54: View Vendors - Logical Use Case Narrative

2.6.4 Edit Vendor

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1.0**
Use case name:	Edit Vendor	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.4	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin wants to update a vendor on the system. The system allows the super admin to select a vendor and then update the vendor's details. The system will validate the details entered before asking the super admin to confirm the updated details. Once confirmed the system will display a success toaster informing the super admin that the vendor's details have been successfully updated. The use case ends once the vendor's updated details have been successfully captured on the system.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin requests to edit a vendor.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to edit a vendor.	Step 2: The system will invoke use case 6.3 (View Vendors).
	Step 3: The super admin will select to update a vendor.	Step 4: The system displays the vendor's current details in editable fields including: • Name • Vendor Type • Email

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Phone • Contact Person • Credit Facility • Address • Rating
	Step 5: The super admin enters the updated details. [ALT]	Step 6: The system validates that all required fields are completed. [ALT]
		Step 7: The system requests confirmation for the vendor to be updated.
	Step 8: The super admin confirms that the vendor can be updated. [ALT]	Step 9: The system saves the updated details in the following table: • Vendor [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a success toaster informing the super admin that the vendor was updated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin cancels their request to edit a vendor. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 8: The super admin selects to go back. Go back to step 5.	
	[ALT] Step 9: The system fails to update the vendor in the relevant entities and displays an error toaster to the super admin. Use case terminates.	
Conclusion:	The system displays a success toaster informing the super admin that the vendor was updated successfully.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Post-condition:	The vendor's updated details are successfully captured on the system.
Business rules:	• Super admins or users with the required permissions can edit a vendor.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 55: Edit Vendor - Logical Use Case Narrative

2.6.5 Reactivate Vendor

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1.0**
Use case name:	Reactivate Vendor	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.5	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which the super admin reactivates a vendor on the system. The system will display the vendors and allow the super admin to select a vendor to reactivate. The system will ask the super admin to confirm the reactivation before the vendor is reactivated. The use case ends once the vendor is reactivated on the system.	
Pre-condition:	Super admin is logged in. Vendor is inactive.	
Trigger:	Super admin requests to reactivate a vendor.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to reactivate a vendor.	Step 2: The system will invoke use case 6.3 (View Vendors).
	Step 3: The super admin selects to reactivate a vendor.	Step 4: The system requests confirmation for the vendor to be reactivated.
	Step 5: The super admin confirms that the vendor can be reactivated. [ALT]	Step 6: The system updates the status of the vendor to active in the Vendor table. [ALT]

INTEGRATION NGO MANAGEMENT SYSTEM		
		Step 7: The system saves the transaction to the AuditLog table.
		Step 8: The system displays a success toaster informing the super admin that the vendor was reactivated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin cancels their request to reactivate a vendor. Use case terminates.	
	[ALT] Step 6: The system fails to update the status of the vendor in the relevant entities and displays an error toaster to the super admin. Use case terminates.	
Conclusion:	The system displays a success toaster informing the super admin that the vendor was reactivated successfully.	
Post-condition:	The vendor is successfully reactivated on the system and can be used when creating a procurement or management within the system.	
Business rules:	• Super admins or users with the required permission can reactivate a vendor.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 56: Reactivate Vendor - Logical Use Case Narrative

2.6.6 Deactivate Vendor

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 8 August 2026** **Version: 1.0**
Use case name:	Deactivate Vendor	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.6	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin wants to deactivate a vendor on the system. The system will display the vendor and allow the super admin to select a vendor to deactivate. The system will ask the super admin to confirm the deactivation before the vendor is deactivated. The use case ends once the vendor has been successfully deactivated on the system.	
Pre-condition:	Super admin is logged in. Vendor is active.	
Trigger:	Super admin requests to deactivate a vendor.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to deactivate a vendor.	Step 2: The system invokes use case 6.3 (View Vendors).
	Step 3: The super admin selects to deactivate a vendor.	Step 4: The system requests confirmation for the vendor to be deactivated.
	Step 5: The super admin confirms that the vendor can be deactivated. [ALT]	Step 6: The system sets the status of the vendor to deactivated in the Vendor table. [ALT]

INTEGRATION NGO MANAGEMENT SYSTEM		
		Step 7: The system saves the transaction to the AuditLog table.
		Step 8: The system displays a message informing the super admin that the vendor was deactivated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin cancels their request to deactivate a vendor. Use case terminates.	
	[ALT] Step 6: The system fails to update the status of the vendor in the relevant entities and displays an error toaster to the super admin. Use case terminates.	
Conclusion:	The system displays a success toaster informing the super admin that the vendor was deactivated successfully.	
Post-condition:	The vendor is successfully deactivated on the system and can no longer be used when creating a procurement or management within the system.	
Business rules:	• Super admins or users with the required permission can deactivate a vendor.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 57: Deactivate Vendor - Logical Use Case Narrative

2.6.7 Create Vendor Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Create Vendor Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.7	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event of when a super admin wants to add a new vendor type to the system. The system prompts the super admin to enter the new vendor type's details. The use case ends when the system saves the new vendor type onto the system.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to add a new vendor type to the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to add a new vendor type to the system.	**Step 2:** The system invokes UC 6.9 (View Vendor Types) to display a page with a table showing all the current types.
	Step 3: The super admin selects to add a vendor type.	**Step 4:** The system displays a Create Vendor Type modal to provide the following information: • Name • Description
	Step 5: The super admin enters the new vendor	**Step 6:** The system validates the details entered. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
	type's details and selects create. [ALT]	
		Step 7: The system saves the new type onto the VendorType table. [ALT]
		Step 8: The system records the transaction to the AuditLog table.
		Step 9: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and an error message is displayed within the Create Vendor Type modal Go back to step 4.	
	[ALT] Step 7: The system fails to save the new vendor type and displays a toaster message. Terminate use case.	
Conclusion:	The new vendor type has been saved onto the system, and a confirmation toaster is shown.	
Post-condition:	The system saves the new vendor type in the VendorType table and makes the newly created type available for assignment to vendors and for management within the system.	
Business rules:	• Only super admin or users given appropriate permissions can add a new vendor type.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 58: Create Vendor Type - Logical Use Case Narrative

2.6.8 Update Vendor Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update Vendor Types	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.8	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event of when a super admin wants to update an existing vendor type on the system The system will display the selected vendor type's details and the super admin will be able to modify the vendor type's information. The use case ends when the system saves the updated vendor type's details onto the system.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to update a vendor type's details on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to update a vendor type's details on the system.	**Step 2:** The system invokes UC 6.9 (View Vendor Types) to display a page with a table showing all the current types.
	Step 3: The super admin selects update on a specific type.	**Step 4:** The system displays the Edit Vendor Type modal with the following information: • Name • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin updates the vendor type's details and selects save. [ALT]	**Step 6:** The system validates the details entered. [ALT]
		Step 7: The system saves the updated vendor type's details onto the VendorType table. [ALT]
		Step 8: The system records the transaction to the AuditLog table.
		Step 9: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and an error message is displayed within the Edit Vendor Type modal Go back to step 4.	
	[ALT] Step 7: The system fails to save the changes and displays a toaster message. Terminate use case.	
Conclusion:	The updated vendor type details have been saved onto the system, and a confirmation toaster is shown.	
Post-condition:	The system saves the updated vendor type details in the VendorType table, and the modified type is available for viewing, assignment, and management within the system.	
Business rules:	• Only super admin or users given appropriate permissions can update a vendor type.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 59: Update Vendor Types - Logical Use Case Narrative

2.6.9 View Vendor Types

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Vendor Types	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.9	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event of when an admin wants to view the vendor types on the system. The system will display a list of vendor types and their details in a table. The use case ends when the admin can see the vendor types and their details on the system.	
Pre-condition:	The admin must be logged into the system.	
Trigger:	The admin wants to view the vendor types on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The admin wants to view the vendor types on the system.	**Step 2:** The system invokes UC 6.3 (View Vendors) to display the list of vendors.
		Step 3: The system retrieves the vendor types from the VendorType table.
		Step 5: The system displays a table showing all the current vendor types. The table includes: • Name • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Status • Actions
	Step 6: The admin can view the vendor types.	
Alternate courses:	None	
Conclusion:	The system displays a screen with a table showing all the current vendor types.	
Post-condition:	The system retrieves and displays the vendor types stored in the VendorType table, allowing authorized users to view, search, and filter the available types within the system.	
Business rules:	• Only super admin or users given appropriate permissions can view vendor types.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 60: View Vendor Types - Logical Use Case Narrative

2.6.10 Reactivate Vendor Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Reactivate Vendor Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.10	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event of when a super admin wants to reactivate a vendor type on the system. The system displays the inactive vendor types and allows the super admin to select a vendor type to reactivate. The use case ends when the vendor type has been successfully reactivated.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to reactivate a vendor type on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to reactivate a vendor type on the system.	**Step 2:** The system invokes UC 6.9 (View Vendor Types) to display a page with a table showing all the current types.
	Step 3: The super admin selects reactivate on a specific vendor type where the status is inactive.	**Step 4:** The system requests confirmation for the vendor type reactivation.
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system updates the vendor type's status to active in the VendorType table. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system records the transaction to the AuditLog table.
		Step 8: The system displays a confirmation toaster.
Alternate courses:	ALT-Step 5: The super admin does not confirm. Go back to step 2.	
	ALT-Step 6: The system fails to reactivate the vendor type and displays a toaster message. Terminate use case.	
Conclusion:	The inactive vendor type has been reactivated, and a confirmation toaster is shown.	
Post-condition:	The system updates the vendor type's status to active in the VendorType table, making the type available for viewing, assignment to vendors, and management within the system.	
Business rules:	• Only super admin or users given appropriate permissions can reactivate vendor type.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 61: Reactivate Vendor Type - Logical Use Case Narrative

2.6.11 Deactivate Vendor Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Deactivate Vendor Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	6.11	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event of when a super admin wants to deactivate a vendor type on the system. The system displays the active vendor types and allows the super admin to select a vendor type to deactivate. The use case ends when the vendor type has been successfully deactivated.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to deactivate a vendor type on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to deactivate a vendor type on the system.	**Step 2:** The system invokes UC 6.9 (View Vendor Types) to display a page with a table showing all the current types.
	Step 3: The super admin selects deactivate on a specific vendor type where the status is active.	**Step 4:** The system requests confirmation for the vendor type deactivation.
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system updates the vendor type's status to inactive in the VendorType table. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system records the transaction to the AuditLog table.
		Step 8: The system displays a confirmation toaster.
Alternate courses:	ALT-Step 5: The super admin does not confirm. Go back to step 2.	
	ALT-Step 6: The system fails to deactivate the vendor type and displays a toaster message. Terminate use case.	
Conclusion:	The active vendor type has been deactivated, and a confirmation toaster is shown.	
Post-condition:	The system updates the vendor type's status to inactive in the VendorType table, making the type unavailable in the active listings while remaining available for viewing and management when inactive types are displayed.	
Business rules:	• Only super admin or users given appropriate permissions can reactivate vendor type.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 62 :Deactivate Vendor Type - Logical Use Case Narrative

2.7 Project Management Subsystem Use Case Narratives

2.7.1 Create Project

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Create Project	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.1	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super-Admin	
Primary system actor (PSA):	• Super-Admin	
Other participating actors:	• None	
Other interested stakeholders:	Users who have been added to the new project	
Description:	A Super-Admin will create a new project that will represent a new NGO goal. The system will open a form where the Super-Admin will input the project: name, dates, staff, teams, milestones and deadlines to help plan and organise the project. The system will capture all the information and alert all the relevant staff and create the schedule.	
Pre-condition:	• The staff have all been previously added to the system. • The staff roles have all been previously added to the system. • The project type has been added to the system previously.	
Trigger:	The super admin clicks on “Create new project” in the Projects nav bar or from the home widgets. A trigger would also be that the NGO wants to create a new segment for a new type of community service. Or they want to distinguish between events in different years, so they create a new project.	
Typical Course of Events:	**Actor Action**	**System Response**

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 1: The super admin will click "create new project."	**Step 2:** The system will load the form displaying all the project: name, dates, staff, teams, milestones and deadlines for the admin to complete.
	Step 3: The super admin will complete the form, adding all locations, staff and other import details to the form. [ALT] [ALT] [ALT]	
		Step 4: The system will verify all the fields are entered correctly and then save the new project and its details into the system. [ALT]
		Step 5: The system will automatically alert all staff members involved and add the project deadlines and details to be easily viewed.
	Step 6: The super admin receives confirmation that the project has been captured.	
Alternate courses:	[ALT] Step 3a: The staff member is not on the system yet: the admin will click "Add user" and the pop-up for the use case "Add user" will appear so the admin can quickly add them and then return to the form.	
	[ALT] Step 3b: The project type is not on the system yet: the admin will click "Add project type" and the pop-up for the use case "Add project type" will appear so the admin can quickly add them and then return to the form.	

INTEGRATED NGO MANAGEMENT SYSTEM	
	[ALT] Step 3c: The role is not on the system yet: the admin will click "Add role" and the pop-up for the use case "Add role" will appear so the admin can quickly add them and then return to the form.
	[ALT] Step 4: The fields have not been filled in correctly: The system will show the admin which field can't be verified, along with the reason, so they can easily try again.
Conclusion:	The project, along with its details, have been successfully added to the system.
Post-condition:	The new project is added to the list, and all staff members involved with the project have been sent details regarding the project's details and deadlines.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• The super admin is the person organising the project
Open issues:	• None

Table 63: Create Project - Logical Use Case Narrative

2.7.2 View Project

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	View Project	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.2	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• User	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	A user clicks on a specific project to see all its details, like the name, staff, dates, project type, milestones, and deadline. They see this in the project's nav bar option.	
Pre-condition:	• The user must already have a pre-existing account on the system • The user is logged in	
Trigger:	The user wants to see a specific project full detail	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user clicks on a specific project from a list of projects on the "Projects" page. [ALT]	**Step 2:** The system retrieves the details of the project they selected, and it will display all the details in sections, relative to the user's role. [ALT]
	Step 3: The user can find the specific detail they need or just view the project details.	
Alternate courses:	**[ALT] Step 1:** The project has not yet been created; the user will select "Create New Project".	
	[ALT] Step 2:	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The system cannot display the details of the event correctly; the system will display an error message asking the user to "Try again later"
Conclusion:	The user can view all the project details for a specific project, broken down into categories.
Post-condition:	The project is displayed along will all other projects.
Business rules:	• Secure Access • Role Based Access • Non admin users cannot access budgeting information
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.

Table 64: View Project - Logical Use Case Narrative

2.7.3 Update Project

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Update Project	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.3	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super-Admin	
Primary system actor (PSA):	• Super-Admin	
Other participating actors:	• Users involved in the project being edited	
Other interested stakeholders:	Administrator	
Description:	A Super-Admin will edit an existing project already saved in the system. The system will open the project's form where the Super-Admin will be able to edit the project: name, dates, staff, teams, milestones and deadlines to help plan and organise the project. The system will capture all the information and alert all the relevant staff and update the schedule.	
Pre-condition:	The project has already been created The staff have all been previously added to the system. The staff roles have all been previously added to the system. The project type has been added to the system previously.	
Trigger:	The super admin wants to change or update details regarding the project.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin will click on the project they wish to update or edit [ALT]	**Step 2:** The system will load the form displaying all the project: name, dates, staff, teams, milestones and deadlines for the admin to complete. [ALT]
	Step 3: The super admin will clicks "edit project".	**Step 4:** The system will load the form displaying all the

INTEGRATED NGO MANAGEMENT SYSTEM		
		project: name, dates, staff, teams, milestones and deadlines and all other information as editable fields so the user can update what they need to.
	Step 5: The super admin will select the field they would like to edit and correct the projects details where needed. This can be: adding staff, changing the project name or type , changing the date or location or several other factors. [ALT] [ALT] [ALT]	**Step 6:** The system will verify all the fields are entered correctly and then save the project updates to the system. [ALT]
		Step 7: The system will automatically alert all staff members involved and add the project deadlines and details to be easily viewed.
Alternate courses:	**[ALT] Step 1:** The project has not yet been created; the super admin must click “Create new project”	
	[ALT] Step 2: The project details have errors displaying, the system will display a message saying: “error, please restore the system”	
	[ALT] Step 5a: The staff member is not on the system yet: the admin will click “Add user” and the pop-up for the use case “Add user” will appear so the admin can quickly add them and then return to the form.	
	[ALT] Step 5b: The project type is not on the system yet: the admin will click “Add project type” and the pop-up for the use case “Add project type” will appear so the admin can quickly add them and then return to the form.	

INTEGRATED NGO MANAGEMENT SYSTEM	
	[ALT] Step 5c: The role is not on the system yet: the admin will click “Add role” and the pop-up for the use case “Add role” will appear so the admin can quickly add them and then return to the form.
	[ALT] Step 6: The fields have not been filled in correctly: The system will show the admin which field can’t be verified, along with the reason, so they can easily try again.
Conclusion:	The project has been successfully updated to meet the new project objectives.
Post-condition:	The schedules of all users involved has been updated.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 65: Update Project - Logical Use Case Narrative

2.7.4 Deactivate Project

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Deactivate Project	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.4	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	•	
Other interested stakeholders:	Administrator	
Description:	A super admin can deactivate projects after they are completed or if the project no longer supports the NPO's organisational goals.	
Pre-condition:	The project is already created and saved on the system	
Trigger:	The super admin wants to remove a project so users can no longer see it for any reason.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin will click on the project they wish to deactivate [ALT]	**Step 2:** The system will load the form displaying all the project: name, dates, staff, teams, milestones and deadlines of the project they wish to deactivate. [ALT]
	Step 3: The user will select "Deactivate project"	**Step 4:** The system will ask the user: "Are you

INTEGRATED NGO MANAGEMENT SYSTEM		
		sure you wish to deactivate" and then it will display the project name for confirmation.
	Step 5: Upon The user will confirm the wish to deactivate the project. [ALT]	**Step 6:** U The system will deactivate the project and show the project name and "has been deactivated". [ALT]
Alternate courses:	**[ALT] Step 1:** The project has not yet been created; the super admin must click "Create new project"	
	[ALT] Step 2: The project details have errors displaying, the system will display a message saying: "error, please restore the system"	
	[ALT] Step 5: The admin doesn't want to deactivate this project and clicks "Cancel"	
	[ALT] Step 6: The project has an error deactivating, and the system shows an error message to the admin to inform them their deactivation has been unsuccessful	
Conclusion:	The project has been deactivated from the system.	
Post-condition:	The admin has been notified of the project's status.	
Business rules:	• Secure Access • Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None	
Open issues:	• None	

Table 66: Deactivate Project - Logical Use Case Narrative

2.7.5 Create Project Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Create Project Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.5	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin would like to create a new category for projects that will represent a type of project.	
Pre-condition:	• The user must already have a pre-existing account on the system • The user has super admin privileges	
Trigger:	The super admin would like to create a new category for projects.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: After use case 7.6 (View Project type) the super admin will click the "Create project type "	**Step 2:** The system will display a form with the fields to add a new project type name.
	Step 3: The super admin will type the name of the new project type and click save. [ALT]	**Step 4:** The system will verify that the information has been filled in correctly and will then save it to the system.
		Step 5: Upon The system will notify the super admin that the new project type has been successfully captured . [ALT]
Alternate courses:	**[ALT] Step 3:**	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The user will click cancel if they don't want to add a new type
	[ALT] Step 5a: The system will tell the admin that there is already a role with that name
	[ALT] Step 5b: The system will tell the admin that their project type has not been successfully captured
Conclusion:	The new project type has been added to the system.
Post-condition:	The new project type appears in the list of project types.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 67: Create Project Type - Logical Use Case Narrative

2.7.6 View Project Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	View Project Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.6	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin can click "View Project types" to be able to see, edit, add or deactivate project types to better categorise the projects being completed.	
Pre-condition:	• The user must already have a pre-existing account on the system • Ther user must have super admin privileges	
Trigger:	The super admin wants to see all project types.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks on "View project types" function on the projects page.	**Step 2:** The system retrieves the recorded project types and displays them in a table for the super admin to view. They can then click "update" or "deactivate" on existing project types or "Create new project type" [ALT]
Alternate courses:	**[ALT] Step 2:** There is an error retrieving the project types and the system displays an error message to the admin	
Conclusion:	The project types are displayed in a table for the admin to easily access or view.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	Each project type has 2 functions: “Update”, “Deactivate” and there is a “create” project type function.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 68: View Project Type - Logical Use Case Narrative

2.7.7 Update Project Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Update Project Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.7	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Users	
Description:	A super admin wants to change the name of a project type. This is so users can better understand what those types of projects represent.	
Pre-condition:	The project type has already been added and captured onto the system	
Trigger:	A super admin wants to change the name of a project type.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks the "Update" function on the table displays, next to the project type they with to update.	**Step 2:** The system retrieves the recorded project type and displays the name as a field so the admin can edit it. [ALT]
	Step 3: The admin types the name of the updated project type. [ALT]	**Step 4:** The system captured the update and notifies the admin of the successful update. [ALT]
Alternate courses:	**[ALT] Step 2:** There is an error retrieving the project types, so the systems display an error message and suggested restoring the system.	
	[ALT] Step 3: The admin clicks the "cancel" function	

INTEGRATED NGO MANAGEMENT SYSTEM	
	[ALT] Step 4: The update couldn't be capture so the systems displays an error message
Conclusion:	The project type details are updated
Post-condition:	The system displays the updated project type.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• The super admin has the correct updated name of the project type
Open issues:	• None

Table 69: Update Project Type - Logical Use Case Narrative

2.7.8 Deactivate Project Type

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Deactivate Project Type	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.8	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin deactivates a project type, and it will remove it from the displayed table of project types, this would be because a project type is no longer being used	
Pre-condition:	The project type has already been captured on the system	
Trigger:	The super admin wants to delete a project type no longer in use.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin will click on the project type they wish to deactivate from the table of all project types [ALT]	**Step 2:** The system will load the form displaying all the project. Type and its name that they wish to deactivate. [ALT]
	Step 3: The user will click the "Deactivate project type function"	**Step 4:** The system will ask the user: "Are you sure you wish to deactivate" and then it will display the project type name for confirmation.
	Step 5: Upon The user will confirm the wish to deactivate that project type. [ALT]	**Step 6:** The system will deactivate the project type and show the project type and "has been deactivated". [ALT]
Alternate courses:	**[ALT] Step 1:**	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The project type has not yet been created; the super admin must click "Create new project type"
	[ALT] Step 2: The project type details have errors displaying, the system will display a message saying: "error, please restore the system"
	[ALT] Step 5: The admin doesn't want to deactivate this project type and clicks "Cancel".
	[ALT] Step 6: The project type has an error deactivating, and the system shows an error message to the admin to inform them their deactivation has been unsuccessful
Conclusion:	The project type has been deactivated.
Post-condition:	The project type is no longer displayed on view project types.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 70: Deactivate Project Type - Logical Use Case Narrative

2.7.9 Add Indemnity Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Add Indemnity Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.9	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin can add a new Indemnity form that can be added to projects to make form completions and capturing easier.	
Pre-condition:	The admin is signed in, and the form hasn't been added yet.	
Trigger:	The project requires a new type of indemnity form.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks on "Add new indemnity form"	**Step 2:** The system shows fields to place a name for the form and text and or field to upload pdf file.
	Step 3: The admin pastes the text and or pdf file and fills in a name [ALT]	**Step 4:** The system verifies the fields, and the form are correct before saving. [ALT]
		Step 5: The system notifies the admin that the new form has been successfully captured . [ALT]
	[ALT] Step 3: The user clicks "Cancel" to no longer add the form	
	[ALT] Step 4:	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The new indemnity form details have errors saving, the system will display a message saying: "error, please restore the system"
	[ALT] Step 5: The new indemnity form has an error saving, and the system shows an error message to the admin to inform them their capture has been unsuccessful
Conclusion:	The new indemnity form has been captures on the system
Post-condition:	The new indemnity form is shown with all indemnity forms in a table.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 71: Add Indemnity Form - Logical Use Case Narrative

2.7.10 Update Indemnity Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	7.10	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	Update Indemnity Form	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin can update an existing Indemnity form that can be added to projects to make form completions and capturing easier.	
Pre-condition:	The admin is signed in, and the form hasn't been added yet.	
Trigger:	The project requires an update to an indemnity forms context in order to be more accurate.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks on "update indemnity form"	**Step 2:** The system shows fields to place a name for the form and text and or field to upload pdf file where the admin can add or edit fields.
	Step 3: The admin pastes the new text and or new pdf file and or changes the name [ALT]	**Step 4:** The system verifies the fields, and the form are correct before saving. [ALT]
		Step 5: The system notifies the admin that the indemnity form has been successfully updated . [ALT]
Alternate courses:	**[ALT] Step 3:** The user clicks "Cancel" to no longer add the form	

INTEGRATED NGO MANAGEMENT SYSTEM	
	[ALT] Step 4: The indemnity form details have errors updating, the system will display a message saying: “error, please restore the system”
	[ALT] Step 5: The indemnity form has an error updating and the system shows an error message to the admin to inform them their capture has been unsuccessful.
Conclusion:	The new indemnity form has been captures on the system
Post-condition:	The new indemnity form is shown with all indemnity forms in a table.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 72: Update Indemnity Form - Logical Use Case Narrative

2.7.11 View Indemnity Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	View Indemnity Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.11	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin can click "View Indemnity Form" to be able to see, edit, add or deactivate Indemnity Forms to better help the projects.	
Pre-condition:	The user must already have a pre-existing account on the system Ther user must have super admin privileges	
Trigger:	The super admin wants to see all Indemnity forms.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks on "View Indemnity Forms" function on the projects page.	**Step 2:** The system retrieves the recorded Indemnity forms and displays them in a table for the super admin to view. They can then click "update" or "deactivate" on Indemnity forms or "Create new Indemnity Form" [ALT]
Alternate courses:	**[ALT] Step 2:** There is an error retrieving the Indemnity forms and the system displays an error message to the admin	
Conclusion:	The Indemnity forms are displayed in a table for the admin to easily access or view.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	The use is able view the indemnity form.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 73: View Indemnity Form - Logical Use Case Narrative

2.7.12 Remove Indemnity Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Remove Indemnity Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.12	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin can remove an existing indemnity form that is no longer being used to reduce clutter.	
Pre-condition:	The admin is signed in, and the form hasn't been added yet.	
Trigger:	The admin wants to remove an unused form.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin will click on the indemnity form they wish to deactivate from the table of all indemnity forms [ALT]	**Step 2:** The system will load the form displaying all the indemnity form content and its name that they wish to deactivate. [ALT]
	Step 3: The user will click the "Deactivate indemnity form function"	**Step 4:** The system will ask the user: "Are you sure you wish to deactivate" and then it will display the indemnity form name for confirmation.
	Step 5: Upon The user will confirm the wish to deactivate that indemnity form. [ALT]	**Step 6:** The system will deactivate the indemnity form and show the indemnity form and "has been deactivated". [ALT]
Alternate courses:	**[ALT] Step 1:**	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The indemnity form has not yet been created, the super admin must click “Create new indemnity form”
	[ALT] Step 2: The indemnity forms details have errors displaying, the system will display a message saying: “error, please restore the system”
	[ALT] Step 5: The admin doesn’t want to deactivate this indemnity form and clicks “Cancel”
	[ALT] Step 6: The indemnity form has an error deactivating, and the system shows an error message to the admin to inform them their deactivation has been unsuccessful
Conclusion:	The indemnity form has been deactivated.
Post-condition:	The indemnity form is no longer displayed on view indemnity forms.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 74: Remove Indemnity Form - Logical Use Case Narrative

2.7.13 Capture Indemnity Form -

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Capture Indemnity Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.13	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A super admin can attach an indemnity form to a project which will make it so that anyone who has been added to that project must complete the compulsory indemnity form to view the project details	
Pre-condition:	The super admin must already have a pre-existing account on the system	
Trigger:	The project would benefit from including an indemnity form.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks a box with the text ”Include an indemnity form”	**Step 2:** The system will display a list of the current indemnity forms for the admin to select from, along with a “create new indemnity form” function
	Step 3: The admin will select the indemnity form they require from the list of options. [ALT]	**Step 4:** The system will capture their selection and make a signature for that indemnity form be required for all staff in that project.
		Step 5: Upon creating the indemnity form task, the

INTEGRATED NGO MANAGEMENT SYSTEM		
		system will notify the admin of the successful [ALT]
Alternate courses:	**[ALT] Step 3:** Admin will create a new form, if their choice isn't on the list	
	[ALT] Step 5: The system will give a notification error if unsuccessful	
Conclusion:	The indemnity form must now be required to be captured before all staff can interact with that project.	
Post-condition:	The required event is now visible to all users involved in the project.	
Business rules:	• Secure Access • Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None	
Open issues:	• None	

Table 75: Remove Indemnity Form - Logical Use Case Narrative

2.7.14 Capture Project Feedback -

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Capture Project Feedback	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.14	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A super admin can attach a project feedback form to a project which will make it so that anyone who has been added to that project will be send a form 1 week after the project commences asking them for a rating and additional comments to help improve the events.	
Pre-condition:	The project assigned to the feedback form must have commenced	
Trigger:	The super admin clicks a radio function called "Include Project Feedback form".	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks a box with the text "Include a Project Feedback form"	**Step 2:** The system will capture their selection and make a project feedback form for all staff in that project after the project ends.
		Step 3: Upon creating the indemnity form task, the system will notify the admin of the successful [ALT]
Alternate courses:	**[ALT] Step 3:** Admin will be notified if there is an error.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Conclusion:	The project feedback form will be displayed to all staff within that project, after the project is over
Post-condition:	The form will open for all users after the event has finished
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 76: Capture Project Feedback - Logical Use Case Narrative

2.7.15 View Project Feedback -

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	View Project Feedback	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.15	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A super admin can view all of the past projects and see the feedback forms ratings and comments for each project	
Pre-condition:	The projects have been saved on the system and not been deactivated	
Trigger:	The admin clicks on a specific project on view feedback forms. Or if the search a completed project in the projects nav bar	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The admin clicks on a specific project that has been completed, or they click on a specific project on the feedback page. [ALT]	**Step 2:** The system retrieves the records for the feedback ratings and comments for that project and displays it neatly for the admin to read. [ALT]
Alternate courses:	**[ALT] Step 1:** The admin clicks cancel if they are on the wrong project.	
	[ALT] Step 2: The system cannot open the information from the system and displays an error message.	
Conclusion:	The admin can see the feedback results of their desired project.	
Post-condition:	Active projects are all displayed with their feedback data inside.	

Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 77: View Project Feedback - Logical Use Case Narrative

2.7.16 Create Impact Feedback Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Create Impact Feedback Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.16	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin can add a new impact form that can be added to projects to make form completions and capturing easier for community members.	
Pre-condition:	The admin creating the form has permissions to create impact forms already.	
Trigger:	The admins would like to capture data from their projects impact or just for general use.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks on "Add new impact form"	**Step 2:** The system shows fields to place a name for the impact form and a function that says "Add question" where the admin can choose a question type and write the question. The admin can add as many questions as they want
	Step 3: The admin pastes the text and all their questions for the community partners. [ALT]	**Step 4:** The system verifies the fields, and the impact form is correct before saving. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 5: The system notifies the admin that the new impact form has been successfully captured . [ALT]
Alternate courses:	**[ALT] Step 3:** The user clicks "Cancel" to no longer add the form	
	[ALT] Step 4: The new indemnity form details have errors saving, the system will display a message saying: "error, please restore the system"	
	[ALT] Step 5: The new indemnity form has an error saving, and the system shows an error message to the admin to inform them their capture has been unsuccessful	
Conclusion:	The new impact form has been captures on the system	
Post-condition:	The new impact form is shown with all impact forms in a table.	
Business rules:	• Secure Access • Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None	
Open issues:	• None	

Table 78: Create Impact Feedback Form - Logical Use Case Narrative

2.7.17 View Impact Feedback Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	View Impact Feedback Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.17	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A super admin can view all of the past projects and see the impact forms questions and answers and see what project they are assigned to.	
Pre-condition:	The projects have been saved on the system and not been deactivated	
Trigger:	The admin clicks on a specific project on view impact forms of. Or if the search a completed project in the projects nav bar	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The admin clicks on a specific project that has been completed, or they click on a specific project on the impact page. [ALT]	**Step 2:** The system retrieves the records for the impact questions and answers for that project and displays it neatly for the admin to read. [ALT]
Alternate courses:	**[ALT] Step 1:** The admin clicks cancel if they are on the wrong project.	
	[ALT] Step 2: The system cannot open the information from the system and displays an error message.	
Conclusion:	The admin can see the impact results of their desired project.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	Active projects are all displayed with their feedback data inside.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 79: View Impact Feedback Forms - Logical Use Case Narrative

2.7.18 Update Impact Feedback Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Update Impact Feedback Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.18	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• User	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The super admin can update an existing impact form that can be added to projects to make form completions and capturing easier. The questions we ask community partners can be updated	
Pre-condition:	The admin is signed in, and the form hasn't been added yet.	
Trigger:	The project requires an update to an impact forms context to capture more accurate.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin clicks on "update impact form"	**Step 2:** The system shows fields to place a name for the form and the questions that are currently on the form
	Step 3: The admin pastes the new questions and question types and or changes the name of the impact form [ALT]	**Step 4:** The system verifies the fields, and the impact form is correct before saving. [ALT]
		Step 5: The system notifies the admin that the impact form has been successfully updated . [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM	
Alternate courses:	**[ALT] Step 3:** The user clicks "Cancel" to no longer add the form
	[ALT] Step 4: The impact form details have errors updating, the system will display a message saying: "error, please restore the system"
	[ALT] Step 5: The impact form has an error updating and the system shows an error message to the admin to inform them their capture has been unsuccessful
Conclusion:	The new impact form has been captures on the system
Post-condition:	The new impact form is shown with all impact forms in a table.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 80: Update Impact Feedback Form - Logical Use Case Narrative

2.7.19 Deactivate Impact Feedback Form

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Deactivate Impact Feedback Form	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.19	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	The admin can deactivate an impact form from the list of all the forms being displays, this can remove unused forms and clear clutter from the system.	
Pre-condition:	The impact form has already been created and is displayed on the list	
Trigger:	The admin wishes to remove a form that is no longer needed or being used.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin will click on the impact form they wish to deactivate from the table of all impact forms [ALT]	**Step 2:** The system will load the form displaying all the impact form content and its name that they wish to deactivate. [ALT]
	Step 3: The user will click the "Deactivate impact form function"	**Step 4:** The system will ask the user: "Are you sure you wish to deactivate" and then it will display the impact form name for confirmation.
	Step 5: Upon The user will confirm the wish to	**Step 6:** The system will deactivate the impact form and show the impact form

INTEGRATED NGO MANAGEMENT SYSTEM		
	deactivate that impact form. [ALT]	and "has been deactivated". [ALT]
Alternate courses:	**[ALT] Step 1:** The impact form has not yet been created, the super admin must click "Create new impact form"	
	[ALT] Step 2: The impact forms details have errors displaying, the system will display a message saying: "error, please restore the system"	
	[ALT] Step 5: The admin doesn't want to deactivate this impact form and clicks "Cancel"	
	[ALT] Step 6: The impact form has an error deactivating, and the system shows an error message to the admin to inform them their deactivation has been unsuccessful	
Conclusion:	The impact form has been deactivated.	
Post-condition:	The impact form is no longer displayed on view forms.	
Business rules:	• Secure Access • Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None	
Open issues:	• None	

Table 81: Deactivate Impact Feedback Form - Logical Use Case Narrative

2.7.20 Capture Impact Feedback

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Capture Impact Feedback	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.20	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• Community Partner	
Other interested stakeholders:	Administrator	
Description:	When creating or editing a project the super admin can check the "Include impact form" function and choose from the list of impact forms or create a new one and add an email address to send the form to. The system will then capture the results and send the impact form.	
Pre-condition:	The impact form has already been captured in the system	
Trigger:	The project admin wants to capture impact data regarding their project.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin will select a box that says: "Include impact form"	**Step 2:** The system retrieves the recorded impact forms for the user to choose from and adds a field to enter the contact's email address or phone number. [ALT]
	Step 3: The super admin chooses the impact form they wish to use and enters the community partners contact details. [ALT]	**Step 4:** The system verifies the number, email and fields and captures the questions. [ALT]
		Step 5: The system sends an email/ message to the community partner so they

INTEGRATED NGO MANAGEMENT SYSTEM		
		can answer the questions, and the system will capture their answer. [ALT]
		Step 6: Once the form has been completed the system will notify the admin.
Alternate courses:	**[ALT] Step 2:** No impact forms could be retrieved, and we send an error message to the admin	
	[ALT] Step 3: The admin clicks "Cancel" and returns to the viewing page.	
	[ALT] Step 4: The fields are not correct, and we notify the admin of their error.	
	[ALT] Step 5: There is an error sending the impact form, and we notify the admin that the form was unable to send	
Conclusion:	The impact form has been sent to the community partner.	
Post-condition:	The community partner can open and complete the form.	
Business rules:	• Secure Access • Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None	
Open issues:	• None	

Table 82: Capture Impact Feedback - Logical Use Case Narrative

2.7.21 View Impact Feedback

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	View Impact Feedback	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.21	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• Super Admin	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A super admin can view all of the project's impact results and see the impact form questions and answers and see what project they are assigned to and which community partner answered.	
Pre-condition:	The impact forms have been saved on the system and not been deactivated	
Trigger:	The admin clicks on a specific impact form on view impact forms of. Or if the search a completed project in the projects nav bar	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The admin clicks on a specific impact form that has been completed, or they click on a specific project on the impact page. [ALT]	**Step 2:** The system retrieves the records for the impact questions and answers for that project and displays it neatly for the admin to read. [ALT]
Alternate courses:	**[ALT] Step 1:** The admin clicks cancel if they are on the wrong impact form.	
	[ALT] Step 2: The system cannot open the information from the system and displays an error message.	
Conclusion:	The admin can see the impact results of their impact form.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	Active projects are all displayed with their feedback data inside.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 83: View Impact Feedback - Logical Use Case Narrative

2.7.22 Add Photo

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Add Photo	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.22	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• User	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A user can click on a project to see all photos from that project; they can then click the add photos function to upload multiple photos to that album.	
Pre-condition:	The project's photo group has already been added, and the user has access to that project	
Trigger:	The user wishes to add more images to the project's image folder.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the project group they wish to add photos to in the photos page and clicks the "Add photos function"	**Step 2:** The system will load a page where the user can upload multiple images from their device. [ALT]
	Step 3: The user drags and drops or uploads their images to the photo album	**Step 4:** The system captures all the new images onto the existing photo album.
		Step 5: Upon the success of the image uploading, the system will notify the user of the success. [ALT]
Alternate courses:	**[ALT] Step 2:**	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The system has an error displaying the correct photo album and sends an error message to the user
	[ALT] Step 5: The images are not uploaded successfully, and the system notifies the user of the error.
Conclusion:	User's photos have been successfully added to the photo album.
Post-condition:	The newly added images appear in the project's photo album.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 84: Add Photo - Logical Use Case Narrative

2.7.23 View Photo

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	View Photo	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.23	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• User	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A user can click on a project to see all photos from that project; they can then click the project to see all photos in that album. All projects are displayed in a table with their names and the project photo.	
Pre-condition:	All the project's photo group has already been added, and the user has access to those projects	
Trigger:	The user wishes to view the images in a project's image folder.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the project group they wish to view the photos of from a list of all the photo albums	**Step 2:** The system will load a page where the user can view multiple images from that project. [ALT]
Alternate courses:	**[ALT] Step 2:** The system has an error displaying the correct photo album and sends an error message to the user	
Conclusion:	The project's photos have been successfully displayed to the user.	
Post-condition:	The images appear in the project's photo album.	
Business rules:	• Secure Access • Role Based Access	

INTEGRATED NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 85: View Photo - Logical Use Case Narrative

2.7.24 Delete Photo

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Alexander Steven	**Date:** 25 March **2026** **Version: 1.0**
Use case name:	Delete Photo	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.24	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• User	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	A user can click on a project to see all photos from that project; they can then click the delete photos function to make all photos in that album that they have uploaded have a remove function the user can press to remove them.	
Pre-condition:	• The project's photo group has already been added, and the user has access to that project • The user uploaded the photos they wish to delete, or they are a super admin	
Trigger:	The user wishes to remove images from the project's image folder.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the project group they wish to remove photos from in the photos page and clicks the "Remove photos function"	**Step 2:** The system will load a page where the user can see multiple images from the folder, with an "X" function they can press to remove that photo. [ALT]
	Step 3: The user clicks on all the photos they wish to delete and then click "done" [ALT]	**Step 4:** The system captures all the images to remove and removes them from the existing photo album.

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 5: Upon the success of the image removal, the system will notify the user of the success. [ALT]
Alternate courses:	**[ALT] Step 2:** The system has an error displaying the correct photo album and sends an error message to the user	
	[ALT] Step 3: The user clicks "Cancel" if they are on the wrong photo album.	
	[ALT] Step 5: The images are not uploaded successfully, and the system notifies the user of the error.	
Conclusion:	User's photos have been successfully removed from the photo album.	
Post-condition:	The removed images do not appear in the project's photo album.	
Business rules:	• Secure Access • Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• The user is deleting photo's they uploaded or are a super admin	
Open issues:	• None	

Table 86: Delete Photo - Logical Use Case Narrative

2.7.25 Add Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 14 August 2026** **Version: 1.0**
Use case name:	Add Photo Album	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.25	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• User	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	This use case describes an event that will allow a User with appropriate permissions to create a new photo album under a specific project.	
Pre-condition:	• User is authenticated. • The target project already exists in the system and is active.	
Trigger:	User selects the "Add Photo Album" function within a project.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: User selects "Add Photo Album" within a project.	Step 2: The System displays the Add Photo Album form.
	Step 3: User enters the album title, optional description, and assigns media permissions or categories.	
	Step 4: User confirms creation [ALT].	Step 5: System validates that the album name is unique within the project.
		Step 6: System creates the album folder structure and records the creation entry.

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: System confirms successful album creation.
Alternate courses:	**[ALT] Step 4:** User cancels the Add Photo Album function.	
Conclusion:	The new photo album is created and made available under the project.	
Post-condition:	• The new album is available for photo uploads and viewing.	
Business rules:	• Secure Access. • Role Based Access.	
Implementation constraints and specifications:	• Album names must be unique within the same project. • Interface must be easy to use.	
Assumptions:	• The user has been granted appropriate permissions to create albums.	
Open issues:	• None	

Table 87: Add Photo Album - Logical Use Case Narrative

2.7.26 View Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date:14 August2026** **Version: 1.0**
Use case name:	View Photo Album	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.26	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• User	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	This use case describes an event that will allow a User with appropriate permissions to view a project's photo album and its contents.	
Pre-condition:	• User is authenticated. • User has read access to the associated project.	
Trigger:	User selects a project photo album to view.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: User selects a project photo album.	Step 2: The System retrieves the album and its associated photos.
		Step 3: System displays a gallery grid of thumbnail images, album metadata, and photo upload timestamps.
	Step 4: User selects a photo to view in detail [ALT].	Step 5: System displays the selected photo at full size.
Alternate courses:	**[ALT] Step 4:** User exits the album view without selecting a photo.	
Conclusion:	The User successfully views the photo album and its contents.	
Post-condition:	• Album contents are displayed to the user.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Business rules:	• Secure Access. • Role Based Access.
Implementation constraints and specifications:	• Images must load efficiently. • Interface must be easy to use.
Assumptions:	• Thumbnail images are pre-generated or generated on demand for efficient loading.
Open issues:	• None

Table 88: View Photo Album - Logical Use Case Narrative

2.7.27 Update Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 14 August 2026** **Version: 1.0**
Use case name:	Update Photo Album	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.27	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Admin	
Primary system actor (PSA):	• System	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	This use case describes an event that will allow a User with appropriate permissions to edit the metadata of an existing photo album, such as the title, description, or cover photo, and manage its photo organization.	
Pre-condition:	• The project's photo group has already been added, and the user has access to that project • User is authenticated	
Trigger:	User selects the "Update Photo Album" function	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: User selects a photo album to edit.	Step 2: The System retrieves the album and displays its current metadata.
	Step 3: User edits the album title, description, cover photo, and/or photo organization.	
	Step 4: User confirms changes [ALT].	Step 5: System validates the input fields.
		Step 6: System updates the album context.
		Step 7: System confirms successful update.

INTEGRATED NGO MANAGEMENT SYSTEM	
Alternate courses:	**[ALT] Step 4:** User cancels the Update Photo Album function.
Conclusion:	The photo album is updated and reflects the latest changes
Post-condition:	Updated album details are visible to users with access
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• The user has verified the accuracy of the updated details before saving
Open issues:	• None

Table 89: Update Photo Album - Logical Use Case Narrative

2.7.28 Remove Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Amahle Khoza	**Date: 14 August 2026** **Version: 1.0**
Use case name:	Remove Photo Album	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	7.28	
Priority:	High	
Source:	EPI-USE	
Primary business actor (PBA):	• Super Admin	
Primary system actor (PSA):	• User	
Other participating actors:	• None	
Other interested stakeholders:	Administrator	
Description:	This use case describes an event that will allow a User with appropriate permissions to delete an entire photo album, and its associated files, from a project.	
Pre-condition:	• The project's photo group has already been added, and the user has access to that project • The user uploaded the photos they wish to delete, or they are a super admin	
Trigger:	User selects the album and clicks the delete button.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: User selects a photo album and clicks "Delete".	Step 2: The System displays a confirmation modal.
	Step 3: User confirms deletion [ALT].	Step 4: System removes the album record.
		Step 5: System purges all underlying photo files associated with the album.
		Step 6: System confirms successful removal.
Alternate courses:	**[ALT] Step 3:** The system has an error displaying the correct photo album and sends an error message to the user	
	[ALT] Step 3:	

INTEGRATED NGO MANAGEMENT SYSTEM	
	User cancels the deletion and the confirmation modal is dismissed without changes
Conclusion:	The photo album and its associated files are removed from the project.
Post-condition:	The album is no longer available for viewing or photo uploads.
Business rules:	• Secure Access • Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use. • Deletion requires user confirmation to prevent accidental data loss.
Assumptions:	• No other processes depend on the album at the time of deletion
Open issues:	• None

Table 90: Remove Photo Album - Logical Use Case Narrative

2.8 K9 Subsystem Use Case Narratives

2.8.1 Add Dog

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Add Dog	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.1	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a super admin creates a new K9 within the system. The system allows the user to enter the K9's details, upload an optional image, assign K9 roles, and optionally select the K9's parents. The system then validates the entered information, confirms the creation, and saves the K9 to the database. The K9 is stored along with its associated roles and image, the K9 list is refreshed, and a confirmation toaster is displayed. The transaction is also recorded in the audit log.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to add a new dog to the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to add a new dog to the system.	**Step 2:** The system invokes UC 8.3 (View Dogs) to display the list of dogs.
	Step 3: The super admin selects to add a new dog.	**Step 4:** The system displays a Create K9 Modal to provide the following information:

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Name • Breed • Gender • Date of Birth With optional fields which include: • K9 Roles • Parents • Image
	Step 5: The super admin enters the dog's details and selects create. [ALT]	**Step 6:** The system validates the entered details. [ALT]
		Step 7: The system requests confirmation for the new dog to be added.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the new dog to the following tables: • Animal • K9 • K9K9Role [ALT]
		Step 10: The system records the transaction to the AuditLog table.
		Step 11: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and displays an error message within the Create K9 modal. Go back to step 4.	
	[ALT] Step 8: The admin does not confirm the changes. Go back to step 4.	
	[ALT] Step 9: The system fails to save the new dog and displays a toaster message. Terminate use case.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Conclusion:	The new dog's details have been saved onto the system, and a confirmation toaster is shown.
Post-condition:	The system saves the new K9 in the K9 and Animal table, stores any associated K9 role assignments and makes the newly created K9 available for viewing and management within the system.
Business rules:	• Only super admins or users given appropriate permissions can add a new dog. •
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 91: Add Dog - Logical Use Case Narrative

2.8.2 View Dog

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Dog	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.2	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Handler	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a handler views the details of a specific K9 within the system. The system allows the handler to search for or filter the available K9s, select a K9 from the list displayed, and view its details. The system retrieves the selected K9's information from the database and displays it in a View K9 Details modal.	
Pre-condition:	The handler must be logged into the system.	
Trigger:	The handler wants to view a dog on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The handler wants to view a specific dog on the system.	**Step 2:** The system invokes UC 8.3 (View Dogs) to display the list of dogs.
	Step 3: The handler searches for a K9 by entering its name or breed in the search field or filters the displayed K9s by gender or status. [ALT]	**Step 4:** The system filters and displays the K9s matching the entered search criteria and selected filters. [ALT]
	Step 5: The handler selects the K9 card.	**Step 6:** The system retrieves the selected dog's details from the following tables:

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Animal • K9 • K9K9Role • K9Role • K9User • User
		Step 7: The system displays a View K9 Details modal with all the dog's details. These include: • Name • Image • Breed • Gender • Age • Date of Birth • Parents • Status • Assigned K9 Roles • Assigned Handler
	Step 8: The handler can view the dog's details.	
Alternate courses:	**[ALT] Step 3:** The handler does not search or filter for a dog and just selects the dog directly from the ones displayed. Go to Step 5.	
	[ALT] Step 4: The system does not find any dogs matching the specified criteria. Go back to step 2.	
Conclusion:	The system displays a modal showing the dog's details.	
Post-condition:	The dog's details are displayed.	
Business rules:	• Only super admins or users given appropriate permissions can view a dog.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 92: View Dog - Logical Use Case Narrative

2.8.3 View Dogs

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Dogs	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.3	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Handler	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a handler views all K9s registered in the system. The system retrieves the K9 records from the database and displays them as K9 cards on the K9 page. The handler can then search for K9s by name or breed, filter the displayed K9s by gender or status, and select a K9 to view its detailed information.	
Pre-condition:	The handler must be logged into the system.	
Trigger:	The handler wants to view all dogs on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The handler wants to view all dogs on the system.	
	Step 2: The handler selects the animal tab. [ALT]	**Step 3:** The system displays a secondary navigation bar.
	Step 4: The handler then selects K9.	**Step 5:** The system retrieves the current dogs on the system from the following tables: • Animal • K9

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 6: The system displays the current dogs with the following details: • Image • Name • Age • Gender • Breed
	Step 7: The handler can view the dogs.	
Alternate courses:	ALT-Step 2: User is on mobile and selects the menu icon. Go to step 3.	
Conclusion:	The system displays all the dogs on the system.	
Post-condition:	All the dogs on the system are displayed.	
Business rules:	• Only super admins or users given appropriate permissions can view dogs.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 93: View Dogs - Logical Use Case Narrative

2.8.4 Update Dog

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update Dog	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.4	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a super admin updates the details of an existing K9 within the system. The system allows the super admin to modify the K9's information, update its assigned roles, change its parent information, and optionally replace the K9's image. The system validates the updated information, requests confirmation before saving the changes, updates the K9 record and its associated role assignments in the database, records the transaction in the audit log, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to update a dog's details on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to update a dog's details on the system.	**Step 2:** The system invokes UC 8.2 (View Dog) to display a pop-up showing all the dog's details.
	Step 3: The super admin selects to update the dog's details. [ALT]	**Step 4:** The system displays an Edit K9 modal with the selected K9's current information.

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Name • Breed • Gender • K9 Roles • Date of Birth • Parents • Image
	Step 5: The super admin updates the dog's details and selects save changes. [ALT]	**Step 6:** The system validates the updated data. [ALT]
		Step 7: The system requests confirmation for the changes to be made to the dog's details.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the changes to the following tables: • Animal • K9 • K9K9Roles [ALT]
		Step 10: The system records the transaction to the AuditLog table.
		Step 11: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 3:** The selected K9 is inactive, and the "Edit" function is not displayed. Terminate use case.	
	[ALT] Step 5: The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and an error message is displayed within the Edit K9 modal. Go back to step 4.	
	[ALT] Step 8: The admin does not confirm the changes. Go back to step 4.	

INTEGRATED NGO MANAGEMENT SYSTEM	
	[ALT] Step 9: The system fails to save the changes and displays a toaster message. Terminate use case.
Conclusion:	The new dog's details have been saved onto the system, and a confirmation toaster is shown.
Post-condition:	The system saves the new K9 details in the K9 and Animal table, stores any associated new K9 role assignments, and makes the newly updated K9's details available for viewing and management within the system.
Business rules:	• Only super admins or users given appropriate permissions can update a dog.
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 94: 8.4 Update Dog - Logical Use Case Narrative

2.8.5 Deactivate Dog

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Deactivate Dog	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.5	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin deactivates an existing K9. The system displays the selected K9's details, requests confirmation before proceeding, updates the K9's status to inactive, records the action in the audit log, and displays a confirmation message.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to deactivate a dog on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to deactivate a dog on the system.	**Step 2:** The system invokes UC 8.2 (View Dog) to display a pop-up showing all the dog's details.
	Step 3: The super admin selects deactivate. [ALT]	**Step 4:** The system requests confirmation for the dog to be deactivated.
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system sets the dog's status to inactive in the Animal table. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system records the transaction to the AuditLog table.
		Step 8: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 3:** The selected K9 is inactive, and the "Deactivate" function is not displayed. Terminate use case.	
	[ALT] Step 5: The super admin does not confirm. Go back to step 2.	
	[ALT] Step 6: The system fails to change the status and displays a toaster message. Terminate use case.	
Conclusion:	The K9 has been deactivated on the system, and a confirmation toaster is shown.	
Post-condition:	The system sets the K9's status to inactive in the Animal table and the K9 is no longer displayed in the active K9 listings.	
Business rules:	• Only super admin or users given appropriate permissions can deactivate a dog.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 95: Deactivate Dog - Logical Use Case Narrative

2.8.6 Assign User to Dog

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Assign User to Dog	**USE CASE TYPE**
Use case id:	8.6	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• Handler (ERA)	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin assigns a handler to a selected dog. The system displays an assignment modal that allows the super admin to search for and select an active handler. Once the assignment is confirmed, the system validates the request, creates a new handler assignment linked to the selected dog, notifies the assigned handler, records the transaction in the audit log, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to assign a handler to a dog on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to assign a handler to a dog on the system.	**Step 2:** The system invokes UC 8.2 (View Dog) to display a pop-up showing all the dog's details.
	Step 3: The super admin selects to assign a handler. [ALT]	**Step 4:** The system displays an Assign Handler modal to allow a handler to be assigned to a dog.

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin searches for and selects a handler and then selects assign. [ALT]	**Step 6:** The system requests confirmation for the handler to be assigned.
	Step 7: The super admin confirms. [ALT]	**Step 8:** The system creates a user-dog assignment in the K9User table, linking the selected handler to the dog. [ALT]
		Step 9: The system notifies the assigned handler of the new assignment.
		Step 10: The system records the transaction to the AuditLog table.
		Step 11: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 3:** The selected K9 is inactive, and the "Assign" function is not displayed. Terminate use case.	
	[ALT] Step 5: The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 7: The super admin does not confirm. Go back to step 4.	
	[ALT] Step 8: The system fails to create user-dog assignment and displays a toaster message. Terminate use case.	
Conclusion:	The dog has been assigned to a handler on the system, and a confirmation toaster is shown.	
Post-condition:	A new handler assignment is created in the K9User table, linking the selected handler to the selected dog, and the assignment becomes the dog's current active handler.	
Business rules:	• Only super admins or users given appropriate permissions can assign a user to a dog. • A dog can only have one handler.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 96: Assign User to Dog - Logical Use Case Narrative

2.8.7 View User-Dog Assignment

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View User-Dog Assignment	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.7	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Handler	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how a handler views all K9 handler assignments recorded in the system. The system retrieves all K9 and handler assignment records from the database and displays them in a table, allowing the handler to search by K9 or handler name and filter assignments by their status (Active, Inactive, or All).	
Pre-condition:	The handler must be logged into the system.	
Trigger:	The handler wants to view all the dogs and their current handlers on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The handler wants to view all the dogs and their current handlers on the system.	**Step 2:** The system invokes UC 8.3 (View Dogs) to display the list of dogs.
	Step 3: The handler selects view assignments.	**Step 4:** The system retrieves the current assignments from the following tables: • K9User • K9 • User
		Step 5: The system displays a the K9 Assignments page

INTEGRATED NGO MANAGEMENT SYSTEM		
		containing a table of all handler assignments. The table includes: • K9 Name • Handler Name • Start Date • End Date • Assignment Status [ALT]
	Step 6: The handler can view the assignments.	
Alternate courses:	**[ALT] Step 5:** There are currently no assignments, and the system displays a message to the handler. Terminate use case.	
Conclusion:	The system displays all the assignments on the system.	
Post-condition:	The current K9 handler assignments are displayed on the K9 Assignments page, and the handler can view, search, and filter the assignment records.	
Business rules:	• Only super admins or users given appropriate permissions can view assignments.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 97: View User-Dog Assignment - Logical Use Case Narrative

2.8.8 Update User-Dog Assignment

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update User-Dog Assignment	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.8	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• Handler (ERA)	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin updates the handler assigned to a selected dog. The system displays an update handler modal that allows the super admin to search for and select a different active handler. Once the update is confirmed, the system validates the request, ends the current handler assignment, creates a new handler assignment linked to the selected dog, notifies both the previously assigned and newly assigned handlers via email, records the transaction in the AuditLog, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to change a dog's handler on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to change a dog's handler on the system.	**Step 2:** The system invokes UC 8.2 (View Dog) to display a pop-up showing all the dog's details.
	Step 3: The super admin selects update to change the handler. [ALT]	**Step 4:** The system displays an Update Handler modal to allow the dog's handler to be changed.

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin searches for and selects a new handler, then selects the update function. [ALT]	**Step 6:** The system requests confirmation for the handler to be changed.
	Step 7: The super admin confirms. [ALT]	**Step 8:** The system creates a new user-dog assignment on the K9User table, linking the selected handler to the dog and ends the previous assignment by setting the end date to the current date. [ALT]
		Step 9: The system notifies the old handler that they are no longer assigned to this dog.
		Step 10: System notifies the new handler that they have been assigned to this dog.
		Step 11: The system records the transaction to the AuditLog table.
		Step 12: The system displays a confirmation toaster.
Alternate courses:	ALT-Step 3: The selected K9 is inactive, and the "Update" function is not displayed. Terminate use case.	
	[ALT] Step 5: The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 7: The super admin does not confirm. Go back to step 4.	
	[ALT] Step 8: The system fails to update the user-dog assignment and displays a toaster message. Terminate use case.	
Conclusion:	The dog has been assigned to a handler on the system, and a confirmation toaster is shown.	
Post-condition:	The current handler assignment is updated successfully. The previous assignment is marked as inactive, a new active assignment is created for the selected handler, notification	

INTEGRATED NGO MANAGEMENT SYSTEM	
	emails are sent to both affected handlers, and the updated assignment is available for viewing on the system.
Business rules:	• Only super admins or users given appropriate permissions can change a dog's handler. • A dog can only have one handler.
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 98: Update User-Dog Assignment - Logical Use Case Narrative

2.8.9 Reactivate User-Dog Assignment

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Reactivate User-Dog Assignment	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.9	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• Handler (ERA)	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin restores a previous handler assignment for a selected dog. The system allows the super admin to reactivate the most recent previous handler assignment after confirmation. The system validates the request, creates a new active assignment for the previous handler, notifies the restored handler via email, records the transaction in the AuditLog, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to reactivate a user-dog assignment on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to reactivate a user-dog assignment on the system.	**Step 2:** The system invokes UC 8.2 (View Dog) to display a pop-up showing all the dog's details.
	Step 3: The super admin selects to reactivate the previous handler assignment. [ALT]	**Step 4:** The system requests confirmation for the user-dog assignment reactivation.

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system retrieves the most recent assignment and reactivates the assignment by creating a new record in the K9User table with the previous handler. [ALT]
		Step 7: System notifies the handler that they have been assigned to this dog.
		Step 8: The system records the transaction to the AuditLog table.
		Step 9: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 3:** The selected K9 is inactive, and the "Reactivate" function is not displayed. Terminate use case.	
	[ALT] Step 5: The super admin does not confirm. Go back to step 2.	
	[ALT] Step 6: The system fails to reactivate the user-dog assignment and displays a toaster message. Terminate use case.	
Conclusion:	The dog's previous assignment has been reactivated, and a confirmation toaster is shown.	
Post-condition:	The previous handler assignment is restored successfully. A new active assignment is created for the previous handler, the restored handler is notified, and the updated assignment is available for viewing on the system.	
Business rules:	• Only super admin or users given appropriate permissions can reactivate a user-dog assignment.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 99: Reactivate User-Dog Assignment - Logical Use Case Narrative

2.8.10 Deactivate User-Dog Assignment

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Deactivate User-Dog Assignment	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.10	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• Handler (ERA)	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin removes the current handler assignment from a selected dog. The system requests confirmation before removing the assignment. Once confirmed, the system validates the request, deactivates the current user-dog assignment by recording an end date, notifies the removed handler, records the transaction in the audit log, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to deactivate a user-dog assignment on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to deactivate a user-dog assignment on the system.	**Step 2:** The system invokes UC 8.2 (View Dog) to display a pop-up showing all the dog's details.
	Step 3: The super admin selects to deactivate the current handler assignment. [ALT]	**Step 4:** The system requests confirmation for the user-dog assignment deactivation.

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system deactivates the current assignment in the K9User table. [ALT]
		Step 7: The system notifies the handler that they are no longer assigned to this dog.
		Step 8: The system records the transaction to the AuditLog table.
		Step 9: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 3:** The selected K9 is inactive, and the "Deactivate" function is not displayed. Terminate use case.	
	[ALT] Step 5: The super admin does not confirm. Go back to step 2.	
	[ALT] Step 6: The system fails to deactivate the user-dog assignment and displays a toaster message. Terminate use case.	
Conclusion:	The dog's current assignment has been deactivated, and a confirmation toaster is shown.	
Post-condition:	The current user-dog assignment has been deactivated. The selected dog no longer has an active handler assignment, the previous handler has been notified of the removal, and the updated assignment is available for viewing on the system.	
Business rules:	• Only super admins or users given appropriate permissions can deactivate a user-dog assignment.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 100: Deactivate User-Dog Assignment - Logical Use Case Narrative

2.8.11 Add K9 Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Add K9 Role	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.11	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin creates a new K9 role within the system. The system displays a modal that allows the super admin to enter the role name and description. Once the information is validated and confirmed, the system creates the new K9 role, records the transaction in the audit log, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to add a new K9 role to the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to add a new K9 role to the system.	**Step 2:** The system invokes UC 8.13 (View K9 Roles) to display a page with a table showing all the current roles
	Step 3: The super admin selects create role.	**Step 4:** The system displays a Create K9 Role modal to provide the following information: • Role • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin enters the new role's details. [ALT]	**Step 6:** The system validates the details entered. [ALT]
		Step 7: The system requests confirmation for the new role to be added.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the new role onto the K9Role table. [ALT]
		Step 10: The system records the transaction to the AuditLog table.
		Step 11: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and an error message is displayed within the Create K9 Role modal. Go back to step 4.	
	[ALT] Step 8: The super admin does not confirm. Go back to step 4.	
	[ALT] Step 9: The system fails to save the new K9 role and displays a toaster message. Terminate use case.	
Conclusion:	The new K9 role's details have been saved onto the system, and a confirmation toaster is shown.	
Post-condition:	The system saves the new K9 Role in the K9Role table and makes the newly created role available for assignment to K9s and for management within the system.	
Business rules:	• Only super admin or users given appropriate permissions can add a new K9 role.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	

INTEGRATED NGO MANAGEMENT SYSTEM	
Open issues:	• None

Table 101: Add K9 Role - Logical Use Case Narrative

2.8.12 Update K9 Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update K9 Role	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.12	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin updates an existing K9 role within the system. The system displays the selected K9 role's details in an editable form, allowing the super admin to modify the role name and description. Once the updated information is validated and confirmed, the system saves the changes, records the transaction in the audit log, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to update a K9 role's detail on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to update a K9 role's detail on the system.	**Step 2:** The system invokes UC 8.13 (View K9 Roles) to display a page with a table showing all the current roles
	Step 3: The super admin selects update on a specific K9 role.	**Step 4:** The system displays the Edit K9 Role modal with the following information: • Role • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin updates the role's details and selects save. [ALT]	**Step 6:** The system validates the details entered. [ALT]
		Step 7: The system requests confirmation for the changes to be made to the role's details.
	Step 8: The super admin confirms. [ALT]	**Step 9:** The system saves the updated role's details onto the K9Role table. [ALT]
		Step 10: The system records the transaction to the AuditLog table.
		Step 11: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin changes their mind and closes the modal. Terminate use case.	
	[ALT] Step 6: The system finds an error after validating and an error message is displayed within the Edit K9 Role modal. Go back to step 4.	
	[ALT] Step 8: The super admin does not confirm. Go back to step 4.	
	[ALT] Step 9: The system fails to save the changes and displays a toaster message. Terminate use case.	
Conclusion:	The updated K9 role's details have been saved onto the system, and a confirmation toaster is shown.	
Post-condition:	The system saves the updated K9 Role details in the K9Role table, and the modified role is immediately available for viewing, assignment to K9s, and management within the system.	
Business rules:	• Only super admin or users given appropriate permissions can update a K9 role.	
Implementation constraints and specifications:	• None	

INTEGRATED NGO MANAGEMENT SYSTEM	
Assumptions:	• None
Open issues:	• None

Table 102: Update K9 Role - Logical Use Case Narrative

2.8.13 View K9 Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View K9 Roles	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.13	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Handler	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how the handler views the K9 Roles available within the system. The system retrieves all K9 Roles from the database and displays them in a searchable and filterable table showing the role name, description, status, and available actions.	
Pre-condition:	The handler must be logged into the system.	
Trigger:	The handler wants to view the K9 roles on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The handler wants to view the K9 roles on the system.	**Step 2:** The system invokes UC 8.3 (View Dogs) to display the list of dogs.
	Step 3: The handler selects manage K9 roles.	**Step 4:** The system retrieves the K9 roles' details from the K9Role table.
		Step 5: The system displays a table showing all the current roles. The table includes: • Role • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Status • Actions
	Step 6: The handler can view the K9 roles.	
Alternate courses:	None	
Conclusion:	The system displays a screen with a table showing all the current roles.	
Post-condition:	The system retrieves and displays the K9 Roles stored in the K9Role table, allowing authorized users to view, search, and filter the available roles within the system.	
Business rules:	• Only super admin or users given appropriate permissions can view K9 roles.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 103: View K9 Roles - Logical Use Case Narrative

2.8.14 Reactivate K9 Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Reactivate K9 Role	**USE CASE TYPE**
Use case id:	8.14	Business Requirements: □
Priority:	High	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin reactivates an inactive K9 Role within the system. The system requests confirmation before restoring the role's active status. Once confirmed, the system updates the K9 Role, records the transaction in the audit log, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to reactivate a K9 role on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to reactivate a K9 role on the system.	**Step 2:** The system invokes UC 8.13 (View K9 Roles) to display a page with a table showing all the current roles.
	Step 3: The super admin selects reactivate on a specific role where the status is inactive.	**Step 4:** The system requests confirmation for the role reactivation.
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system changes the role's status to active in the K9Role table. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system records the transaction to the AuditLog table.
		Step 8: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin does not confirm. Go back to step 2.	
	[ALT] Step 6: The system fails to reactivate the K9 role and displays a toaster message. Terminate use case.	
Conclusion:	The inactive K9 role has been reactivated, and a confirmation toaster is shown.	
Post-condition:	The system updates the K9 Role's status to active in the K9Role table, making the role available for viewing, assignment to K9s, and management within the system.	
Business rules:	• Only super admin or users given appropriate permissions can reactivate K9 roles.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 104: Reactivate K9 Role - Logical Use Case Narrative

2.8.15 Deactivate K9 Role

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Deactivate K9 Role	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.15	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how the super admin deactivates an active K9 Role within the system. The system requests confirmation before deactivating the role's active status. Once confirmed, the system updates the K9 Role, records the transaction in the audit log, and displays a confirmation toaster.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin wants to deactivate a K9 role on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super admin wants to deactivate a K9 role on the system.	**Step 2:** The system invokes UC 8.13 (View K9 Roles) to display a page with a table showing all the current roles.
	Step 3: The super admin selects deactivate on a specific role where the status is active. [ALT]	**Step 4:** The system requests confirmation for the role deactivation.
	Step 5: The super admin confirms. [ALT]	**Step 6:** The system updates the role's status to inactive in the K9Role table. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system records the transaction to the AuditLog table.
		Step 8: The system displays a confirmation toaster.
Alternate courses:	**[ALT] Step 5:** The super admin does not confirm. Go back to step 2.	
	[ALT] Step 6: The system fails to deactivate the K9 role and displays a toaster message. Terminate use case.	
Conclusion:	The active K9 role has been deactivated, and a confirmation toaster is shown.	
Post-condition:	The system updates the K9 Role's status to inactive in the K9Role table, making the role unavailable in the active K9 Role listings while remaining available for viewing and management when inactive roles are displayed.	
Business rules:	• Only super admin or users given appropriate permissions can deactivate K9 roles.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 105: Deactivate K9 Role - Logical Use Case Narrative

2.8.16 View K9 Ancestry

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View K9 Ancestry	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	8.16	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Handler	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes how the handler views the ancestry of a selected K9 within the system. The system retrieves the selected K9's lineage information from the database and displays it as a graphical ancestry tree showing the current K9, its parents, grandparents, and great-grandparents.	
Pre-condition:	The handler must be logged into the system.	
Trigger:	The handler wants to view the ancestry tree of the dogs on the system.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The handler wants to view the ancestry tree of the dogs on the system.	**Step 2:** The system invokes UC 8.2 (View Dog) to display a pop-up showing all the dog's details.
	Step 3: The handler selects view ancestry.	**Step 4:** The system retrieves the dog's parent and lineage information from the K9 table and the dog's details from the associated Animal record. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 5: The system generates and displays the family tree of the selected dog.
	Step 6: The handler can view the family tree.	
Alternate courses:	[ALT] Step 4: The system fails to retrieve the K9 ancestry and displays a toaster message. Terminate use case.	
Conclusion:	The system generates and displays the family tree of the selected dog.	
Post-condition:	The system retrieves and displays the selected K9's ancestry from the K9 table, allowing the handler to view the K9's lineage up to the configured number of generations.	
Business rules:	• Only super admin or users given appropriate permissions can view a dog's ancestry.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 106: View K9 Ancestry - Logical Use Case Narrative

2.9 Rhino Subsystem Use Case Narratives

2.9.1 Add Rhino

<table>
<tr><th colspan="3">INTEGRATION NGO MANAGEMENT SYSTEM</th></tr>
<tr><td>Author:</td><td>Christopher Bailey</td><td>Date: 6 August 2026
Version: 1</td></tr>
<tr><td>Use case name:</td><td>Add Rhino</td><td rowspan="4">USE CASE TYPE
Business Requirements: □
System Analysis: ✓
System Design: □</td></tr>
<tr><td>Use case id:</td><td>9.1</td></tr>
<tr><td>Priority:</td><td>Medium</td></tr>
<tr><td>Source:</td><td>EPI-USE</td></tr>
<tr><td>Primary business actor (pba):</td><td colspan="2">Super Admin</td></tr>
<tr><td>Primary system actor (psa):</td><td colspan="2">• None</td></tr>
<tr><td>Other participating actors:</td><td colspan="2">• None</td></tr>
<tr><td>Other interested stakeholders:</td><td colspan="2">• None</td></tr>
<tr><td>Description:</td><td colspan="2">This use case describes the event in which a super admin adds a rhino. The system will display the rhino list and allow the super admin to add a new rhino. The system will request the rhino’s details and allow the super admin to enter the new rhino’s details. Once entered the system will validate the details entered before asking for confirmation to add the rhino. The use case ends once the super admin confirms and the success toaster is displayed.</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">Super Admin is logged into the system.</td></tr>
<tr><td>Trigger:</td><td colspan="2">The super admin requests to add a rhino.</td></tr>
<tr><td rowspan="3">Typical Course of Events:</td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>Step 1: The super admin requests to add a rhino.</td><td>Step 2: The system invokes use case 9.3 (View Rhinos)</td></tr>
<tr><td>Step 3: The super admin will select to add a rhino.</td><td>Step 4: The system requests the super admin to provide the following information:
• Name
• Species
• Weight (Optional)
• Gender</td></tr>
</table>

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Last Seen (Optional) • Date of Birth (Optional) • Approximate Location • Province (Auto filled based on approximate location) • Horn Status • Project (Optional) • Identification Notes • Photo (Optional)
	Step 5: The super admin enters the new rhino's details and selects to create. [ALT]	Step 6: The system validates the details entered. [ALT]
		Step 7: The system requests confirmation for the rhino to be added.
	Step 8: The super admin confirms that the rhino can be added. [ALT]	Step 9: The system adds the rhino to the following tables: • Animal • Rhino [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a success toaster informing the super admin that the rhino has been added successfully.
Alternate courses:	**[ALT] Step 5:** The super admin cancels their request to add a rhino. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 8: The super admin does not confirm. Go back to step 5.	
	[ALT] Step 9: The system fails to add the rhino into the relevant entities and displays an error message to the super admin.	
Conclusion:	The system displays a success toaster informing the super admin that the rhino has been added successfully.	
Post-condition:	• The rhino has been added successfully to the system. • The rhino can be assigned to projects.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Business rules:	• Only super admins or users with the required permissions can add rhinos to the system.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature. • Exact locations of rhinos are not saved for security reasons.
Assumptions:	• None
Open issues:	• None

Table 107: Add Rhino - Logical Use Case Narrative

2.9.2 View Rhino

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 6 August 2026** **Version: 1**
Use case name:	View Rhino	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	9.2	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin views a rhino's details. The system will display the rhinos and allow the super admin to filter and search for a rhino. The super admin will then select the rhino to view. The system will retrieve and display the selected rhino's details. The use case ends once the rhino's details are displayed.	
Pre-condition:	Super Admin is logged into the system.	
Trigger:	The super admin requests to view a rhino.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to view a rhino.	Step 2: The system invokes use case 9.3 (View Rhinos).
	Step 3: The super admin can filter the rhinos by status, species, gender, horn status or by searching by name or location. [ALT]	Step 4: The system filters and displays the rhino matching the criteria. [ALT]
	Step 5: The super admin selects to view the rhino.	Step 6: The system retrieves the selected rhino's details from the following tables: • Animal • Rhino

INTEGRATION NGO MANAGEMENT SYSTEM		
		[ALT]
		Step 7: The system displays the selected rhino's details including: • Name • Photo (If applicable) • Species • Gender • Weight (if applicable) • Approximate Location • Province • Last Seen (if applicable) • Age (if applicable) • Horn Status • Active Status • Dehorned Date (If applicable) • Identification notes
Alternate courses:	**[ALT] Step 3:** The super admin does not use the filters and selects a displayed rhino. Continue to step 6.	
	[ALT] Step 4: The system fails to find any rhinos matching the filters. Go back to step 3.	
	[ALT] Step 6: The system fails to retrieve the rhino from the relevant entities and displays an error toaster to the super admin.	
Conclusion:	The selected rhino's details are displayed.	
Post-condition:	The rhino's details are displayed.	
Business rules:	• Super admins or users with the required permissions can view a rhino.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 108: View Rhino - Logical Use Case Narrative

2.9.3 View Rhinos

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 7 August 2026** **Version: 1**
Use case name:	View Rhinos	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	9.3	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin views all rhinos on the system. The admin will navigate to and select the Animals tab and then select the Rhino tab. The system will load and display all the rhinos on the system. The use case ends once the system displays all the rhinos on the system.	
Pre-condition:	Super admin is logged into the system.	
Trigger:	Super admin requests to view all rhinos.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to view all rhinos.	
	Step 2: The super admin will select the animal tab.	
	Step 3: The super admin selects to view rhinos.	Step 4: The system retrieves all the rhinos on the system from the following tables: • Animal table • Rhino table [ALT]
		Step 5: The system displays the rhinos in a card format

INTEGRATION NGO MANAGEMENT SYSTEM		
		with the following details for each rhino: • Photo • Name • Gender • Species • Location • Age • Horn Status • Status
Alternate courses:	[ALT] Step 4: The system fails to retrieve the rhinos from the relevant entities and displays an error toaster to the super admin.	
Conclusion:	The super admin can view all the rhinos.	
Post-condition:	All rhinos on the system are displayed.	
Business rules:	• Only super admins or users with the required permissions can view all rhinos.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 109: View Rhino - Logical Use Case Narrative

2.9.4 Update Rhino

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 7 August 2026** **Version: 1**
Use case name:	Update Rhino	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	9.4	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin updates a rhino on the system. The system will display the rhino's details and request the rhino's updated details. Once entered the system will validate the updated details and ask the super admin for confirmation. Once confirmed the system will save the rhino's updated details. The use case ends once the success toaster is displayed and the rhino's updated details are successfully saved on the system.	
Pre-condition:	Super admin is logged in. Rhino already exists and is active.	
Trigger:	Super admin requests to update a rhino's details.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to update a rhino's details.	Step 2: The system invokes use case 9.2 (View Rhino)
	Step 3: The super admin selects to edit the rhino's details.	Step 4: The system populates the input fields for the selected rhino including: • Name • Species • Weight • Gender • Last Seen • Date of Birth

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Approximate Location • Province • Horn Status • Identification notes • Photo
	Step 5: The super admin enters the updated details and selects to update the rhino's details. [ALT]	Step 6: The system validates the details entered. [ALT]
		Step 7: The system requests confirmation for the rhino to be updated.
	Step 8: The super admin confirms that the rhino can be updated. [ALT]	Step 9: The system saves the updated details in the following tables: • Animal • Rhino [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a success toaster informing the super admin that the rhino was updated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin cancels their request to update the rhino. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful. Go back to step 5.	
	[ALT] Step 8: The super admin does not provide confirmation. Go back to step 5.	
	[ALT] Step 9: The system fails to update the rhino in the relevant entities and displays an error message to the super admin.	
Conclusion:	The system displays a success toaster informing the super admin that the rhino was updated successfully.	
Post-condition:	The rhino's details are updated.	
Business rules:	• Only super admins or users with the required permissions can update a rhino.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 110: Update Rhino - Logical Use Case Narrative

2.9.5 Deactivate Rhino

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 07 August 2026** **Version: 1**
Use case name:	Deactivate Rhino	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	9.5	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin deactivates a rhino on the system. The system will display the rhino's details and allow the super admin to deactivate the rhino. The system will then ask the super admin for confirmation. Once confirmed the system will deactivate the rhino. The use case ends once the success toaster is displayed and the rhino is successfully deactivated on the system.	
Pre-condition:	Super admin is logged in. Rhino already exists and is active.	
Trigger:	Super admin requests to deactivate a rhino.	
Typical Course of Events:	Actor Action	System Response
	Step 1: Super admin requests to deactivate a rhino.	Step 2: The system invokes use case 9.2 (View Rhino)
	Step 3: The super admin selects to deactivate the rhino.	Step 4: The system requests confirmation for the rhino to be deactivated.
	Step 5: The super admin confirms that the rhino can be deactivated. [ALT]	Step 6: The system sets the status of the rhino to deactivated in the Animal table. [ALT]
		Step 7: The system saves the transaction to the AuditLog table.

INTEGRATION NGO MANAGEMENT SYSTEM		
		Step 8: The system displays a success toaster informing the super admin that the rhino was deactivated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin does not provide confirmation. Use case terminates.	
	[ALT] Step 6: The system fails to update the status of the rhino in the relevant entities and displays an error toaster to the super admin.	
Conclusion:	The system displays a success toaster informing the super admin that the rhino was deactivated successfully.	
Post-condition:	• The rhino is deactivated on the system. • The rhino can no longer be assigned to projects.	
Business rules:	• Only super admins or users with the required permissions can deactivate a rhino.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 111: 9.5 Deactivate Rhino - Logical Use Case Narrative

2.10 Elephant Subsystem Use Case Narratives

2.10.1 Add Elephant

<table>
<tr><th colspan="3">INTEGRATED NGO MANAGEMENT SYSTEM</th></tr>
<tr><td>Author:</td><td>Christopher Bailey</td><td>Date: 14 March 2026
Version: 1.0</td></tr>
<tr><td>Use case name:</td><td>Add Elephant</td><td rowspan="4">USE CASE TYPE
Business Requirements: □
System Analysis: ✓
System Design: □</td></tr>
<tr><td>Use case id:</td><td>10.1</td></tr>
<tr><td>Priority:</td><td>High</td></tr>
<tr><td>Source:</td><td>EPI-USE</td></tr>
<tr><td>Primary business actor (pba):</td><td colspan="2">Super Admin</td></tr>
<tr><td>Primary system actor (psa):</td><td colspan="2">• None</td></tr>
<tr><td>Other participating actors:</td><td colspan="2">• None</td></tr>
<tr><td>Other interested stakeholders:</td><td colspan="2">• None</td></tr>
<tr><td>Description:</td><td colspan="2">This use case describes the event in which a super admin adds an elephant to the system. The super admin will select to add an elephant and enter the elephant’s information. The system will validate the details inputted before informing the super admin that the elephant has been successfully added. The use case ends once the elephant has successfully been added to the system.</td></tr>
<tr><td>Pre-condition:</td><td colspan="2">Super admin is logged into the system.</td></tr>
<tr><td>Trigger:</td><td colspan="2">Super Admin wants to add an elephant.</td></tr>
<tr><td rowspan="3">Typical Course of Events:</td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>Step 1: Super Admin wants to add an elephant.</td><td>Step 2: The system invokes use case 10.3 (View Elephants)</td></tr>
<tr><td>Step 3: The super admin will select add.</td><td>Step 4: The system displays the “Add Elephant” form to provide the following information:
• Name
• Species</td></tr>
</table>

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Weight • DateOfBirth • Gender • Tusk Status • Status • Last Seen • Identification notes • Approximate Location • Photo (Optional)
	Step 5: The super admin enters the details and selects add. [ALT]	**Step 6:** The system validates that all required fields are completed. [ALT]
		Step 7: The system requests confirmation for the elephant to be added.
	Step 8: The super admin confirms that the elephant can be added. [ALT]	**Step 9:** The system adds the elephant to the following tables: • Animal • Elephant [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a message informing the super admin that the elephant has been added successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects cancellation. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful; the super admin is prompted to enter the details into the empty fields. Go back to step 5.	
	[ALT] Step 8: The super admin selects back. Go back to step 6.	
	[ALT] Step 9:	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The system fails to add the elephant into the relevant entities and displays an error message to the super admin. Use case terminates.
Conclusion:	The system displays a message informing the super admin that the elephant has been added successfully.
Post-condition:	The elephant has been added successfully to the system.
Business rules:	• Only super admins can add elephants to the system.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 112: Add Elephant - Logical Use Case Narrative

2.10.2 View Elephant

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Elephant	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	10.2	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• Admin	
Description:	This use case describes the event in which a super admin wants to view an elephant on the system. The system will retrieve and display all the elephants on the system before the super admin searches using the elephant's name and selects an elephant to view. The system will retrieve the selected elephant's details and display them to the super admin. The use case ends once the system displays the selected elephant's details.	
Pre-condition:	Super admin is logged into the system.	
Trigger:	Super Admin wants to view an elephant.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super Admin wants to view an elephant.	**Step 2:** The system invokes use case 10.3 (View Elephants).
	Step 3: The super admin searches for an elephant using the elephant's name. [ALT]	**Step 4:** The system filters and displays the elephant matching the criteria searched. [ALT]
	Step 5: The super admin selects the elephant.	**Step 6:** The system retrieves the selected elephant's details from the following tables:

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Animal • Elephant
		Step 7: The system displays the "View Elephant" popup displaying the selected elephant's details including: • Name • Species • Weight • Age • Gender • Tusk Status • Status • Last Seen • Identification notes • Approximate Location • Photo (Optional)
	Step 8: Super admin can view an elephant.	
Alternate courses:	[ALT] Step 3a: The super admin selects a displayed elephant.	
	[ALT] Step 3b: The super admin filters the displayed elephants using the filter option.	
	[ALT] Step 4: The system fails to find any elephants with a matching name. Go back to step 3.	
Conclusion:	Super admin can view an elephant.	
Post-condition:	The elephant's details are displayed.	
Business rules:	• Super admins can view an elephant. • Admins can view an elephant on projects they are in.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	

INTEGRATED NGO MANAGEMENT SYSTEM	
Open issues:	• None

Table 113: View Elephant - Logical Use Case Narrative

2.10.3 View Elephants

<table>
<tr><th colspan="4">INTEGRATED NGO MANAGEMENT SYSTEM</th></tr>
<tr><td>Author:</td><td colspan="2">Christopher Bailey</td><td>Date: 14 March 2026
Version: 1.0</td></tr>
<tr><td>Use case name:</td><td colspan="2">View Elephants</td><td rowspan="4">USE CASE TYPE
Business Requirements: □
System Analysis: ✓
System Design: □</td></tr>
<tr><td>Use case id:</td><td colspan="2">10.3</td></tr>
<tr><td>Priority:</td><td colspan="2">High</td></tr>
<tr><td>Source:</td><td colspan="2">EPI-USE</td></tr>
<tr><td>Primary business actor (pba):</td><td colspan="3">Super Admin</td></tr>
<tr><td>Primary system actor (psa):</td><td colspan="3">• None</td></tr>
<tr><td>Other participating actors:</td><td colspan="3">• None</td></tr>
<tr><td>Other interested stakeholders:</td><td colspan="3">• None</td></tr>
<tr><td>Description:</td><td colspan="3">This use case describes the event in which a super admin wants to view all the elephants on the system. The system will retrieve all the elephants on the system and display them to the super admin. The use case ends once all the elephants are displayed.</td></tr>
<tr><td>Pre-condition:</td><td colspan="3">Super admin is logged into the system.</td></tr>
<tr><td>Trigger:</td><td colspan="3">Super Admin wants to view all the elephants.</td></tr>
<tr><td rowspan="5">Typical Course of Events:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: Super Admin wants to view all the elephants.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The super admin will select the animal tab.</td><td colspan="2"></td></tr>
<tr><td>Step 3: The super admin selects elephant.</td><td colspan="2">Step 4: The system retrieves all the elephants on the system from the following tables:
• Animal table
• Elephant table</td></tr>
<tr><td></td><td colspan="2">Step 5: The system displays the “Elephant View”,</td></tr>
</table>

INTEGRATED NGO MANAGEMENT SYSTEM		
		displaying the following details for each elephant: • Name • Age • Species • Approximate Location
Alternate courses:	None	
Conclusion:	The system displays the "Elephant View" with all the elephants on the system.	
Post-condition:	All elephants on the system are displayed.	
Business rules:	• Only super admins can view all elephants.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 114: View Elephants - Logical Use Case Narrative

2.10.4 Update Elephant

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update Elephant	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	10.4	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin wants to update an elephant on the system. The system allows the super admin to select an elephant and then update the elephant's details. The system will validate the details entered before asking the admin to confirm the updated details. Once confirmed the system will inform the super admin that the elephant's details have been successfully updated. The use case ends once the elephant's updated details have been successfully captured on the system.	
Pre-condition:	Super admin is logged into the system.	
Trigger:	Super Admin wants to update an elephant.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super Admin wants to update an elephant.	**Step 2:** The system invokes use case 10.2 (View Elephant)
	Step 3: The super admin selects update.	**Step 4:** The system displays the "Update Elephant" form to update the following information: • Name • Species

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Weight • DateOfBirth • Gender • Tusk Status • Status • Last Seen • Identification notes • Approximate Location • Photo (Optional)
	Step 5: The super admin enters the updated details. [ALT]	**Step 6:** The system validates that all required fields are completed. [ALT]
		Step 7: The system requests confirmation for the elephant to be updated.
	Step 8: The super admin confirms that the elephant can be updated. [ALT]	**Step 9:** The system saves the updated details in the following tables: • Animal • Elephant [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a message informing the super admin that the elephant was updated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects cancel. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful; the super admin is prompted to enter the details into the empty fields. Go back to step 5.	
	[ALT] Step 8: The super admin selects back. Go back to step 5.	
	[ALT] Step 9:	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The system fails to update the elephant in the relevant entities and displays an error message to the super admin. Use case terminates.
Conclusion:	The system displays a message informing the super admin that the elephant was updated successfully.
Post-condition:	The elephant's details are updated.
Business rules:	• Only super admins can update an elephant.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 115: Update Elephant - Logical Use Case Narrative

2.10.5 Deactivate Elephant

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Deactivate Elephant	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	10.5	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin wants to deactivate an elephant on the system. The system will display the elephant and allow the super admin to deactivate the elephant. The system will ask the super admin to confirm the deactivation before the elephant is deactivated. The use case ends once the elephant has been successfully deactivated on the system.	
Pre-condition:	Super admin is logged into the system.	
Trigger:	Super Admin wants to deactivate an elephant.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super Admin wants to deactivate an elephant.	**Step 2:** The system invokes use case 10.2 (View Elephant)
	Step 3: The super admin selects deactivate.	**Step 4:** The system requests confirmation for the elephant to be deactivated.
	Step 5: The super admin confirms that the elephant can be deactivated. [ALT]	**Step 6:** The system sets the status of the elephant to deactivated in the Animal table. [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system saves the transaction to the AuditLog table.
		Step 8: The system displays a message informing the super admin that the elephant was deactivated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects cancel. Use case terminates.	
	[ALT] Step 6: The system fails to update the status of the elephant in the relevant entities and displays an error message to the super admin. Use case terminates.	
Conclusion:	The system displays a message informing the super admin that the elephant was deactivated successfully.	
Post-condition:	The elephant is deactivated on the system.	
Business rules:	• Only super admins can deactivate an elephant.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 116: Deactivate Elephant - Logical Use Case Narrative

2.11 Messages Subsystem Use Case Narratives

2.11.1 Create Message

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Create Message	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	11.1	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• Group Members (ERA)	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a user creates and sends a message on a group. The user will navigate to the messages tab after which the system retrieves all the groups the user is currently in. The user will select a group and the system will retrieve all the messages for that group from the Firebase Firestore database. The user will write their message and send it to the group for other group members to see. The use case ends once the new message is sent and stored in the Firebase Firestore database.	
Pre-condition:	User must be logged into the system and on their account.	
Trigger:	The user requests to send a message on a group.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user requests to send a message on a group.	**Step 2:** The system invokes use case 11.2 (View Messages)
	Step 3: The user writes their message in the message	**Step 4:** The system sends the message on the specific group

INTEGRATED NGO MANAGEMENT SYSTEM		
	input field and selects to send their message. [ALT]	and stores the message in the appropriate group's Firestore message collection. [ALT]
		Step 5: All clients subscribed to the group's messages collection receive the new message instantly and the chat UI updates automatically.
Alternate courses:	[ALT] Step 3a: The user decides not to send a message and exits the Messages tab. Use case terminates.	
	[ALT] Step 3b: The user selects the microphone function and records an audio message.	
	[ALT] Step 4: The system saves the audio, and the message is sent to the Firestore service, where the audio URL and message details are stored in the appropriate group's message collection.	
Conclusion:	The user's message is saved in the Firebase Firestore database and viewable by other group members.	
Post-condition:	The user's message is successfully saved and appears for other users to view.	
Business rules:	• Users can only send messages on groups they are a part of.	
Implementation constraints and specifications:	• The user requires internet access to access this feature.	
Assumptions:	• None	
Open issues:	None	

Table 117: Create Message - Logical Use Case Narrative

2.11.2 View Messages

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Messages	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	11.2	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	User	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a user views messages on a group. The user will navigate to the messages tab after which the system retrieves all the groups the user is currently in. The user will select a group and the system will retrieve all the messages for that group from the Firebase Firestore database. The use case ends once all the messages for the selected group are displayed for the user.	
Pre-condition:	User must be logged into the system and on their account.	
Trigger:	The user requests to view a message on a group.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user requests to view a message on a group.	
	Step 2: The user navigates to and selects the Messages tab. [ALT]	**Step 3:** The system retrieves all the user's groups from the UserGroup table. [ALT]
		Step 4: The system displays the messages screen.

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The user clicks the group they wish to view. [ALT]	**Step 6:** When a group is selected the system retrieves messages from the Firebase Firestore database.
		Step 7: The system displays messages for the selected group.
Alternate courses:	**[ALT] Step 2:** The user is using a mobile device such as a phone. The user will select the menu (hamburger) icon and then the Messages option. Continue to step 3.	
	[ALT] Step 3: The system fails to retrieve the user's groups. The system will displays an error toaster based on the status code.	
	[ALT] Step 5: The user does not select a group and exits the messages tab. Use case terminates.	
Conclusion:	The groups messages are displayed.	
Post-condition:	The user can view messages from the selected group.	
Business rules:	• Users can only view messages in groups they are a part of.	
Implementation constraints and specifications:	• The user requires internet access to access this feature.	
Assumptions:	• None	
Open issues:	None	

Table 118: View Messages - Logical Use Case Narrative

2.12 Report Subsystem Use Case Narratives

2.12.1 Generate Custom Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.1	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	Generate Custom Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None.	
Other interested stakeholders:	None.	
Description:	A user with appropriate permissions creates a new report by selecting specific data fields and filters to retrieve tailored information from the system.	
Pre-condition:	The user must be logged in with "Generate Report" permissions.	
Trigger:	The user selects the "Create New Custom Report" option in the Reporting dashboard.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user enters a name and description for the report in the provided fields.	**Step 2:** The system provides a list of available data categories that can be used to make custom reports.
	Step 3: The user selects a category and selects specific fields.	**Step 4:** The system allows users to define filter criteria.
	Step 5: The user selects the "Generate" function	**Step 6:** The system validates the query. [ALT]
		Step 7: The system retrieves all required data from the database.
		Step 8: The system displays the report and stores a copy of

INTEGRATED NGO MANAGEMENT SYSTEM		
		the criteria used in the Report and ReportConfiguration tables: **Report table:** • ReportID [PK] • Name • Description • Date **ReportConfiguration Table:** • ReportConfigurationID [PK] • ReportID [FK • SelectedFields • Filter • ReportStartDate • ReportEndDate • Type
	Step 9: The User downloads the report in PDF format. [ALT]	**Step 10:** Custom report creation transaction is saved to the Audit Log. **AuditLog:** • AuditLogID [PK] • UserID [FK] • Timestamp • LogEntry
Alternate courses:	**[ALT] Step 6:** The user selects conflicting filters that return no data, the system warns the user and suggests adjusting the criteria.	
	[ALT] Step 9: The user chooses not to download the report.	
Conclusion:	A custom report is created.	
Post-condition:	A new entry is saved in the Report Table.	
Business rules:	• Role Based Access	

INTEGRATED NGO MANAGEMENT SYSTEM	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None.
Open issues:	• None.

Table 119: Generate Custom Report - Logical Use Case Narrative

2.12.2 View Custom Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.2	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	View Custom Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None.	
Other interested stakeholders:	None.	
Description:	A user selects an existing custom report configuration to view the most up-to-date data presented in that specific format.	
Pre-condition:	At least one custom report must already exist in the system.	
Trigger:	The user selects an existing report from the "Saved Reports" list.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects a custom report from the list of previously defined reports.	**Step 2:** The system retrieves the saved configuration from the Report and ReportConfiguration tables. **Report table:** • ReportID [PK] • Name • Description • Date **ReportConfiguration Table:** • ReportConfigurationID [PK] • ReportID [FK • SelectedFields

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Filter • ReportStartDate • ReportEndDate • Type
		Step 3: The system executes the query against the relevant data tables. [ALT]
		Step 4: The system displays the report
	Step 5: The User Downloads the report in PDF format. [ALT]	
Alternate courses:	**[ALT] Step 3:** No valid data is available; user is requested to update the report. (Use Case 12.3)	
	[ALT] Step 5: The user chooses not to download the report.	
Conclusion:	The user views the current data based on the saved report parameters.	
Post-condition:	The database reflects the updated report settings.	
Business rules:	• Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None.	
Open issues:	• None.	

Table 120: View Custom Report - Logical Use Case Narrative

2.12.3 Update Custom Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.3	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	Update Custom Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None	
Other interested stakeholders:	None.	
Description:	A user modifies the parameters of an existing custom report, such as adding a new column or changing the date range.	
Pre-condition:	The report must already exist on the system.	
Trigger:	The user selects "Edit Report" on a saved report configuration.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the custom report they wish to update.	**Step 2:** The system retrieves the saved configuration from the Report and ReportConfiguration tables. **Report table:** • ReportID [PK] • Name • Description • Date **ReportConfiguration Table:** • ReportConfigurationID [PK] • ReportID [FK • SelectedFields • Filter • ReportStartDate • ReportEndDate

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Type
		Step 3: The system displays the configuration to the user and allows the user to edit these fields.
	Step 4: The user modifies the provided fields.	**Step 5:** The system validates the new parameters. [ALT]
	Step 6: The user selects the "Save" Function.	**Step 7:** The system saves the changes to the custom report in the Report and ReportConfiguration tables and displays a confirmation message to the user. **Report table:** • ReportID [PK] • Name • Description • Date **ReportConfiguration Table:** • ReportConfigurationID [PK] • ReportID [FK • SelectedFields • Filter • ReportStartDate • ReportEndDate • Type
		Step 8: The system stores the transaction to the Audit Log. **AuditLog:** • AuditLogID [PK] • UserID [FK] • Timestamp • LogEntry
Alternate courses:	**[ALT] Step 5:**	

INTEGRATED NGO MANAGEMENT SYSTEM	
	The system could not successfully validate the provided parameters and prompts the user to re-enter those fields.
Conclusion:	The report configuration is successfully modified.
Post-condition:	None.
Business rules:	• Role Based Access
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None.
Open issues:	• None.

Table 121: Update Custom Report - Logical Use Case Narrative

2.12.4 Remove Custom Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.4	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	Remove Custom Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None.	
Other interested stakeholders:	None.	
Description:	An administrator removes the custom report and its associated configuration from the database.	
Pre-condition:	The report must already exist on the system.	
Trigger:	The user selects "Remove Report" on a saved report configuration.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The administrator selects the "Remove" function when viewing a custom report.	**Step 2:** The system requests the user to confirm their intentions.
	Step 3: The user confirms the deactivation. [ALT]	**Step 4:** The removes the report and its configuration from the Report and Report Configuration tables. **Report table:** • ReportID [PK] • Name • Description • Date **ReportConfiguration Table:** • ReportConfigurationID [PK]

INTEGRATED NGO MANAGEMENT SYSTEM		
		• ReportID [FK • SelectedFields • Filter • ReportStartDate • ReportEndDate • Type
		Step 5: The system saves the Deactivation Transaction to the Audit Log. **AuditLog:** • AuditLogID [PK] • UserID [FK] • Timestamp • LogEntry
Alternate courses:	**[ALT] Step 3:** The user does not confirm the removal; they are redirected to viewing the report.	
Conclusion:	Users can no longer see the custom report in the list of custom reports.	
Post-condition:	Custom report record is removed from the Report table.	
Business rules:	• Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None.	
Open issues:	• None.	

Table 122: Remove Custom Report - Logical Use Case Narrative

2.12.5 View Bike Impact Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.5	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	View Bike Impact Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None.	
Other interested stakeholders:	Communities, Donors	
Description:	The user accesses a report that visualizes the impact of bike drops on communities by analysing historical attendance and learner performance data.	
Pre-condition:	User must be logged in. Bike location and community data must exist.	
Trigger:	The user selects "Bike Impact Report" from the reporting dashboard.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the desired period and specific community.	**Step 2:** The system retrieves data from the Inventory Transaction table for InventoryType bikes for projects of type bike mathing the provided criteria and information from the Project Impact table. **InventoryTransaction:** • InvetoryTransactionID [PK] • InventoryID [FK] • WarehouseID [FK] • ProjectID [FK]

INTEGRATED NGO MANAGEMENT SYSTEM		
		• TransactionType • ReferenceID • ReferenceType • TransactionDate • Quantity • DocumentID [FK] **ProjectImpact:** • ProjectImpactID • Description • ImpactType • GradeImprovement • AttendanceImprovement [ALT]
	Step 3: The user selects "View Report".	**Step 4:** The system displays the impact data visually on the screen
	Step 5: The user downloads the Report in PDF form. [ALT]	
Alternate courses:	**[ALT] Step 2:** No data could be found for the provided period or community; the system displays an error message.	
	[ALT] Step 5: The user decides not to download the report.	
Conclusion:	The user views the social impact metrics of the bicycle donation program.	
Post-condition:	None.	
Business rules:	• Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None.	
Open issues:	• None.	

Table 123: View Bike Impact Report - Logical Use Case Narrative

2.12.6 View Budget Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.6	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	View Budget Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None.	
Other interested stakeholders:	Financial Officers	
Description:	The user views a financial summary to monitor spending, compare actual costs against budgets, and support financial decision-making.	
Pre-condition:	User must be logged in. Budget data must exist.	
Trigger:	The user selects "Budget Report" and chooses a project.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the desired project and/or financial year.	**Step 2:** The system retrieves data from the Budget and Expense tables. **Budget:** • BudgetID [PK] • ProjectID [FK] • DateCreated • AllocatedAmount • CurrentBalance **Expense:** • ExpenseID [PK] • DocumentID [FK] • ExpenseItemName • Amount • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
		[ALT]
	Step 3: The user selects "View Report".	**Step 4:** The system displays the budget data visually on the screen
	Step 5: The user downloads the Report in PDF form. [ALT]	
Alternate courses:	**[ALT] Step 2:** No data could be found for the provided period or project; the system displays an error message.	
	[ALT] Step 5: The user decides not to download the report.	
Conclusion:	The user assesses the financial health of the project.	
Post-condition:	None.	
Business rules:	• Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None.	
Open issues:	• None.	

Table 124: View Budget Report - Logical Use Case Narrative

2.12.7 View Rhino Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.7	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	View Rhino Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None.	
Other interested stakeholders:	Conservation Team	
Description:	The user views detailed rhino information to monitor conservation efforts. This includes location, population, de-horning and general animal information.	
Pre-condition:	User must be logged in. Rhino data must exist.	
Trigger:	User selects "View Rhino Report" from the menu.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the desired location for the report.	**Step 2:** The system retrieves data from the Rhino and Animal tables. **Animal:** • AnimalID [PK] • Name • Species • Weight • DateOfBirth • Gender • ActiveStatus • LastSeen • IdentificationNotes • ApproximateLocation • Photo

INTEGRATED NGO MANAGEMENT SYSTEM		
		Rhino: • RhinoID [PK] • AnimalID [FK] • HornStatus [ALT]
	Step 3: The user selects "View Report".	**Step 4:** The system displays the rhino data visually on the screen
	Step 5: The user downloads the Report in PDF form. [ALT]	
Alternate courses:	**[ALT] Step 2:** No data could be found for the provided location; the system displays an error message.	
	[ALT] Step 5: The user decides not to download the report.	
Conclusion:	The user assesses overall Rhino information to gain a better understanding of conservation efforts.	
Post-condition:	None.	
Business rules:	• Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None.	
Open issues:	• None.	

Table 125: View Rhino Report - Logical Use Case Narrative

2.12.8 View Elephant Report

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 26 March 2026 **Version:** 2.0
Use case name:	12.8	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	View Elephant Report	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	None.	
Other participating actors:	• None.	
Other interested stakeholders:	Conservation Team	
Description:	The user views detailed elephant information to monitor conservation efforts. This includes location, population, tusk-status and general animal information.	
Pre-condition:	User must be logged in. Elephant data must exist.	
Trigger:	User selects "View Elephant Report" from the menu.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The user selects the desired location for the report.	**Step 2:** The system retrieves data from the Elephant and Animal tables. **Animal:** • AnimalID [PK] • Name • Species • Weight • DateOfBirth • Gender • ActiveStatus • LastSeen • IdentificationNotes • ApproximateLocation • Photo **Elephant**

INTEGRATED NGO MANAGEMENT SYSTEM		
		• ElephantID [PK] • AnimalID [FK] • TuskStatus [ALT]
	Step 3: The user selects "View Report".	**Step 4:** The system displays the elephant data visually on the screen
	Step 5: The user downloads the Report in PDF form. [ALT]	
Alternate courses:	**[ALT] Step 2:** No data could be found for the provided location; the system displays an error message.	
	[ALT] Step 5: The user decides not to download the report.	
Conclusion:	The user assesses overall elephant information to gain a better understanding of conservation efforts.	
Post-condition:	None.	
Business rules:	• Role Based Access	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None.	
Open issues:	• None.	

Table 126: View Elephant Report - Logical Use Case Narrative

2.12.9 View K9 Report

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date:** 8 August 2026 **Version:** 1.0
Use case name:	View K9 Report	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	12.9	
Priority:	Medium	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin generates and views a K9 report. The super admin will navigate to the Admin tab in the navbar and will then select Reports and then the K9 Report option. The system will retrieve the K9 report's details and display the report to the super admin. The super admin will then have the option to download the report as a pdf or export as a CSV. The use case ends once the super admin downloads the report.	
Pre-condition:	Super Admin is logged into the system.	
Trigger:	The super admin requests to view the K9 report.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin requests to view the K9 report.	
	Step 2: The super admin will navigate to the Admin tab, select Reports and then select K9 Report. [ALT]	
	Step 3: The super admin selects a reporting period using the predefined options or selecting between two dates. [ALT]	Step 4: The system retrieves the K9 reports information from the following tables: • K9

INTEGRATION NGO MANAGEMENT SYSTEM		
		• Animal • K9Role • K9K9Role • K9User • User • AnimalProject • Project • Inventory • InventoryType • Warehouse [ALT]
		Step 5: The system saves the transaction to the AuditLog table.
	Step 6: The super admin selects to download the PDF. [ALT]	Step 7: The system generates a pdf report and downloads the report to the user's device.
		Step 8: The system saves the transaction to the AuditLog table.
Alternate courses:	**[ALT] Step 2:** The super admin is using a mobile device such as a phone. The super admin will select the menu (hamburger) icon and then Admin, Reports and K9 Report. Continue to step 3.	
	[ALT] Step 3: The super admin does not select a reporting period and the default "Year to date" is used. Continue to step 4.	
	[ALT] Step 4: The system fails to retrieve the report data from the relevant entities and displays an error message to the super admin.	
	[ALT] Step 6a: The super admin does not download the report as a pdf, just views it. [ALT] Step 6b: The super admin selects to export the report as a CSV. The system generates a csv file and downloads the file to the user's device.	
Conclusion:	The report pdf is downloaded on the super admin's device and ready to be viewed.	
Post-condition:	The report has been generated and downloaded as a pdf.	

INTEGRATION NGO MANAGEMENT SYSTEM	
Business rules:	• Only super admins or users with the required permissions can view K9 reports.
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.
Assumptions:	• None
Open issues:	• None

Table 127: View K9 Report - Logical Use Case Narrative

2.12.10 View Machine Learning Dashboard

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 25 March 2026 **Version: 1.0**
Use case name:	12.10	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	View Machine Learning Dashboard	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Administrator	
Primary system actor (psa):	• None.	
Other participating actors:	• User	
Other interested stakeholders:	None.	
Description:	This use case describes the event where a user wants to view all current Machine Learning Predictions in one place. Once the User requests to view this information, the Machine Learning Model runs with the most current data and displays it for the user.	
Pre-condition:	There should be enough data to accurately run the model,	
Trigger:	User selects the Machine Learning Dashboard.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: A user such as the administrator requests to view the Machine Learning dashboard.	**Step 2:** The system runs the most recent data for Risk Predictions, Team Predictions and budget Predictions through the Sci-kit learn model. [ALT]
		Step 3: The system displays the predictions by category and project.
Alternate courses:	**[ALT] Step 1:** The model fails to process all predictions; an error message is shown.	
Conclusion:	The dashboard is successfully displayed to the user.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Post-condition:	The dashboard is successfully displayed to the user.
Business rules:	• System must have Machine Learning Predictions.
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.
Assumptions:	• None.
Open issues:	• None.

Table 128: View Machine Learning Dashboard - Logical Use Case Narrative

2.13 System Subsystem Use Case Narratives

2.13.1 Backup System

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 25 March 2026 **Version: 1.0**
Use case name:	13.1	**USE CASE TYPE**
Use case id:	Backup System	Business Requirements: □
Priority:	Low	System Analysis: ✓
Source:	EPI-USE	System Design: □
Primary business actor (pba):	Time	
Primary system actor (psa):	• None.	
Other participating actors:	• Super Administrator	
Other interested stakeholders:	• None.	
Description:	This use case describes the event where a super administrator makes a backup of the database in its current form. The action is also initiated automatically on a weekly basis, always keeping 4 automatic backups. The super administrator will select to create a backup, which prompts the system to make a backup of the database directory. The use case concludes when the database has been backed up and a success tis displayed to the super administrator.	
Pre-condition:	The super administrator must be logged in.	
Trigger:	Time initiates an automatic backup.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The system automatically initiates a backup at 03:00 every Sunday morning. **[ALT]**	**Step 2:** The system displays a message to all users logged in that they will be logged out until the system backup is finalised.
		Step 3: The system creates a copy of the database directory and adds details pertaining to the backup to

INTEGRATED NGO MANAGEMENT SYSTEM		
		the Backup table with the following attributes: **Backup:** • BackupID [PK] • Date • FilePath • Size
		Step 4: The system deletes the oldest backup to keep a maximum of 5 backups. [ALT]
		Step 5: System stores backup transaction to Audit Log **AuditLog:** • AuditLogID [PK] • UserID [FK] • Timestamp • LogEntry
Alternate courses:	**[ALT] Step 1:** The super administrator manually initiates a backup at any time.	
	[ALT] Step 4: There are less than 5 backups, no backup is removed.	
Conclusion:	User is successfully authenticated and granted access.	
Post-condition:	An active session is created for the user.	
Business rules:	• Role Based Access • Ability to Backup Database.	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	
Assumptions:	• None.	
Open issues:	• None.	

Table 129: Backup System - Logical Use Case Narrative

2.13.2 Restore System

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Stephan Bredenhann	**Date:** 25 March 2026 **Version: 1.0**
Use case name:	13.2	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	Restore System	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Administrator	
Primary system actor (psa):	• None.	
Other participating actors:	• None.	
Other interested stakeholders:	None.	
Description:	This use case describes the event where the super administrator wants to restore the system's database from a previous backup. The system will display all available backups stored for the admin user to select from. The admin user will select the version they wish to restore to and select to submit. The system will display a confirmation message prompting the admin user to confirm the restoration of the system to that backup version. Upon receiving confirmation from the super administrator, the system will retrieve the backup version and restore the database to that previous state.	
Pre-condition:	The super administrator must be logged in.	
Trigger:	The super administrator requests to restore the system to a previous state.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: The super administrator requests to restore the system to a previous state.	**Step 2:** The system displays a list of stored backups referenced in the Backup table. **Backup:** • BackupID [PK] • Date • FilePath

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Size
	Step 3: The super administrator selects the backup version they would like to restore to.	**Step 4**: The system displays a confirmation message to the super administrator.
	Step 5: The super administrator confirms their intention to restore the system to a previous backup. **[ALT]**	**Step 6:** The system Logs out all users before the Restore takes place.
		Step 7: The system replaces the current database directory with the version selected by the super administrator. **[ALT]**
		Step 8: The system records the transaction to the audit log. **AuditLog:** • AuditLogID [PK] • UserID [FK] • Timestamp • LogEntry
Alternate courses:	**[ALT] Step 5:** The super administrator does not provide confirmation to restore the system. Process is cancelled.	
	[ALT] Step 6: The system fails to restore to a previous state.	
Conclusion:	User is successfully authenticated and granted access. The system displays an error message, and the use case terminates.	
Post-condition:	An active session is created for the user.	
Business rules:	• Role Based Access • Ability to Backup Database.	
Implementation constraints and specifications:	• Must be completed before project day. • Must be user-friendly and easy to use.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Assumptions:	• None.
Open issues:	• None.

Table 130: Restore System - Logical Use Case Narrative

2.13.3 View Audit Log

INTEGRATION NGO MANAGEMENT SYSTEM		
Author:	Nicholas Bailey	**Date: 9 August 2026** **Version: 1**
Use case name:	View Audit Log	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	13.3	
Priority:	Low	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes when the super admin views audit logs recorded within the system. The system retrieves and displays audit log records in a table, including details such as the user who performed the action, the action performed, the affected entity, and additional information about the activity. The system also provides audit statistics, searching, and pagination functionality to assist the super admin in monitoring system activity. The use case ends when the super admin can view and review recorded system actions and their details.	
Pre-condition:	The super admin must be logged into the system.	
Trigger:	The super admin requests to view the full audit log.	
Typical Course of Events:	Actor Action	System Response
	Step 1: The super admin wants to view all audit logs.	
	Step 2: The super admin hovers the admin tab and then selects audit logs. [ALT]	**Step 3:** The system retrieves all the audit logs on the system from the AuditLog table.
		Step 4: The system displays the Audit Logs page showing all the retrieved audit logs in a table: • Date

INTEGRATION NGO MANAGEMENT SYSTEM		
		• User • Action • Entity • Details [ALT]
	Step 5: The super admin can view the audit logs.	
Alternate courses:	ALT-Step 2: User is on mobile and selects the menu icon. Go to step 3.	
	ALT-Step 4: The system fails to load the audit log data and displays an error message.	
Conclusion:	The system has successfully retrieved a list of all the audit logs within the system.	
Post-condition:	All the audit logs within the system are displayed.	
Business rules:	• Only super admins or users given appropriate permissions can view audit logs. • Ability to Backup Database.	
Implementation constraints and specifications:	• None	
Assumptions:	• None	
Open issues:	• None	

Table 131: View Audit Log - Logical Use Case Narrative

2.14 Budget Subsystem Use Case Narratives

2.14.1 View Project Budget

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Project Budget	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	14.1	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• Admin	
Description:	This use case describes the event in which a super admin wants to view a project budget on the system. The system will retrieve and display all the budgets of the active projects on the system before the super admin selects a budget to view. The system will retrieve the selected project's budget as well as expenses and display them to the super admin. The use case ends once the system displays the selected project's budget.	
Pre-condition:	Super admin is logged into the system.	
Trigger:	Super Admin wants to view a project budget.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super admin wants to view a project budget.	**Step 2:** The system invokes use case 15.3 (View Project Budgets).
	Step 3: The super admin selects the project budget.	**Step 4:** The system retrieves the selected project's budget and expenses from the following tables:

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Project • Budget • Expense • ProjectExpense
		Step 7: The system displays the "View Project Budget", displaying the selected project budget as well as expenses.
	Step 8: Super admin can view a project budget.	
Alternate courses:	None	
Conclusion:	Super admin can view a project budget.	
Post-condition:	The project budget is displayed.	
Business rules:	• Financial data such as this is privileged, and thus only persons with appropriate roles may view it.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 132: View Project Budget - Logical Use Case Narrative

2.14.2 View Project Budgets

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Project Budgets	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	14.2	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin wants to view all the project budgets on the system. The system will retrieve all the project budgets on the system and display them to the super admin. The use case ends once all the project budgets are displayed.	
Pre-condition:	Super admin is logged into the system.	
Trigger:	Super Admin wants to view all the project budgets.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super Admin wants to view all the project budgets.	
	Step 2: The super admin will select the budget tab.	
	Step 3: The super admin selects view budget.	**Step 4:** The system retrieves all the active project budgets on the system from the following tables: • Project • Budget
		Step 5: The system displays the "Project Budget View",

INTEGRATED NGO MANAGEMENT SYSTEM		
		displaying the following details for each project: • Name • Budget
	Step 6: The super admin can view all the project budgets.	
Alternate courses:	None	
Conclusion:	The super admin can view all the project budgets.	
Post-condition:	All project budgets on the system are displayed.	
Business rules:	• Financial data such as this is privileged, and thus only persons with appropriate roles may view it.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 133: View Project Budgets - Logical Use Case Narrative

2.14.3 Log An Expense

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Log An Expense	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	14.3	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which an expense is logged. The system will display the project's budget and allow the super admin to add an expense by entering the expenses details and the system will validate the inputted data before asking for confirmation. Once confirmed, the system will capture the expense. The use case ends once the expense is captured on the system.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin wants to log an expense.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super admin wants to log an expense.	**Step 2:** The system invokes use case 15.2 (View Project Budget)
	Step 3: The super admin will select to add an expanse.	**Step 4:** The system displays the "Log Expense" form to provide the following information: • Name • Description • Amount

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin enters the expense's details and selects add. [ALT]	**Step 6:** The system validates that all required fields are completed. [ALT]
		Step 7: The system requests confirmation for the expense to be logged.
	Step 8: The super admin confirms that the expense can be logged. [ALT]	**Step 9:** The system adds the expense to the following tables: • Expense • ProjectExpense [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a message informing the super admin that the expense has been logged successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects cancel. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful; the super admin is prompted to enter the details into the empty fields. Go back to step 5.	
	[ALT] Step 8: The super admin selects back. Go back to step 5.	
	[ALT] Step 9: The system fails to add the expense to the relevant entities and displays an error message to the super admin. Use case terminates.	
Conclusion:	The system displays a message informing the super admin that the expense has been logged successfully.	
Post-condition:	The expense is added to the system.	
Business rules:	• Financial data such as this is privileged, and thus only persons with appropriate roles may view it.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	

INTEGRATED NGO MANAGEMENT SYSTEM	
	• There should be sufficient budget to log an expense, if this is not the case, the user will be informed, and the expense will not be logged.
Assumptions:	• None
Open issues:	• None

Table 134: Log An Expense – Logical Use Case Narrative

2.14.4 Update An Expense

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Update An Expense	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	14.4	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which the super admin updates an expense. The system will display the expense's details and allow the super admin to update its details. The super admin will enter the updated details, and the system will validate the updated details before asking the super admin to confirm the updated details. Once confirmed the system will inform the super admin that the updated details have been successfully captured. The use case ends once the expenses updated details are captured on the system.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin wants to update an expense.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super admin wants to update an expense.	**Step 2:** The system will invoke use case 15.7 (View Expense).
	Step 3: The super admin will select update.	**Step 4:** The system displays the "Update Expense" form to update the following: • Name • Description • Amount

INTEGRATED NGO MANAGEMENT SYSTEM		
	Step 5: The super admin enters the updated details. [ALT]	**Step 6:** The system validates that all required fields are completed. [ALT]
		Step 7: The system requests confirmation for the expense to be updated.
	Step 8: The super admin confirms that the expense can be updated. [ALT]	**Step 9:** The system saves the updated details in the Expense table. [ALT]
		Step 10: The system saves the transaction to the AuditLog table.
		Step 11: The system displays a message informing the super admin that the expense was updated successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects cancel. Use case terminates.	
	[ALT] Step 6: The validation is unsuccessful; the super admin is prompted to enter the details into the empty fields. Go back to step 5.	
	[ALT] Step 8: The super admin selects back. Go back to step 5.	
	[ALT] Step 9: The system fails to update the expense in the relevant entities and displays an error message to the super admin. Use case terminates.	
Conclusion:	The system displays a message informing the super admin that the expense was updated successfully.	
Post-condition:	The expense's details are updated.	
Business rules:	• Financial data such as this is privileged, and thus only persons with appropriate roles may view it.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature. • There should be sufficient budget to log an expense if the value is updated, if this is not the case, the user will be informed, and the update will not be logged.	

INTEGRATED NGO MANAGEMENT SYSTEM	
Assumptions:	• None
Open issues:	• None

Table 135: Update An Expense - Logical Use Case Narrative

2.14.5 View Expense

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	View Expense	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	14.5	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which a super admin wants to view an expanse. The system will display the project budget and allow the super admin to select an expense to view. The use case ends once the expense is displayed for the super admin.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin wants to view an expense.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super admin wants to view the expense.	**Step 2:** The system invokes use case 15.2 (View Project Budget)
	Step 3: The super admin selects an expense	**Step 4:** The system retrieves the selected expense's details form the Expense table.
		Step 5: The system displays the "View Expense" popup displaying the selected expense's details including: • Name • Description

INTEGRATED NGO MANAGEMENT SYSTEM		
		• Amount • Date Logged
	Step 6: The super admin can view an expense.	
Alternate courses:	None	
Conclusion:	The super admin can view an expense.	
Post-condition:	The expense's details are displayed.	
Business rules:	• Financial data such as this is privileged, and thus only persons with appropriate roles may view it.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 136: View Expense - Logical Use Case Narrative

2.14.6 Remove Expense

INTEGRATED NGO MANAGEMENT SYSTEM		
Author:	Christopher Bailey	**Date: 14 March 2026** **Version: 1.0**
Use case name:	Remove Expense	**USE CASE TYPE** Business Requirements: □ System Analysis: ✓ System Design: □
Use case id:	14.6	
Priority:	High	
Source:	EPI-USE	
Primary business actor (pba):	Super Admin	
Primary system actor (psa):	• None	
Other participating actors:	• None	
Other interested stakeholders:	• None	
Description:	This use case describes the event in which the super admin removes an expense. The system will display the projects budget and expenses and allow the super admin to remove an expense. The system will ask for confirmation before the expense is removed. Once confirmed the system will remove the expense. The use case ends once an expense is removed.	
Pre-condition:	Super admin is logged in.	
Trigger:	Super admin wants to remove an expense.	
Typical Course of Events:	**Actor Action**	**System Response**
	Step 1: Super admin wants to remove an expense.	**Step 2:** The system will invoke use case 15.7 (View Expense).
	Step 3: The super admin will select remove.	**Step 4:** The system requests confirmation for the expense to be removed.
	Step 5: The super admin confirms that the expense can be removed. [ALT]	**Step 6:** The system removes the expense from the following tables: • Expense • Project Expense [ALT]

INTEGRATED NGO MANAGEMENT SYSTEM		
		Step 7: The system saves the transaction to the AuditLog table.
		Step 8: The system displays a message informing the super admin that the expense was removed successfully.
Alternate courses:	**[ALT] Step 5:** The super admin selects cancel. Use case terminates.	
	[ALT] Step 6: The system fails to remove expenses in the relevant entities and displays an error message to the super admin. Use case terminates.	
Conclusion:	The system displays a message informing the super admin that the expense was removed successfully.	
Post-condition:	The expense is removed.	
Business rules:	• Financial data such as this is privileged, and thus only persons with appropriate roles may view it.	
Implementation constraints and specifications:	• The user requires an Internet connection to access this feature.	
Assumptions:	• None	
Open issues:	• None	

Table 137: Remove Expense - Logical Use Case Narrative

3. Conclusion

This document outlined the logical narratives for the use cases defined for the proposed system for EPI-USE and groupelephant.com.

4. Sign-off by Client

Sign-Off Confirmation:

I, the undersigned, have reviewed the contents of this document and approved it as complete and accurate. I acknowledge that the information provided meets the requirements agreed upon with team 35, Canvas.

Client Name: Dané Coetzee

Signature: Dané Coetzee

Date: 11/08/2026

5. Sign-off by Team

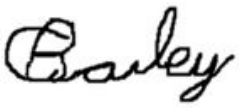
Team member:
Name: Alexander Steven
Date: 16 August 2026
Signature:



Team member:
Name: Amahle Khoza
Date: 16 August 2026
Signature:



Team member:
Name: Christopher Bailey
Date: 16 August 2026
Signature:



Team member:
Name: Nicholas Bailey
Date: 16 August 2026
Signature:



Team member:
Name: Stephan Bredenhann
Date: 16 August 2026
Signature:

