

ITERATION 6

Group 35



NGO PROJECT PORTFOLIO MANAGEMENT SYSTEM

Our Iteration 6 document contains the list of functional requirements and their descriptions, a technical use case diagram, technical ERD, technical use case narratives, technical context diagram, technical primitive level diagrams, screen designs, sequence diagrams, and an updated complexity matrix. Outlining the development of the remaining use cases for the EPI-USE system.

OUR CLIENT:



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	4
Table of Figures	8
1. Document Introduction	10
2. Design and Development of the Use Cases	11
2.1 Introduction	11
2.2 Use Case Outline	11
2.3 Screen Design.....	12
2.3.1 Project Management Subsystem	13
2.3.2 Report Subsystem	15
2.3.3 System Subsystem	19
2.4 Technical Use Case Narrative	21
2.4.1 Project Management Subsystem	21
2.4.2 Report Subsystem	34
2.4.3 System Subsystem	63
2.5 Technical Context Diagram	77
2.5.1 Project Management Subsystem	77
2.5.2 Report Subsystem	78
2.5.3 System Subsystem	79
2.6 Technical Primitive Level Diagram	80
2.6.1 Project Management Subsystem	80
2.6.2 Report Subsystem	84
2.6.3 System Subsystem	90
2.7 Sequence Diagram	93
2.7.1 Project Management Subsystem	93
2.7.2 Report Subsystem	97
2.7.3 System Subsystem	103
2.8 Use Case Diagram	106
2.8.1 Project Management Subsystem	106
2.8.2 Report Subsystem	108
2.8.3 System Subsystem	109
2.9 Technical Entity Relationship Diagram	110
2.10. Updated Functional Requirements Lists	111
2.10.1 Introduction	111
2.10.2 Updated Functional Requirements List.....	111

2.10.3 Updated Functional Requirements Lists Descriptions	114
2.10.4 Functional Requirements List Detailed Changes Description	251
2.11 State Diagrams	257
2.11.1 Inventory State Diagram	257
2.11.2 Deadline State Diagram	257
2.11.3 Animal State Diagram	258
3. Document Conclusion	259
4. Sign-off by Client	260
5. Sign-off by Team	261

Table of Tables

Table 1: Technical Narrative 7.25 Add Photo Album	24
Table 2: Technical Narrative 7.26 View Photo Album	27
Table 3: Technical Narrative 7.27 Update Photo Album	30
Table 4: Technical Narrative 7.28 Remove Photo Album	33
Table 5: Technical Narrative 12.5 View Bike Impact Report.....	36
Table 6: Technical Narrative 12.6 View Budget Report	39
Table 7: Technical Narrative 12.7 View Rhino Report	43
Table 8: Technical Narrative 12.8 View Elephant Report	47
Table 9: Technical Narrative 12.9 View K9 Report.....	57
Table 10: Technical Narrative 12.6 View Machine Learning Dashboard System.....	62
Table 11: Technical Narrative 13.1 Backup System	67
Table 12: Technical Narrative 13.2 Restore System.....	72
Table 13: Technical Narrative 13.3 View Audit Log	76
Table 14: 1.1 User Login – Description.....	114
Table 15: 1.2 User Logout Description	115
Table 16: 1.3 Reset Password – Description.....	116
Table 17: 1.4 Create User – Description.....	117
Table 18: 1.5 Update User – Description.....	118
Table 19: 1.6 View User – Description	119
Table 20: 1.7 Reactivate User - Description	120
Table 21: 1.8 Deactivate User - Description	121
Table 22: 1.9 View FAQ – Description	122
Table 23: 2.1 Create User Profile - Description	123
Table 24: 2.2 Update User Profile - Description.....	124
Table 25: 2.3 View User Profile - Description.....	125
Table 26: 2.4 Deactivate User Profile - Description.....	126
Table 27: 2.5 Add FAQ – Description.....	127
Table 28: 2.6 Update FAQ – Description	128
Table 29: 2.7 Remove FAQ – Description	129
Table 30: 2.8 Add FAQ Type – Description	130
Table 31: 2.9 Update FAQ Type – Description.....	131
Table 32: 2.10 Remove FAQ Type – Description.....	132
Table 33: 2.10 Create Role – Description	133
Table 34: 2.9 View Role – Description.....	134
Table 35: 2.10 Update Role – Description.....	135
Table 36: 2.11 Reactivate Role – Description.....	136
Table 37: 2.12 Deactivate Role – Description.....	137
Table 38: 2.13 Update Role Permission – Description	138
Table 39: 2.14 View Role Permission – Description	139
Table 40: 3.1 Receive Inventory Item – Description.....	140
Table 41: 3.2 View Procurement – Description.....	141
Table 42: 3.2 View Inventory – Description	142
Table 43: 3.3 Write-Off Inventory – Description	143
Table 44: 3.4 Allocate Inventory – Description	144
Table 45: 3.5 Create Inventory Item Type – Description.....	145

Table 46:3.7 View Inventory Item Types – Description	146
Table 47: 3.8 Update Inventory Item Type – Description.....	147
Table 48: 3.9 Deactivate Inventory Item Type - Description	148
Table 49: 3.10 Reactivate Inventory Item Type - Description	149
Table 50: 4.1 Create Group – Description	150
Table 51:4.2 View Group – Description.....	151
Table 52: 4.3 Update Group – Description	152
Table 53: 4.4 Deactivate Group - Description	153
Table 54: 4.5 Create Skill - Description.....	154
Table 55: 4.6 View Skills - Description.....	155
Table 56: 4.7 View Skill - Description	156
Table 57: 4.8 Update Skill - Description	157
Table 58: 4.9 Deactivate Skill - Description	158
Table 59 :5.1 Create Warehouse - Description.....	159
Table 60: 5.2 View Warehouse - Description	160
Table 61: 5.3 Update Warehouse - Description	161
Table 62 5.4: Reactivate Warehouse - Description	162
Table 63: 5.5 Deactivate Warehouse - Description	163
Table 64: Inventory Stock Take - Description	164
Table 65: 6.1 Create Vendor - Description	165
Table 66 6.2: View Vendor - Description.....	166
Table 67: 6.3 View Vendors - Description	167
Table 68: 6.4 Edit Vendor - Description.....	168
Table 69: 6.5 Reactivate Vendor - Description	169
Table 70: 6.6 Deactivate Vendor - Description.....	170
Table 71: 6.7 Create Vendor Type - Description.....	171
Table 72: 6.8 Update Vendor Type - Description	172
Table 73: 6.8 View Vendor Types - Description.....	173
Table 74: 6.10 Reactivate Vendor Type – Description.....	174
Table 75: 6.11 Deactivate Vendor Type - Description	175
Table 76: 7.1 Create Project - Description.....	176
Table 77: 7.2 View Project - Description	177
Table 78: 7.3 Update Project - Description	178
Table 79: 7.4 Deactivate Project - Description	179
Table 80: 7.5 Create Project Type - Description	180
Table 81: 7.6 View Project Type - Description.....	181
Table 82: 7.7 Update Project Type - Description.....	182
Table 83: 7.8 Deactivate Project Type - Description.....	183
Table 84: 7.9 Add Indemnity Form - Description	184
Table 85: 7.10 Update Indemnity Form - Description.....	185
Table 86: 7.11 View Indemnity Form - Description.....	186
Table 87: 7.12 Remove Indemnity Form - Description.....	187
Table 88: 7.13 Capture Indemnity Form - Description	188
Table 89: 7.14 Capture Project Feedback - Description	189
Table 90: 7.15 View Project Feedback - Description	190
Table 91: 7.16 Create Impact Feedback Form - Description	191
Table 92: 7.17 View Impact Forms - Description.....	192

Table 93: 7.18 Update Impact Form - Description	193
Table 94: 7.19 Deactivate Impact Form - Description	194
Table 95: 7.20 Capture Impact Form - Description	195
Table 96: 7.21 View Impact Feedback - Description	196
Table 97: 7.22 Add Photo - Description	197
Table 98: 7.23 View Photo - Description	198
Table 99: 7.24 Delete Photo - Description	199
Table 100: 7.25 Add Photo Album - Description	200
Table 101: 7.26 View Photo Album - Description	201
Table 102: 7.27 Update Photo Album - - Description	202
Table 103: 2.28 Remove Photo Album – Description.....	203
Table 104: 8.1 Add Dog - Description.....	204
Table 105: 8.2 View Dog - Description	205
Table 106: 8.3 View All Dogs- Description	206
Table 107: 8.4 Update Dog - Description	207
Table 108: 8.5 Deactivate Dog - Description	208
Table 109: 8.6 Assign User to Dog – Description	209
Table 110: 8.7 View User-Dog Assignment – Description	210
Table 111: 8.8 Update User-Dog Assignment - Description	211
Table 112: 8.9 Reactivate User-Dog Assignment - Description	212
Table 113: 8.10 Deactivate User-Dog Assignment - Description.....	213
Table 114: 8.11 Add K9 Role - Description	214
Table 115: 8.12 Update K9 Role - Description.....	215
Table 116: 8.13 View K9 Role - Description.....	216
Table 117: 8.14 Reactivate K9 Role - Description.....	217
Table 118: 8.15 Deactivate K9 Role – Description.....	218
Table 119: 8.16 View K9 Ancestry – Description.....	219
Table 120: 9.1 Add Rhino - Description.....	220
Table 121: 9.2 View Rhino - Description	221
Table 122: 9.3 View Rhinos - Description.....	222
Table 123: 9.4 Update Rhino - Description	223
Table 124: 9.5 9.5 Deactivate Rhino - Description	224
Table 125: 10.1 Add Elephant - Description.....	225
Table 126: 10.2 View Elephant - Description	226
Table 127: 10.3 View Elephants - Description.....	227
Table 128: 10.4 Update Elephant – Description.....	228
Table 129: 10.5 Deactivate Elephant - Description	229
Table 130: 11.1 Create Message – Description	230
Table 131: 11.2 View Message – Description.....	231
Table 132: 12.1 Generate Custom Report - Description	232
Table 133: 12.2 View Custom Report - Description	233
Table 134: 12.3 Update Custom Report - Description	234
Table 135: 12.4 Remove Custom Report - Description	235
Table 136: 12.5 View Bike Impact Report - Description.....	236
Table 137: 12.6 View Budget Report – Description	237
Table 138: 12.7 View Rhino Report - Description	238
Table 139: 12.10 View Elephant Report - Description	239

Table 140: 12.9 View K9 Report – Description	240
Table 141: 12.10 View Machine Learning Dashboard - Description	241
Table 142: 13.1 Backup System - Description	242
Table 143: 13.2 Restore System - Description.....	243
Table 144: 13.3 View Audit Log - Description	244
Table 145: 14.1 View Project Budget – Description	245
Table 146: 14.2 View Project Budgets – Description	246
Table 147: 14.3 Log An Expense – Description.....	247
Table 148: 14.4 Update An Expense – Description	248
Table 149: 14.5 View An Expense – Description	249
Table 150: 16.6 Remove an Expense – Description	250
Table 151: User Subsystem Modifications	251
Table 152: Admin Subsystem Modifications	252
Table 153: Inventory Subsystem Modifications	252
Table 154: Group Subsystem Modifications.....	252
Table 155: Warehouse Subsystem Modifications	253
Table 156: Vendor Subsystem Modifications	253
Table 157: Vendor Subsystem Modifications	254
Table 158: K9 Subsystem Modifications.....	255
Table 159: Rhino Subsystem Modifications	255
Table 160: Elephant Subsystem Modifications.....	255
Table 161: Message Subsystem Modifications.....	255
Table 162: Report Subsystem Modifications.....	256
Table 163: System Subsystem Modifications	256
Table 164: Budget Subsystem Modifications	256

Table of Figures

Figure 1: Use Case 7.25 Add Photo Album Screen Design	13
Figure 2: Use Case 7.26 View Photo Album Screen Design	13
Figure 3: Use Case 7.27 Update Photo Album Screen Design.....	14
Figure 4: Use Case 7.28 Remove Photo Album Screen Design.....	14
Figure 5: Use Case 12.5 View Bike Impact Report Screen Design	15
Figure 6: Use Case 12.6 View Budget Report Screen Design	15
Figure 7: Use Case 12.7 View Rhino Report Screen Design	16
Figure 8: Use Case 12.8 View Elephant Report Screen Design	16
Figure 9: Use Case 12.9 View K9 Report Screen Design	17
Figure 10: Use Case 12.10 View Machine Learning Dashboard Screen Design	18
Figure 11: Use Case 13.1 Backup Subsystem Screen Design	19
Figure 12: Use Case 13.1 Restore Subsystem Screen Design.....	19
Figure 13: Use Case 13.3 View Audit Log Screen Design.....	20
Figure 14: Project Management Technical Context Diagram.....	77
Figure 15: Report Subsystem Technical Context Diagram	78
Figure 16: System Subsystem Context Diagram	79
Figure 17: Use Case 7.25 Add Photo Album Technical Primitive Diagram	80
Figure 18: Use Case 7.26 View Photo Album Technical Primitive Diagram	81
Figure 19: Use Case 7.27 Update Photo Album Technical Primitive Diagram	82
Figure 20: Use Case 7.28 Remove Photo Album Technical Primitive Diagram	83
Figure 21: Use Case 12.5 View Bike Impact Report Technical Primitive Diagram	84
Figure 22: Use Case 12.6 View Budget Report Technical Primitive Diagram.....	85
Figure 23: Use Case 12.7 View Rhino Report Technical Primitive Diagram.....	86
Figure 24: Use Case 12.8 View Elephant Report - Technical Primitive Diagram	87
Figure 25: Use Case 12.9 View K9 Report Technical Primitive Diagram.....	88
Figure 26: Use Case 12.10 View Rhino Report Technical Primitive Diagram	89
Figure 27: Use Case 13.1 Backup System Technical Primitive Diagram.....	90
Figure 28: Use Case 13.2 Restore System Technical Primitive Diagram	91
Figure 29: Use Case 13.3 View Audit Log Technical Primitive Diagram	92
Figure 30: Use Case 7.25 Add Photo Album Sequence Diagram	93
Figure 31: Use Case 7.26 View Photo Album Sequence Diagram.....	94
Figure 32: Use Case 7.27 Update Photo Album Sequence Diagram	95
Figure 33: Use Case 7.28 Remove Photo Album Sequence Diagram	96
Figure 34: Use Case 12.5 View Bike Impact Report Sequence Diagram	97
Figure 35: Use Case 12.6 View Budget Report Sequence Diagram	98
Figure 36: Use Case 12.7 View Rhino Report Sequence Diagram	99
Figure 37: Use Case 12.8 View Elephant Report Sequence Diagram	100
Figure 38: Use Case 12.9 View K9 Report Sequence Diagram	101
Figure 39: Use Case 12.10 View Machine Learning Dashboard Sequence Diagram	102
Figure 40: Use Case 13.1 Backup System Sequence Diagram.....	103
Figure 41: Use Case 13.2 Restore System Sequence Diagram	104
Figure 42: Use Case 13.3 View Audit Log Sequence Diagram	105
Figure 43: Project Management Subsystem Use Case Diagram.....	106
Figure 44: Project Management Subsystem Use Case Diagram (Continued)	107
Figure 45: Report Subsystem Use Case Diagram	108

Figure 46: System Subsystem Use Case Diagram.....	109
Figure 47: Technical Entity Relationship Diagram	110
Figure 48: Inventory State Diagram.....	257
Figure 49: Deadline State Diagram	257
Figure 50: Animal State Diagram	258

1. Document Introduction

This document presents the sixth iteration of the Integrated ERP Project Portfolio Management System for EPI-USE and groupelephant.com.

Iteration 6 contains an updated list of requirements together with the summary description of the requirements, a use case diagram for the use cases which will be designed during Iteration 6, technical ERD for the data elements involved in Iteration 6. For the assigned use cases, this document has included the screen designs for the use cases, along with the technical use case narratives, technical primitive diagrams, technical context diagrams, the sequence diagrams, as well as state diagrams.

2. Design and Development of the Use Cases

2.1 Introduction

This section of the document outlines the design and development of 13 use cases assigned to Iteration 6. It includes screen designs that follow efficient, intuitive, and visually appealing user interface aimed at optimizing user experience.

This section also includes technical use case narratives, technical context diagrams, technical primitive diagrams, sequence diagrams and state diagrams which help better understand how each use case works and help visualize the system's flow and interactions.

The present portion of the document also includes the requirements list, technical use case diagram, and the technical ERD which have been refined using client feedback from previous iterations.

2.2 Use Case Outline

As the system consists of 137 distinct use cases, at least 5% of the use cases need to be completed.

$$137 \times 5\% = 6.85$$

We have decided to complete 13 use cases and based on the outcomes of the dependency matrices; we decided to complete the following use cases:

12.10 View Machine Learning Dashboard

Project Management Subsystem (Subsystem 7):

7.25 Add Photo Album

7.26 View Photo Album

7.27 Update Photo Album

7.28 Delete Photo Album

System Subsystem (Subsystem 13):

13.1 Backup System

13.2 Restore System

13.3 View Audit Log

Report Subsystem (Subsystem 12):

12.5 View Bike Impact Report

12.6 View Budget Report

12.7 View Rhino Report

12.8 View Elephant Report

12.9 View K9 Report

2.3 Screen Design

Each screen design has been created focusing on creating intuitive, efficient and visually appealing user interfaces. Attached figures of the screen design will be parted with its use case.

2.3.1 Project Management Subsystem

2.3.1.1 Add Photo Album

The screenshot shows the 'Photo Albums' interface with a 'Create Album' modal open. The modal contains the following fields and controls:

- Album Name:** A text input field containing 'Bike Drop'.
- Description:** A text area containing 'View Bike Drop Images in this album.'
- Link to Project:** A dropdown menu with 'Bike drop 2026' selected.
- Cover Photo:** A section showing a thumbnail of a file named 'donedone.png' with a 'Choose File' button and a 'Remove' link.
- Buttons:** 'Cancel' and 'Create' buttons at the bottom right of the modal.

The background interface includes a navigation bar with links like Home, Teams, Projects, Calendar, Messages, Animals, Inventory, and Admin. A search bar is also present in the top right.

Figure 1: Use Case 7.25 Add Photo Album Screen Design

2.3.1.2 View Photo Album

The screenshot displays the 'View Photo Album' screen for an album titled 'Bike Drop'. The interface includes:

- Navigation:** A 'Back to Albums' link at the top left.
- Album Title:** 'Bike Drop' in a large, bold font.
- Description:** 'View Bike Drop images' and a sub-label 'Bike drop 2026'.
- Actions:** 'Edit Album', 'Delete Album', and 'Upload Photos' buttons in the top right.
- Cover Image:** A thumbnail image labeled 'Cover' showing a document with a line graph.

The background features the same navigation bar and search bar as seen in Figure 1.

Figure 2: Use Case 7.26 View Photo Album Screen Design

2.3.1.3 Update Photo Album

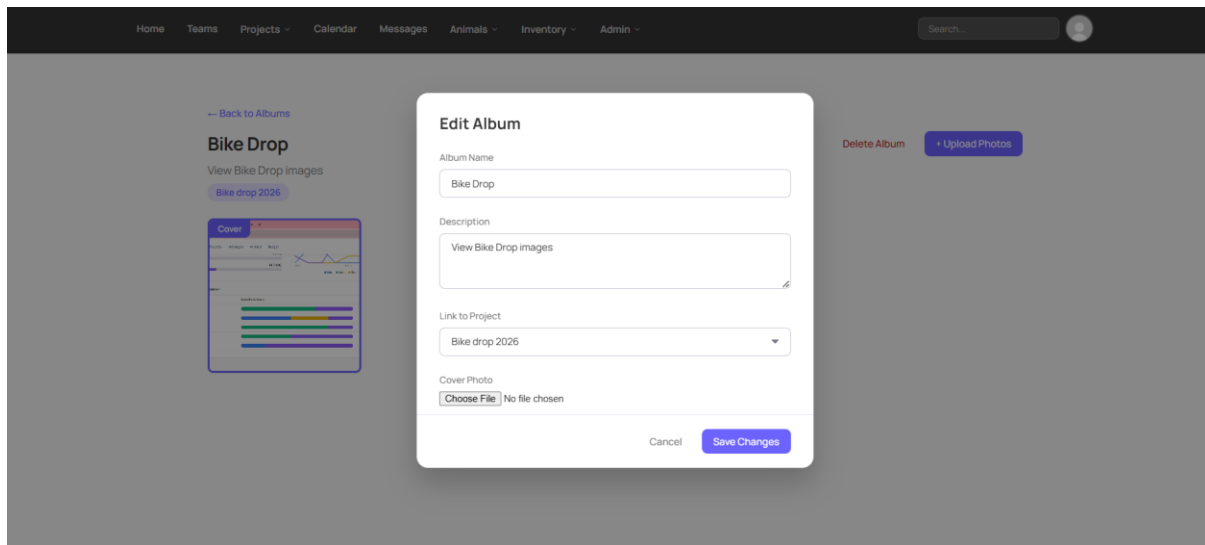


Figure 3: Use Case 7.27 Update Photo Album Screen Design

2.3.1.4 Remove Photo Album

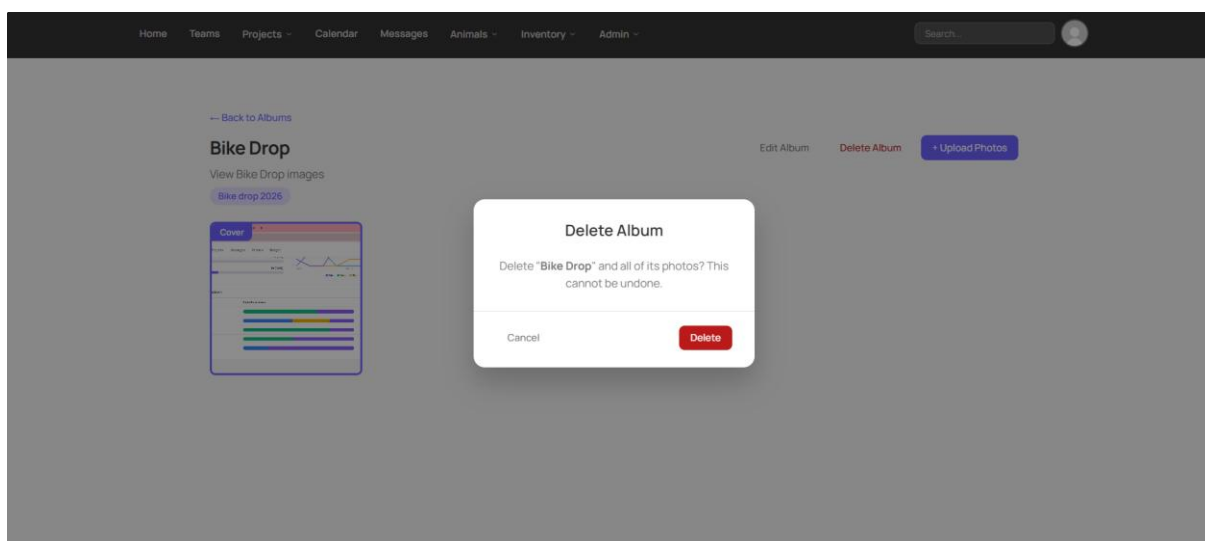


Figure 4: Use Case 7.28 Remove Photo Album Screen Design

2.3.2 Report Subsystem

2.3.2.1 View Bike Impact Report

Bike Impact Report

Programme reach, bicycle deployment and attendance outcomes for bike projects

Export CSV Download PDF

Reporting period

This month This quarter **Year to date** Last 12 months

From To

01/01/2026 27/07/2026 Generate report

Period 01 Jan 2026 - 27 Jul 2026 (208 days) Generated 26 Jul 2026 23:40 UTC by Stephan Bredenmann

BIKE PROJECTS IN PERIOD 3 running at some point in the window	BENEFICIARIES ENROLLED 102 93 still active on the register	BICYCLES DEPLOYED 92 6 written off	ATTENDANCE RATE 78.2% 2 073 of 2 652 records	ARRIVED BY BICYCLE 76.8% share of attended days recorded as arriving with a bike
BENEFICIARY RETENTION 91.2% 9 removed from the register	IMPACT FORM SUBMISSIONS 21 across 3 registered forms	COST PER BENEFICIARY R 7 106.71 R 724 884.00 spent in the window		

Figure 5: Use Case 12.5 View Bike Impact Report Screen Design

2.3.2.2 View Budget Report

Budget Report

Approved budget against actual and committed spend

Export CSV Download PDF

Reporting period

This month This quarter **Year to date** Last 12 months

From To

01/01/2026 27/07/2026 Generate report

Period 01 Jan 2026 - 27 Jul 2026 (208 days) Generated 26 Jul 2026 23:41 UTC by Stephan Bredenmann

APPROVED BUDGET R 5 755 000 across 7 of 8 projects in scope	SPEND IN PERIOD R 1 676 937 102 expense(s) dated inside the window	LIFE-TO-DATE SPEND R 3 662 018 all expenses ever logged against these projects	COMMITTED, UNSETTLED R 240 360 4 open order(s)	TOTAL EXPOSURE R 3 902 378 life-to-date spend plus unsettled orders
BUDGET CONSUMED 67.8% exposure against approved budget	PROJECTS OVER BUDGET 1 R 222 583.50 over in total	AVERAGE EXPENSE R 16 440.55 mean value of expenses logged in the period		

Figure 6: Use Case 12.6 View Budget Report Screen Design

2.3.2.3 View Rhino Report

Rhino Report

Population register, dehorning activity, sightings and relocations

Export CSV Download PDF

Reporting period

This month This quarter **Year to date** Last 12 months

From 01/01/2026 To 27/07/2026 Generate report

Period 01 Jan 2026 - 27 Jul 2026 (208 days) Generated 26 Jul 2026 23:42 UTC by Stephan Bredenhann

RHINOS ON REGISTER 16 2 retired record(s) excluded	SIGHTED IN PERIOD 12 75.0% of the active register	NOT SIGHTED IN PERIOD 4 2 have never been sighted	DEHORNING IN PERIOD 2 animals dehorned inside the window	CURRENTLY WITHOUT HORN 43.8% 7 of 16 on the register
RELOCATION PROJECTS 1 capture date or project window falls in the period	RHINOS IN RELOCATIONS 3 distinct animals attached to those projects	NEW RECORDS IN PERIOD 2 date of birth falls inside the window	AVERAGE WEIGHT 1778 kg from 14 record(s) with a weight captured	

Figure 7: Use Case 12.7 View Rhino Report Screen Design

2.3.2.4 View Elephant Report

Elephant Report

Population register, tusk management, sightings and relocations

Export CSV Download PDF

Reporting period

This month This quarter **Year to date** Last 12 months

From 01/01/2026 To 27/07/2026 Generate report

Period 01 Jan 2026 - 27 Jul 2026 (208 days) Generated 26 Jul 2026 23:43 UTC by Stephan Bredenhann

ELEPHANTS ON REGISTER 13 1 retired record(s) excluded	SIGHTED IN PERIOD 10 76.9% of the active register	NOT SIGHTED IN PERIOD 3 2 have never been sighted	DETUSKING IN PERIOD 1 animals detusked inside the window	CURRENTLY WITHOUT TUSKS 15.4% 2 of 13 on the register
RELOCATION PROJECTS 1 capture date or project window falls in the period	ELEPHANTS IN RELOCATIONS 3 distinct animals attached to those projects	NEW RECORDS IN PERIOD 1 date of birth falls inside the window	AVERAGE WEIGHT 4 242 kg from 12 record(s) with a weight captured	

Figure 8: Use Case 12.8 View Elephant Report Screen Design

2.3.2.5 View K9 Report

K9 Operational Report

K9 register, handler assignments and operational activity

[Export CSV](#)
[Download PDF](#)

Reporting period

This month
This quarter
Year to date
Last 12 months

From
To

2026/01/01
2026/08/02

[Generate report](#)

Period 01 Jan 2026 - 02 Aug 2026 (214 days)
Generated 02 Aug 2026 20:37 UTC by Chris Bailey

DOGS ON REGISTER
3

RETIRED DOGS
1

ACTIVE HANDLERS
2

ACTIVE ASSIGNMENTS
3

Active K9 Register

The operational K9 register as it stands today. The reporting period filters assignment activity rather than the register itself.

Name	Breed / Species	Gender	Age	Current Handler	Operational Roles
------	-----------------	--------	-----	-----------------	-------------------

Figure 9: Use Case 12.9 View K9 Report Screen Design

2.3.2.6 View Machine Learning Dashboard

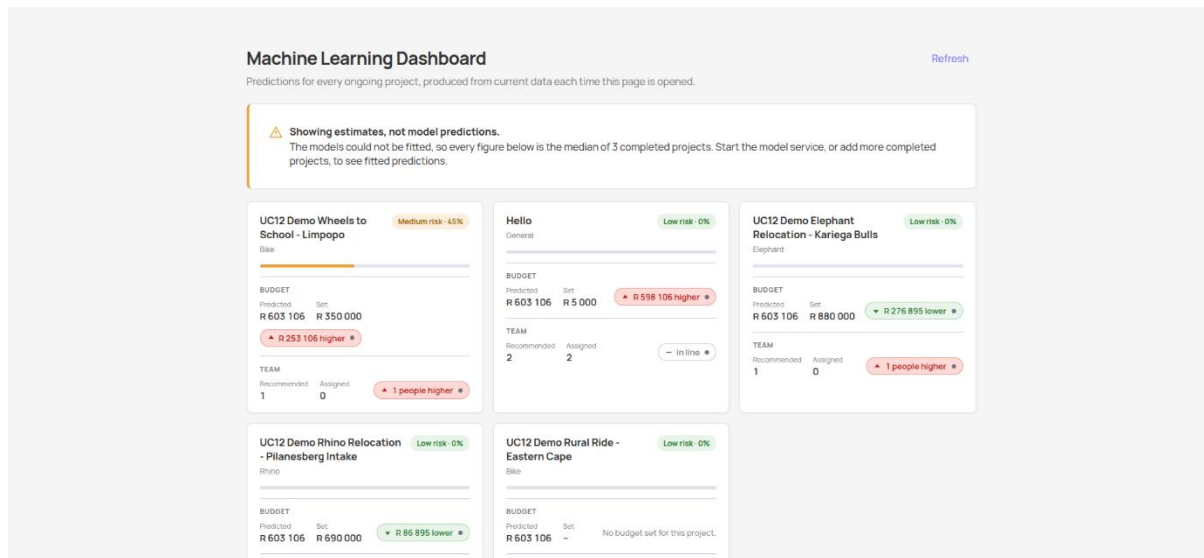


Figure 10: Use Case 12.10 View Machine Learning Dashboard Screen Design

2.3.3 System Subsystem

2.3.3.1 Backup System

Database Backups Create Backup

Create, schedule, download, and restore full database backups.

Automatic backups

When enabled, the system creates a full database backup at the chosen local day and time. Up to 5 automatic backups are kept; manual backups are never auto-deleted.

☒ Enabled
 Day: Sunday
Time: 03:00
Save Schedule

No automatic backup has run yet.

Date	File	Size	Type	Created by	Actions
Jul 26, 2026, 2:55:20 PM	EPI_USE_ERP_DB_20260726_145520_651a9cfb68b244bd8eefb6e594ffc5b5.bak	14.1 MB	Manual	Stephan Bredenmann	Download Restore Delete
Jul 26, 2026, 2:54:52 PM	EPI_USE_ERP_DB_20260726_145452_dedba49cf619414a854780cae1721d97.bak	14.1 MB	Manual	Stephan Bredenmann	Download Restore Delete

Figure 11: Use Case 13.1 Backup Subsystem Screen Design

2.3.3.2 Restore System

Database Backups Create Backup

Create, schedule, download, and restore full database backups.

Automatic backups

When enabled, the system creates a full database backup at the chosen local day and time. Up to 5 automatic backups are kept; manual backups are never auto-deleted.

☒ Enabled
 Day: Sunday
Time: 03:00
Save Schedule

No automatic backup has run yet.

Confirm database restore

This will replace the current database with the selected backup. All users will be signed out while the restore runs. Continue?

EPI_USE_ERP_DB_20260726_145520_651a9cfb68b244bd8eefb6e594ffc5b5.bak — Jul 26, 2026, 2:55:20 PM

Cancel Restore

Date	File	Size	Type	Created by	Actions
Jul 26, 2026, 2:55:20 PM	EPI_USE_ERP_DB_20260726_145520_651a9cfb68b244bd8eefb6e594ffc5b5.bak	14.1 MB	Manual	Stephan Bredenmann	Download Restore Delete
Jul 26, 2026, 2:54:52 PM	EPI_USE_ERP_DB_20260726_145452_dedba49cf619414a854780cae1721d97.bak	14.1 MB	Manual	Stephan Bredenmann	Download Restore Delete

Figure 12: Use Case 13.1 Restore Subsystem Screen Design

2.3.3.3 View Audit Log

[Home](#)
[Teams](#)
[Projects](#)
[Calendar](#)
[Messages](#)
[Animals](#)
[Inventory](#)
[Admin](#)



Audit Logs

View all recorded system actions performed by users.

Total Logs

22

Last 24 Hours

16

Last 7 Days

22

2 Log(s)

Date	User	Action	Entity	Details
01/08/2026 18:49	Nick Bailey	Update	Project	Updated project 'K9 Walk'
01/08/2026 18:49	Nick Bailey	Create	Project	Created project 'K9 Walk'

[Previous](#)
[Page 1](#)
[Next](#)

Integrated NGO Management System

[Help](#)
[All FAQs](#)
[Feedback](#)

© 2026 EPI-USE

Figure 13: Use Case 13.3 View Audit Log Screen Design

2.4 Technical Use Case Narrative

2.4.1 Project Management Subsystem

2.4.1.1 Add Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Amahle Khoza	Date: 28 July 2026	
		Version: 1.0	
USE CASE NAME:	Add Photo Album	USE CASE TYPE	
USE CASE ID:	7.25	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☐	
SOURCE:	EPI-USE	System Design: ☒	
Primary Business Actor (PBA):	Administrator		
Primary System Actor (PSA):	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	Project Team members		
Description:	This use case describes how an authorized user creates a new photo album container within a project. The user enters album details in a modal form, and upon frontend validation and backend processing, the system creates the record, initializes the media storage path, logs the audit entry, and confirms creation.		
Pre-condition:	The user must be logged into the system and have access to the target active project.		
Trigger:	The user selects the "Add Album" option inside a project's media section.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: The user navigates to the Project Media gallery tab.		Step 2: The system invokes UC 7.26 (View Photo Album) to load existing photo albums associated with the

			project via an HTTP GET request (PhotoAlbumController – GetAlbumsByProjectId).
	Step 3: The user selects the "Add Album" button.		Step 4: The system displays the Add Photo Album modal containing: • Modal title: "Add Photo Album" • Album Name field (Required, text input) • Album Description field (Optional, textarea input) • Action buttons: Create, Cancel. The Create button remains disabled until mandatory fields are filled.
	Step 5: The user enters the album details and selects the "Create" button. [ALT]		Step 6: The Angular frontend validates that the Album Name is non-empty and compliant with naming policies. [ALT]
			Step 7: The system routes the request via the Angular PhotoAlbumService to the Web API endpoint (PhotoAlbumController – CreatePhotoAlbum). The backend validates uniqueness within the project and persists the record to SQL Server

			using Entity Framework Core. PhotoAlbum Entity: • PhotoAlbumId (PK) • ProjectId (FK) • Name • Description • CreatedByUserId (FK) • CreatedAt
			Step 8: The system writes a log to the AuditLog table capturing the operation user ID, timestamp, and details ("Created Photo Album [Name] for Project [ID]").
			Step 9: The system displays a toaster notification: <i>"Photo album created successfully."</i>
Alternate courses:	[ALT] Step 5: The user selects "Cancel" or closes the modal. Terminate use case.		
	[ALT] Step 6: Validation fails (e.g., duplicate album name within project). An inline error message is displayed: <i>"An album with this name already exists in this project."</i> Go back to Step 4.		
Conclusion:	A new photo album record is created, linked to the project, and ready to receive image uploads.		
Post-condition:	The new album appears in the project's media view.		
Business Rules:	• Only project contributors or admins can create albums. • Album names must be unique per project.		

Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None.

Table 1: Technical Narrative 7.25 Add Photo Album

2.4.1.2 View Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Amahle Khoza	Date: 28 July 2026	
		Version: 1.0	
USE CASE NAME:	View Photo Album	USE CASE TYPE	
USE CASE ID:	7.26	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☐	
SOURCE:	EPI-USE	System Design: ☒	
Primary Business Actor (PBA):	Administrator		
Primary System Actor (PSA):	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case describes how a user views a project's photo album list or opens a specific album to view contained thumbnail images and metadata.		
Pre-condition:	The user must be logged into the system and have read access to the project.		
Trigger:	The user selects a project's media/photo section or clicks a specific photo album card.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: The user selects a project photo album from the gallery grid.		Step 2: The system sends an HTTP GET request via the Angular HTTP Client to PhotoAlbumController – GetAlbumDetails with PhotoAlbumId.
			Step 3: The backend verifies user JWT authorization claims and retrieves album metadata

			along with child photo records (.AsNoTracking()) from SQL Server via EF Core. Entities Involved: PhotoAlbum: PhotoAlbumId, Name, Description, CreatedAt Photo: PhotoId, PhotoAlbumId (FK), FilePath, UploadedByUserId (FK), UploadedAt [ALT]
			Step 4: The system renders the album viewer page displaying album metadata, cover details, and a grid of photo thumbnails with light-box expansion capabilities.
Alternate courses:	[ALT] Step 3: The requested album ID is invalid or deleted. System displays a notification: "Photo album not found." Terminate use case.		
Conclusion:	The album contents and associated image media are retrieved and rendered on screen.		
Post-condition:	The user views album contents without changing underlying data.		
Business Rules:	Secure Access (User must have read permissions for the project).		
Implementation constraints and specifications:	Lazy loading or thumbnail optimization for fast image gallery load times.		
Assumptions:	None		

Open issues:

- None.

Table 2: Technical Narrative 7.26 View Photo Album

2.4.1.3 Update Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Amahle Khoza	Date: 28 July 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):	Administrator		
Primary System Actor (PSA):	• None		
Other Participating Actors:	• None		
Other Interested Stakeholders:	None		
Description:	This use case describes how a user modifies existing photo album details (e.g., album title or description). The changes are validated, updated in the database, and logged in audit history.		
Pre-condition:	The user is logged in, has access to the project.		
Trigger:	The user clicks the "Edit Album" button on an existing photo album card/header.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: The user clicks the "Edit Album" action button on an album card.		Step 2: The system displays the Edit Photo Album modal pre-populated with existing data (Name and Description fields).
	Step 3: The user modifies the fields and clicks the "Save Changes" button.		Step 4: The system validates that the mandatory Name field is non-empty.

	[ALT]		[ALT]
			Step 5: The system sends an HTTP PUT request (PhotoAlbumController – UpdatePhotoAlbum). The backend validates authorization rules, verifies that the user is the creator or Super Admin, and updates the database record via EF Core.
			Step 6: The system creates an entry in the AuditLog table ("Updated Photo Album [ID] details").
			Step 7: The system closes the modal and displays a toaster notification: "Photo album updated successfully."
Alternate courses:	[ALT] Step 3: The user clicks "Cancel". Changes are discarded. Terminate use case.		
	[ALT] Step 4: Validation fails (e.g., blank title). Error message displayed: "Album name cannot be empty." Go back to Step 2.		
Conclusion:	Photo album details are updated in SQL Server and reflected across the UI.		
Post-condition:	The photo album reflects updated metadata.		
Business Rules:	Admin can update it.		
Implementation constraints and specifications:	None		
Assumptions:	None		

Open issues:

- None.

Table 3: Technical Narrative 7.27 Update Photo Album

2.4.1.4 Remove Photo Album

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Amahle Khoza	Date: 28 July 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):	Administrator		
Primary System Actor (PSA):	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case describes how an authorized user deletes a photo album along with its associated photo file references. The user confirms deletion via a modal, and the system purges the database records and logs the deletion action.		
Pre-condition:	The user is logged in, has access to the project.		
Trigger:	The user clicks the "Delete Album" button.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: The user selects the "Delete Album" option on a photo album.		Step 2: The system displays a confirmation modal displaying: - Title: "Confirm Delete Album" - Message: "Are you sure you want to delete '[Album Name]'? All photos in this album will

			be permanently removed."
	Step 3: The user selects the "Confirm" button. [ALT]		Action buttons: Confirm, Cancel.
			Step 4: The request is routed via Angular PhotoAlbumService via an HTTP DELETE call to PhotoAlbumController – DeletePhotoAlbum.
			Step 5: The API verifies permissions, removes the album record and child photo entries from SQL Server via EF Core cascading delete, and triggers cloud/file storage cleanup for physical image files.
			Step 6: The system records an entry in AuditLog ("Deleted Photo Album [ID] and associated media").
			Step 7: The system closes the modal and displays a toaster notification: "Photo album updated successfully."
Alternate courses:	[ALT] Step 3: The user clicks "Cancel". Changes are discarded. Terminate use case.		
Conclusion:	The photo album and its linked photos are deleted from the system.		
Post-condition:	The photo album and its images are permanently removed.		
Business Rules:	- Admin can delete a photo album. - Confirmation modal is mandatory to prevent accidental data loss.		

Implementation constraints and specifications:	• Cascading deletion logic must clean up database records and associated file storage assets cleanly.
Assumptions:	• None
Open issues:	• None.

Table 4: Technical Narrative 7.28 Remove Photo Album

2.4.2 Report Subsystem

2.4.2.1 View Bike Impact Report

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Alexander Steven	Date: 24 July 2026 Version: 1.0	
USE CASE NAME:	View Bike Impact Report	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	12.5		
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 a user viewing the Bike Impact Report from the Admin section of the system. The user opens the report page, where the system loads a year-to-date view showing bicycle deployment, beneficiary attendance and cost figures for all bike projects. The user can narrow or widen the period using preset ranges or specific dates and can download the report as a PDF or CSV file. Every view or export is written to the audit log.		
Pre-condition:	The user is logged in and holds the Reports.BikeImpact.View permission.		
Trigger:	The user selects Bike Impact Report from the Admin menu.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: User selects Bike Impact Report from the Admin menu.		Step 2: The system checks that the user is logged in and holds the required permission, then loads the report page with the

			Year-to-date period applied by default.
			Step 3: The system requests the report from the API, which reads the attendance, inventory and expense records for the period and returns the finished report.
			Step 4: The page displays the key figures at the top, followed by the report's sections covering project summaries, attendance, bicycle deployment and cost.
	Step 5: User changes the reporting period, either by choosing a different preset or entering two dates, and selects Generate report. [ALT]		Step 6: The system repeats the same checks and rebuilds the report for the new period.
	Step 7: User selects Download PDF.		Step 8: The system rebuilds the report, produces a PDF on the server, and the browser saves the file. The system records the export in the audit log. AuditLogs Entity: • UserId (FK) • Action • EntityType • EntityId • Timestamp
Alternate courses:	[ALT] Step 1: User is not logged in. The system redirects to the login page; a direct request to the API returns an unauthorised response.		

	[ALT] Step 1: User is logged in but does not hold the Reports.BikeImpact.View permission. The system redirects to a not-authorized page; a direct request returns a forbidden response.
	[ALT] Step 5: User enters a start date later than the end date. The Generate report button is disabled and an inline message explains why.
	[ALT] Step 5: User selects a period longer than five years. The system rejects the request and asks for a shorter range.
	[ALT] Step 3: No bike project falls inside the selected period. The report still returns successfully, showing a single message that the period was empty rather than leaving any section blank.
	[ALT] Step 3: A section has no rows for the period even though the rest of the report does. That section shows its own short message instead of an empty table.
Conclusion:	The use case concludes when the user has viewed the report on screen or has downloaded it as a PDF or CSV file.
Post-condition:	The report has been displayed to the user and, if requested, saved as a file. No attendance, inventory or expense record has been changed; the only change is the new entry in the audit log.
Business Rules:	• A bike project is included in the report if any part of its timeline falls inside the selected period, not only if it started there. • Attendance records are matched to present, present with bicycle, or absent; anything that cannot be matched is counted separately rather than guessed at.
Implementation constraints and specifications:	• The report page is served from /reports/bike-impact and is protected by a route guard and a permission guard on the Angular side.
Assumptions:	• The user has been assigned the correct role and permissions by an administrator before requesting this report.
Open issues:	• None.

Table 5: Technical Narrative 12.5 View Bike Impact Report

2.4.2.2 View Budget Report

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Alex Steven	Date: 24 July 2026	
		Version: 1.0	
USE CASE NAME:	View Budget Report	USE CASE TYPE	
USE CASE ID:	12.6	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☐	
SOURCE:	EPI-USE	System Design: ☒	
Primary Business Actor (PBA):	User		
Primary System Actor (PSA):	None		
Other Participating Actors:	None		
Other Interested Stakeholders:	None		
Description:	This use case describes a user viewing the Budget Report from the Admin section of the system. The user opens the report page, where the system loads a year-to-date view comparing each project's approved budget against actual and committed spend. The user can narrow or widen the period using preset ranges or specific dates and can download the report as a PDF or CSV file. Every view or export is written to the audit log.		
Pre-condition:	The user is logged in and holds the Reports.Budget.View permission.		
Trigger:	The user selects Budget Report from the Admin menu.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: User selects Budget Report from the Admin menu.		Step 2: The system checks that the user is logged in and holds the required permission, then loads the report page with the Year-to-date period applied by default.

			Step 3: The system requests the report from the API, which reads the expenses, purchase orders, projects and inventory for the period, works out how much of each budget has been committed or spent, and returns the finished report.
			Step 4: The page displays the key figures at the top, followed by the report's sections covering budget against actual spend, spend by category, spend by supplier and any budget exceptions.
	Step 5: User changes the reporting period and selects Generate report. [ALT]		Step 6: The system repeats the same checks and rebuilds the report for the new period.
	Step 7: User selects Download PDF.		Step 8: The system rebuilds the report, produces a PDF on the server, and the browser saves the file. The system records the export in the audit log. AuditLogs Entity: • UserId (FK) • Action • EntityType • EntityId • Timestamp
Alternate courses:	[ALT] Step 1: User is not logged in. The system redirects to the login page; a direct request to the API returns an unauthorised response.		

	[ALT] Step 1: User is logged in but does not hold the Reports.Budget.View permission. The system redirects to a not-authorized page; a direct request returns a forbidden response.
	[ALT] Step 5: User enters a start date later than the end date. The Generate report button is disabled and an inline message explains why.
	[ALT] Step 5: User selects a period longer than five years. The system rejects the request and asks for a shorter range.
	[ALT] Step 3: A project has no approved budget recorded. It is shown with a dash instead of a budget figure and listed separately, rather than being treated as a budget of zero.
	[ALT] Step 3: A project's spend and commitments exceed its budget. It is marked as over budget and listed in the exceptions section.
Conclusion:	The use case concludes when the user has viewed the report on screen or has downloaded it as a PDF or CSV file.
Post-condition:	The report has been displayed to the user and, if requested, saved as a file. No financial record has been changed; the only change is the new entry in the audit log.
Business Rules:	A purchase order counts as committed spend only until an expense settles it, so nothing is counted twice.
Implementation constraints and specifications:	The report page is served from /reports/budget and is protected by a route guard and a permission guard on the Angular side.
Assumptions:	The user has been assigned the correct role and permissions by an administrator before requesting this report.
Open issues:	None.

Table 6: Technical Narrative 12.6 View Budget Report

2.4.2.3 View Rhino Report

INTEGRATED NGO MANAGEMENT SYSTEM			
Author:	Amahle Khoza		Date: 24/07/2026 Version: 1.0
Use case name:	View Rhino Report		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
Use case id:	12.7		
Priority:	High		
Source:	EPI-USE		
Primary business actor (PBA):	Administrator		
Primary system actor (PSA):	• None		
Other participating actors:	• None		
Other interested stakeholders:	• None		
Description:	This use case describes how an authorized user views a pre-structured Rhino Report within a selected date range. The user specifies the filter criteria, and the system fetches, calculates, and presents aggregated rhino operational, tracking, and health metrics.		
Pre-condition:	The user must be logged into the system and possess appropriate permissions.		
Trigger:	The user wants to generate and view a Rhino Report for a specific date range.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: The user navigates to the Reports module and selects the "Rhino Report" option.		Step 2: The system loads and displays the Rhino Report filter page, containing date range selection inputs (Start Date and End Date) and a "Generate Report" button. The Generate Report button remains disabled until valid date values are selected.
	Step 3: The user selects the Start Date and End Date parameters and clicks		Step 4: The system validates the selected date parameters using Angular frontend validation before

	the "Generate Report" button. [ALT]		submitting the query. The system checks that: • The Start Date field is not null or empty. • The End Date field is not null or empty. • The Start Date is less than or equal to the End Date. [ALT]
			Step 5: The system routes the query through the Angular frontend and Report Service, which dispatches an HTTP GET request via the Angular HTTP Client to the corresponding Web API endpoint in Visual Studio (ReportsController – GetRhinoReport).
			Step 6: The API endpoint validates user authentication/RBAC claims and passes the date parameters to the Report Repository/Service. An optimized, read-only (.AsNoTracking()) Entity Framework Core query is executed against Microsoft SQL Server to fetch matching records (Timestamp >= StartDate && Timestamp <= EndDate). The following entities are involved in the process: Rhino Sighting / Observation Entity: • RhinoId (PK) • TagNumber • SpeciesType • RecordedDate • LocationCoordinates

			• HealthStatus [ALT]
			Step 7: The backend calculates aggregate metrics, maps the results into a RhinoReportResultDto payload, and returns an HTTP 200 OK response to the Angular frontend.
			Step 8: The system records the view action in the AuditLog table for auditing and compliance tracking. AuditLog Entity: • AuditLogID (PK) • UserID (FK) • Timestamp • LogEntry ("Viewed Rhino Report for period X to Y")
			Step 9: The system renders the generated Rhino Report on the screen, displaying metric summary cards (e.g., total observations, active alerts) along with a structured data table of observation records.
Alternate courses:	ALT-Step 3: The user changes their mind and selects the "Clear Filters" or "Cancel" button. The form resets. Terminate use case.		
	ALT-Step 4: The Start Date is greater than the End Date. The system displays an inline validation error: <i>"Start date cannot be greater than end date."</i>		
	ALT-Step 6: The system fails to retrieve data or encounters a server timeout, returning a toaster message: <i>"Failed to generate report. Please try again later."</i> Terminate use case.		
Conclusion:	The requested Rhino Report for the specified date range is successfully generated, displayed on the screen, and logged in the audit trail.		
Post-condition:	The system presents the structured Rhino Report to the user without altering underlying observation records.		
Business rules:	• Only users with appropriate role-based permissions (e.g., Admin, Manager, Ranger Lead) can view Rhino Reports. • Users must be authenticated and maintain an active session.		

Implementation constraints and specifications:	Pre-structured data projection utilizing read-only (.AsNoTracking()) EF Core query execution for optimal query performance.
Assumptions:	Rhino observation and health tracking data are correctly populated in the database.
Open issues:	None

Table 7: Technical Narrative 12.7 View Rhino Report

2.4.2.4 View Elephant Report

INTEGRATED NGO MANAGEMENT SYSTEM			
Author:	Amahle Khoza		Date: 24/07/2026 Version: 1.0
Use case name:	View Elephant Report		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
Use case id:	12.8		
Priority:	High		
Source:	EPI-USE		
Primary business actor (PBA):	Administrator		
Primary system actor (PSA):	None		
Other participating actors:	None		
Other interested stakeholders:	None		
Description:	This use case describes how an authorized user generates and views an Elephant Report for a specified date range. The system processes parameter inputs, executes data aggregation queries, and presents structured telemetry, herd movement, and activity data.		
Pre-condition:	The user must be logged into the system and possess appropriate permissions.		
Trigger:	The user wants to view an Elephant Report for a designated timeframe.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action
	Step 1: The user navigates to the Reports module and selects the "Elephant Report" option.		Step 2: The system loads and displays the Elephant Report parameter view, rendering Start Date and End Date input controls and a "Generate Report" action button. The Generate Report button remains disabled until both date inputs are populated.
	Step 3: The user inputs the desired Start Date and End Date values and		Step 4: The system validates the date parameters via Angular frontend form controls prior to API

	selects the "Generate Report" button. [ALT]		submission. Validation criteria include: • Start Date field is not null or empty. • End Date field is not null or empty. • Start Date is less than or equal to End Date. [ALT]
			Step 5: The system routes the query request via the Angular frontend and Report Service through the Angular HTTP Client to the corresponding Web API endpoint in Visual Studio (ElephantReportsController – GetElephantReport).
			Step 6: The API endpoint enforces JWT security and role claims, delegating the request to the Elephant Report Service. An optimized, read-only (.AsNoTracking()) LINQ query runs against Microsoft SQL Server to aggregate matching telemetry and event logs (LoggedAt >= StartDate && LoggedAt <= EndDate). The following entities are involved in the process: Elephant Observation / Telemetry Entity: • ElephantId (PK) • IdentifierCode • HerdIdentifier • LoggedAt • ObservedSector • ActivityType [ALT]

			Step 7: The server projects the aggregated datasets into an ElephantReportResultDto structure and transmits an HTTP 200 OK response back to the client UI.
			Step 8: The backend writes a transaction log entry into the AuditLog table for historical tracking and regulatory auditing. AuditLog Entity: • AuditLogID (PK) • UserID (FK) • Timestamp • LogEntry ("Viewed Elephant Report for period X to Y")
			Step 9: The system renders the populated Elephant Report layout, presenting high-level herd metric summary panels alongside structured detailed event tables.
Alternate courses:	ALT-Step 3: The user selects "Cancel" or clears the form controls. The report generation process terminates.		
	ALT-Step 4: Date range validation fails (Start Date > End Date). The system presents an error alert: <i>"Invalid temporal range. Start date must precede End date."</i> Go back to Step 3.		
	ALT-Step 6: Database query processing fails. The system throws an error toaster: <i>"Failed to fetch elephant report metrics."</i> Terminate use case.		
Conclusion:	The Elephant Report for the requested date range is successfully generated, displayed in the web application, and recorded in system audit logs.		
Post-condition:	The Elephant Report is displayed on-screen in a read-only view. No underlying telemetry data is modified.		
Business rules:	• Only authenticated users with role-based permissions (e.g., Admin, Manager, Conservationist) can view Elephant Reports. • Session bearer token must be valid.		

Implementation constraints and specifications:	High-performance aggregate data retrieval using indexed temporal fields and non-tracking EF Core queries.
Assumptions:	Elephant movement and herd telemetry records are populated and accessible in SQL Server.
Open issues:	None

Table 8: Technical Narrative 12.8 View Elephant Report

2.4.2.5 View K9 Report

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Christopher Bailey	Date: 25 July 2026 Version: 1	
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 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 from the backend API and display the report to the super admin. The super admin will then have the option to save the report as a pdf or export CSV. The use case ends once the super admin saves 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	
		Manual Action	Automated Action
	Step 1: The super admin requests to view the K9 report.		
	Step 2: The admin navigates to the Admin tab and selects Reports then K9 Report. [ALT]		Step 3: The Angular frontend calls the StructuredReportService service to send a GET request to the C# Web API backend StructuredReports controller through the GetK9 endpoint. The

		backend retrieves the report details from the following entities: K9 Entity: • K9Id (PK) • AnimalId (FK) Animal Entity: • AnimalId (PK) • AnimalName • AnimalSpecies (Breed) • AnimalGender • AnimalDateOfBirth • ActiveStatus K9Role Entity: • K9RoleId (PK) • RoleName K9K9Role Entity: • K9Id (PK, FK) • K9RoleId (PK, FK) K9User Entity: • AssignmentId (PK) • K9Id (FK) • UserId (FK) • StartDate • EndDate User Entity: • UserId (PK) • UserName • UserSurname AnimalProject Entity: • AnimalId (PK, FK) • ProjectId (PK, FK) Project Entity: • ProjectId (PK) • ProjectName • StartDate • EndDate
--	--	---

			• IsActive Inventory Entity: • InventoryItemId (PK) • ProjectId (FK) • InventoryTypeId (FK) • WarehouseId (FK) • ModelName • Value InventoryType Entity: • InventoryTypeId (PK) • Name Warehouse Entity: • WarehouseId (PK) • Name [ALT]
			Step 4: The system will then record the transaction in the AuditLog table to maintain a history of actions performed within the system. This is handled in the backend after the successful creation of the report. The system will store information such as the authenticated user performing the action, the timestamp of the operation, and a descriptive log entry indicating that a K9 report has been created. AuditLog Entity: • AuditLogID (PK) • UserID (FK) • Timestamp • LogEntry
			Step 5: The system loads and displays the K9 Operational Report screen using the Angular frontend with the following controls: A header section containing:

			• A heading displaying K9 Operational Report. • A description displaying K9 register, handler assignments and operational activity. • A reporting period displaying the selected date range. • A Generated On value displaying the report generation date and time. • A Generated By value displaying the user who generated the report. A report filter section containing: • A Start Date field allowing the user to select the beginning of the reporting period. • An End Date field allowing the user to select the end of the reporting period. • Preset date range buttons including: • This Month. • This Quarter. • Year to Date. • Last 12 Months. • A Generate Report button to regenerate the report for the selected period. • A Download PDF button to export the report as a PDF document. • A Download CSV button to export the report as a CSV file. A KPI summary section containing:
--	--	--	--

			• A Dogs on Register KPI displaying the total number of active K9s. • A Retired Dogs KPI displaying the number of inactive K9s. • An Active Handlers KPI displaying the number of handlers currently assigned to operational K9s. • An Active Assignments KPI displaying the number of current handler assignments. • An Active K9 Register section containing a table with the following columns: • Name displaying the K9's name. • Breed / Species displaying the K9 breed or species. • Gender displaying the K9's gender. • Age displaying the calculated age of the K9. • Current Handler displaying the currently assigned handler. • Operational Roles displaying the operational role or roles assigned to the K9. A Current Handler Assignments section containing a table with the following columns: • Dog displaying the K9 name. • Handler displaying the assigned handler. • Assignment Started displaying the assignment start date.
--	--	--	--

			• Days Assigned displaying the number of days the current assignment has been active. • A fallback message displaying "No current handler assignments." when no current assignments exist. A Completed Handler Assignments section containing a table with the following columns: • Dog displaying the K9 name. • Handler displaying the assigned handler. • Assignment Started displaying the assignment start date. • Assignment Ended displaying the assignment end date. • Days Assigned displaying the duration of the completed assignment. • A fallback message displaying "No handler assignments were completed during this reporting period." when no completed assignments exist. An Operational Roles section containing a table with the following columns: • Dog displaying the K9 name. • Current Handler displaying the assigned handler. • Operational Roles displaying the operational role or roles assigned to the K9. A Role Distribution section containing:
--	--	--	---

			• A breakdown chart displaying the number of K9s assigned to each operational role. • A fallback message displaying "No K9 roles have been assigned." when no role assignments exist. A Project Assignments section containing a table with the following columns: • Project displaying the project name. • K9 displaying the assigned K9. • Current Handler displaying the currently assigned handler. • Project Status displaying whether the project is Active or Closed. • Project Started displaying the project start date. • Project End displaying the project end date. • A fallback message displaying "No K9 project assignments." when no project assignments exist. A K9s by Project section containing: • A breakdown chart displaying the number of K9s assigned to each project. • A fallback message displaying "No project assignments." when no project assignments exist. An Inventory Supporting K9 Operations section containing a table with the following columns:
--	--	--	--

			• Project displaying the project name. • Inventory Item displaying the inventory item name. • Inventory Type displaying the inventory category. • Warehouse displaying the warehouse in which the item is stored. • Estimated Value displaying the recorded monetary value of the inventory item. • A fallback message displaying "No inventory allocated to K9 projects." when no inventory has been allocated. An Inventory by Type section containing: • A breakdown chart displaying the number of inventory items grouped by inventory type. • A fallback message displaying "No inventory has been allocated." when no inventory records exist. An Observations section containing: • A narrative list highlighting operational findings identified from the report data, including K9s without handlers, K9s without operational roles, K9s without projects, projects without allocated inventory, handlers responsible for multiple K9s, K9s with multiple active handler assignments, and the total recorded value of
--	--	--	---

			inventory allocated to K9 projects. A fallback message displaying "No operational or data quality issues were detected for K9 deployments during the selected reporting period." when no issues are identified.
	Step 6: The super admin selects "Download PDF" button. [ALT]		Step 7: The Angular frontend calls the StructuredReportService service to send a GET request to the C# Web API backend StructuredReports controller through the GetK9Pdf endpoint. The backend validates the selected reporting period and retrieves the required K9, animal, handler, assignment, project, inventory, inventory type and warehouse data from the database. The generated data is then rendered using the QuestPDF library to compose the report into a structured PDF document. The backend records the PDF export in the audit log and returns the generated PDF file as a file response. The Angular frontend receives the PDF as a Blob and automatically downloads it to the user's device using the filename format: k9-report_YYYY-MM-DD_YYYY-MM-DD.pdf
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 Admin option. Continue to step 3.		
	[ALT] Step 3: The system fails to retrieve the report data. The system will display an error message.		
	[ALT] Step 6a: The super admin does not save the pdf, just views it.		

	[ALT] Step 6b: The super admin selects “Export CSV”. The Angular frontend calls the StructuredReportService service to send a GET request to the C# Web API backend StructuredReports controller through the GetK9Csv endpoint. The backend validates the selected reporting period and retrieves the required K9, animal, handler, assignment, project, inventory, inventory type, and warehouse data from the database. The backend iterates through each report section and converts the report tables into Comma-Separated Values (CSV) format. The backend records the CSV export in the audit log and returns the generated CSV file as a file response. The Angular frontend receives the CSV file as a Blob and automatically downloads it to the user's device using the filename format: k9-report_YYYY-MM-DD_YYYY-MM-DD.csv
Conclusion:	The K9 report has been saved and downloaded.
Post-condition:	K9 report is saved.
Business Rules:	• Super admins can view K9 report.
Implementation constraints and specifications:	• None
Assumptions:	• None
Open issues:	• None

Table 9: Technical Narrative 12.9 View K9 Report

2.4.2.6 View Machine Learning Dashboard

INTEGRATED NGO MANAGEMENT SYSTEM		
AUTHOR:	Stephan Bredenhann	Date: 26 July 2026 Version: 1.0
USE CASE NAME:	Backup System	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	12.10 View Machine Learning	
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 views the predictive analytics ML Dashboard, covering budget forecasting, team size recommendation, risk scoring, and inventory demand across all ongoing projects. The dashboard is never pre-computed: every time it is opened, the “.NET 8” API (PredictionsController, PredictionService, TrainingDataService) assembles a fresh feature matrix from live operational data and dispatches it to a separate Python FastAPI sidecar (ml-service/app.py), which fits scikit-learn pipelines and returns predictions with cross-validated diagnostics. The use case concludes with the Super Admin reviewing project risk cards, drilling into a single project's detail, and checking predicted stock shortfalls.	
Pre-condition:	The Super Admin must be authenticated and mapped to an active user record. PredictionAccessService.EnsureCanViewPredictionsAsync() enforces authentication only ;there is no additional role restriction beyond the domain's own access rules.	
Trigger:	The Super Admin selects “ML Dashboard” from the main navigation menu.	
	Actor Action	System Response

Typical Course of Events:		Manual Action	Automated Action
	Step 1: Super Admin selects "ML Dashboard" from the main navigation menu.		Step 2: The Angular frontend sends an authenticated GET request to "/api/predictions/dashboard". PredictionsController first calls PredictionAccessService.EnsureCanViewPredictionsAsync() to confirm the caller is an authenticated, active user.
			Step 3: TrainingDataService runs 8 set-based queries against Projects, ProjectDeadlines, Locations, AnimalProjects, ProjectVendors, Permits, TeamRoles, UserProjects/UserProjectRoles/ProjectTeamRoleMembers, Groups/UserGroups, Inventory/InventoryTypes, ProcurementLines, UserSkills, ProjectFeedbacks, and Expenses, assembling the feature matrix (ProjectFeatureCatalogue, 25 base + 4 progress features) for every ongoing project alongside the completed-project training set.
	Step 4: Super Admin selects "Create Backup Now" and confirms "Create a full database backup now?". [ALT]		Step 4: PredictionService builds four concurrent FitPredictRequest payloads (budget, team, risk, stock) and posts each to the Python sidecar's "POST /fit-predict" endpoint at the configured ModelService:BaseUrl (default "http://127.0.0.1:8000", 5-second timeout). [ALT]
			Step 5: For each task with at least 8 completed training rows, the FastAPI service runs its scikit-learn pipeline (SmoothedTargetEncoder → SelectKBest → StandardScaler → RidgeCV / LogisticRegression), fits the model, cross-validates it (leave-one-out for the regression tasks, stratified k-fold for risk classification), and returns predictions plus a diagnostics block (r2Loo, mae, accuracyCv, selected features, coefficients).
			Step 6: PredictionService assembles the MIDashboardDto, combining each ongoing project's budget/team/risk predictions and PredictionMetaDto confidence data with the stock array (30/60/90-day demand versus stock-on-hand) and a modelInfo summary, then returns "200 OK". MIDashboardDto:

			• generatedAtUtc • projects[] • stock[] • modelInfo
	Step 7: Super Admin reviews the dashboard: project cards sorted by risk (highest first) showing budget delta, team recommendation, and a colour-banded risk percentage; a stock shortfall panel; and a model info footer reporting service health and sample sizes.		Step 9: BackupService.RestoreAsync() looks up the backup, validates the file exists inside the configured backup directory, snapshots the current disk “.bak” files, acquires the semaphore, and calls MaintenanceState.Enter() ;from this point every request outside “/api/backups/*” receives “503”. [ALT]
	Step 8: Super Admin selects a project card to drill into its detail page.		Step 9: The frontend navigates to the project detail route and sends an authenticated GET request to “/api/predictions/projects/{projectId}”, which returns that project's budget, team, and risk predictions (including the top contributing risk factors) for display. [ALT]
	Step 10: Super Admin returns to the dashboard and selects “View Stock Predictions”.		Step 11: The frontend calls “GET /api/predictions/stock”, and the stock shortfall panel renders predicted demand against available and allocated stock for each inventory type across the 30, 60, and 90-day horizons.
	Step 12: Super Admin selects “Refresh Dashboard”.		Step 13: The frontend re-issues “GET /api/predictions/dashboard”, causing all four models to be refit against the now-current live data (Steps 2–6 repeat).
	Step 14: Super Admin selects		

	"Delete" on a backup entry and confirms "Permanently delete this backup?".		
Alternate courses:	[ALT] Step 3: Fewer than 8 completed projects exist for a given task. PredictionService skips the model fit and applies the task's baseline directly (median spend by category for budget, median team size of well-rated projects for team, a burn-vs-schedule heuristic for risk, average monthly purchasing for stock), setting basis: "Baseline" and a note such as "Needs 8 completed projects to fit a model; N available". [ALT] Step 4: The Python sidecar is unreachable (connection refused or the 5-second timeout elapses). PredictionService catches the failure and falls back to the same baseline computations for every affected task; the dashboard still renders in full, never returning an error to the Super Admin. [ALT] Step 5: The sidecar receives fewer than its own MIN_ROWS = 4 completed rows (a backstop behind the .NET layer's 8-row gate) and returns a validation error with reason="insufficient rows"; PredictionService falls back to baseline for that task. [ALT] Step 9: "GET /api/predictions/projects/{projectId}" is called for a project that is no longer ongoing or does not exist. The controller returns "404 Not Found" and the frontend shows an error toast, returning the Super Admin to the dashboard.		
Conclusion:	The use case concludes when the Super Admin has reviewed the current predictive analytics for ongoing projects, whether summarised on the dashboard, drilled into a single project's detail, or checked as a stock demand forecast.		
Post-condition:	No database state changes as a result of this use case ;it is read-only. The dashboard reflects a model fit computed from live operational data at the moment it was requested; reopening or refreshing it repeats the fit rather than reusing a cached result.		
Business Rules:	• A model is fit only when at least 8 completed projects are available for that task; below this threshold the system always falls back to a baseline rather than failing. • Confidence is reported as "High" only when sample size is at least 20 and R^2 is at least 0.5, "Medium" or "Low" otherwise.		
Implementation constraints and specifications:	• The Python sidecar is a stateless FastAPI service ("ml-service/app.py") reached via a typed HttpClient registered against ModelServiceOptions (BaseUrl, TimeoutSeconds); it is started independently of the ".NET" API with "uvicorn app:app".		

Assumptions:	Historical completed-project data is assumed to be representative of current operating conditions; predictions naturally carry lower confidence for project categories with few historical examples.
Open issues:	None

Table 10: Technical Narrative 12.6 View Machine Learning Dashboard System

2.4.3 System Subsystem

2.4.3.3. Backup System

INTEGRATED NGO MANAGEMENT SYSTEM		
AUTHOR:	Stephan Bredenhann	Date: 26 July 2026 Version: 1.0
USE CASE NAME:	Backup System	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	13.1	
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 manages full database backups and restores through the Backup Management page, and how the system additionally performs backups automatically on a schedule. Manual and scheduled backups both use SQL Server's native BACKUP DATABASE command, staging the resulting “.bak” file through a SQL Server service-account-writable directory before the “.NET 8” API copies it into its own “secure-backups/” store and records metadata in the Backups table. A restore enters maintenance mode (returning “503” on all other requests), restores the database in single-user mode, then reconciles backup metadata from the “.bak” files that survive the rollback. The use case concludes with the Super Admin having created, restored, deleted, or rescheduled backups, or a scheduled backup completing unattended.	
Pre-condition:	The Super Admin must be authenticated (enforced by the global fallback authorization policy in Program.cs). For a restore, at least one backup record with a corresponding “.bak” file must exist.	
Trigger:	The Super Admin navigates to the Backup Management page, or one minute elapses on BackupSchedulerHostedService's polling timer.	
	Actor Action	System Response

Typical Course of Events:		Manual Action	Automated Action
	Step 1: Super Admin navigates to the Backup Management page from the admin menu.		Step 2: The frontend sends three authenticated GET requests in parallel: “/api/backups”, “/api/backups/status”, and “/api/backups/schedule”. “GET /api/backups” additionally checks whether the Backups table is empty while “.bak” files exist on disk (for example, following a prior restore) and, if so, auto-rebuilds the metadata rows from disk before returning the list.
			Step 3: The list, status ({backupInProgress, maintenanceMode}), and schedule ({isEnabled, dayOfWeek, timeOfDay, lastSuccessfulRunAt}) responses populate the backup list table, the status indicator, and the schedule panel respectively.
	Step 4: Super Admin selects “Create Backup Now” and confirms “Create a full database backup now?”. [ALT]		Step 5: BackupService.CreateAsync() acquires a SemaphoreSlim(1,1), sets backupInProgress to true, computes the filename {DatabaseName}_{yyyyMMdd_HH:mm:ss}_{Guid:N}.bak, and resolves the SQL Server staging directory from BackupSettings:SqlServerDirectory or SERVERPROPERTY('InstanceDefaultBackupPath').
			Step 6: The service executes BACKUP DATABASE ... TO DISK = @stagingPath WITH FORMAT, INIT, NAME, SKIP, NOREWIND, NOUNLOAD, STATS = 10, then copies the resulting file into “secure-backups/” and deletes the staging copy. [ALT]
			Step 7: A metadata row is inserted into the Backups table, PruneAutomaticBackupsAsync() removes any automatic backups beyond MaxAutomaticBackups (default 5), an AuditLogs entry (Action="Create") is written, and the semaphore is released; the API returns the new backup and the frontend refreshes the list. Backups Entity: • BackupId (PK) • Date

		• FilePath • Size • IsAutomatic • CreatedByUserId (FK)
	Step 8: Super Admin selects “Restore” on an existing backup entry and confirms “Restore will overwrite the current database and sign everyone out. Continue?”. [ALT]	Step 9: BackupService.RestoreAsync() looks up the backup, validates the file exists inside the configured backup directory, snapshots the current disk “.bak” files, acquires the semaphore, and calls MaintenanceState.Enter() ;from this point every request outside “/api/backups/*” receives “503”. [ALT]
		Step 10: The service copies the file to the SQL staging directory if needed, connects to the master database, and executes ALTER DATABASE SET SINGLE_USER WITH ROLLBACK IMMEDIATE, then RESTORE DATABASE ... WITH REPLACE, then ALTER DATABASE SET MULTI_USER. [ALT]
		Step 11: In a fresh scope, the service re-scans disk for “.bak” files, clears and re-registers the Backups table from those files, re-seeds the BackupScheduleSettings singleton if missing from the restored snapshot, writes an AuditLogs entry (Action=“Restore”), exits maintenance mode, and releases the semaphore. The API responds “Database restored successfully. Please sign in again.”
	Step 12: Super Admin opens the Schedule panel, adjusts the enabled toggle, day of week, and time	Step 13: “PUT /api/backups/schedule” validates that dayOfWeek is 0–6 and timeOfDay is a valid HH:mm value, updates the BackupScheduleSettings singleton row (Id = 1), and returns the updated schedule. [ALT]

	of day, then selects Save.		
	Step 14: Super Admin selects “Delete” on a backup entry and confirms “Permanently delete this backup?”.		Step 15: “DELETE /api/backups/{id}” removes both the “.bak” file from disk and its metadata row, writes an AuditLogs entry (Action=“Delete”), and returns “Backup deleted.”. [ALT]
			Step 16: Independently of any admin action, BackupSchedulerHostedService polls every minute. When the schedule is enabled, the current day matches DayOfWeek, the current time falls within the 5-minute window after TimeOfDay, and no successful run has been recorded for that window, it calls CreateAsync(isAutomatic: true, userId: null) ;repeating Steps 5–7 with IsAutomatic = true and an audit Action of “AutoCreate” ;then updates LastSuccessfulRunAt to the current UTC time on success. [ALT]
Alternate courses:	[ALT] Step 4: A backup is already in progress. “POST /api/backups” returns “409 Conflict”; the frontend shows “A backup is already running. Please wait.” and disables the Create/Delete/Restore buttons until status polling reports backupInProgress = false.		
	[ALT] Step 6: BACKUP DATABASE fails (for example, disk full or the staging directory is not writable). The service returns “500 Internal Server Error”; the semaphore is still released so the system is not left locked, and the frontend shows a toast with the error message.		
	[ALT] Step 8: Super Admin selects Download instead of Restore. “GET /api/backups/{id}/download” validates the file path is inside the configured backup directory via EnsurePathIsInsideBackupDirectory(), then streams the “.bak” file as application/octet-stream; “404” if the backup record or file is missing.		
	[ALT] Step 9: The requested backup's file cannot be found on disk, or the record does not exist. RestoreAsync() returns “400 Bad Request” before entering maintenance mode, so no other users are affected.		
	[ALT] Step 10: RESTORE DATABASE fails mid-operation (for example, a corrupted file). The service returns “500 Internal Server Error”; the frontend shows an error toast, and maintenance mode is still exited so the system remains usable while the Super Admin investigates.		

	[ALT] Step 13: dayOfWeek is outside 0–6 or timeOfDay is not a valid HH:mm value. The endpoint returns “400 Bad Request” and the frontend shows an inline validation message without saving.
	[ALT] Step 15: The backup ID does not exist. “DELETE /api/backups/{id}” returns “404 Not Found”; the frontend shows “Backup file not found on disk.”
	[ALT] Step 16: The scheduler tick fires while the semaphore is already held by a manual backup or restore in progress. TryRunScheduledBackupAsync() cannot acquire the lock immediately, skips this tick, and returns false without retry until the next minute's poll.
Conclusion:	The use case concludes when the Super Admin has successfully created, restored, deleted, or rescheduled a database backup, or when a scheduled backup completes automatically in the background without any admin action.
Post-condition:	The Backups table and “secure-backups/” directory accurately reflect the current set of available backups, BackupScheduleSettings reflects the active schedule, and every operation performed is recorded in AuditLogs.
Business Rules:	• All backup endpoints require authentication; there is no additional role check beyond the global fallback authorization policy. • Only one backup or restore operation may run at a time, enforced process-wide by a SemaphoreSlim(1,1). • Automatic backups beyond MaxAutomaticBackups (default 5) are pruned after every run; manual backups are never pruned automatically. • During a restore, every request outside “/api/backups/*” receives “503” until maintenance mode is exited.
Implementation constraints and specifications:	• SQL Server writes backups to a staging directory that must be writable by the SQL Server service account, distinct from the app's own “secure-backups/” directory (BackupSettings:SqlServerDirectory, auto-detected via SERVERPROPERTY('InstanceDefaultBackupPath') when unset).
Assumptions:	• The SQL Server service account is assumed to have write access to the configured staging directory; if not, PromoteStagingFileToAppStorageAsync() throws a descriptive configuration error rather than failing silently.
Open issues:	• None

Table 11: Technical Narrative 13.1 Backup System

2.4.3.2 Restore System

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Stephan Bredenhann	Date: 26 July 2026 Version: 1.0	
USE CASE NAME:	Restore System	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	13.2		
PRIORITY:	High		
SOURCE:	EPI-USE		
Primary Business Actor (PBA):	Super Admin		
Primary System Actor (PSA):	BackupService, MaintenanceMiddleware		
Other Participating Actors:	SQL Server		
Other Interested Stakeholders:	All authenticated users		
Description:	This use case describes how a Super Admin replaces the entire database with a prior state captured in a “.bak” file. Restore is implemented inside the same BackupService.cs as backup creation (BackupService.RestoreAsync()), but is a distinct, strictly destructive, manually-triggered operation: it can never be scheduled or automated. While it runs, the system enters a process-wide maintenance mode in which MaintenanceMiddleware ;mounted ahead of authentication and routing ;returns “503” for every request, with no carve-out for backup endpoints. After the RESTORE DATABASE command completes, the system reconciles backup metadata from whichever “.bak” files survived the rollback, since the restored database snapshot almost never contains the metadata row for the backup that was just taken.		
Pre-condition:	The Super Admin must be authenticated. At least one Backups record must exist whose “.bak” file is present and readable inside the configured backup directory, and no other backup or restore operation may currently hold the system-wide lock.		
Trigger:	The Super Admin selects “Restore” on a backup row and confirms the destructive warning dialog.		
Typical Course of Events:	Actor Action	System Response	
		Manual Action	Automated Action

	Step 1: Super Admin selects “Restore” on a backup entry on the Backup Management page.		Step 2: The frontend displays a confirmation dialog: “This will overwrite the entire database with the backup from {date}. All current data will be replaced. Everyone (including you) will be signed out. The system will be unavailable during the restore. This action cannot be undone.”. [ALT]
	Step 3: Super Admin confirms by selecting “Restore” in the dialog.		Step 4: The frontend sends “POST /api/backups/{id}/restore”. BackupsController.Restore() calls BackupService.RestoreAsync(id, userId), which looks up the backup by BackupId (AsNoTracking), confirms the “.bak” file exists on disk, and validates the path with EnsurePathIsInsideBackupDirectory(). [ALT]
			Step 5: SnapshotDiskBackupFilesOnDisk() enumerates every “.bak” file in “secure-backups/”, recording each file's path, size, and last-write time. This snapshot is essential: the Backups table itself will be rolled back by the restore, and a backup's own metadata row is inserted only after BACKUP DATABASE completes, so it is almost never present in the restored snapshot.
			Step 6: The service acquires the SemaphoreSlim(1,1) lock [ALT] and calls MaintenanceState.Enter(reason), setting the process-wide IsInMaintenance flag. From this instant, MaintenanceMiddleware ;mounted at the very top of the pipeline, ahead of authentication and routing ;returns “503 Service Unavailable” with the reason message for every subsequent request, with no exception path for “/api/backups/*”. [ALT]
			Step 7: If the app's “secure-backups/” directory differs from the SQL Server staging directory (resolved from BackupSettings:SqlServerDirectory or SERVERPROPERTY('InstanceDefaultBackupPath')), the “.bak” file is copied into the staging directory, overwriting any existing file there; the staging

			copy is tracked for cleanup regardless of outcome. [ALT]
			Step 8: The service connects to the master database (InitialCatalog overridden to “master”, since the target database cannot be restored while any connection to it remains open) and executes, with no command timeout, ALTER DATABASE SET SINGLE_USER WITH ROLLBACK IMMEDIATE, then RESTORE DATABASE FROM DISK = @path WITH REPLACE, STATS = 10, then ALTER DATABASE SET MULTI_USER. [ALT]
			Step 9: ReconcileBackupsFromDiskAsync() runs inside a freshly created IServiceScope, since the original request-scoped DbContext is dead after SINGLE_USER killed its connection. It re-scans “secure-backups/”, removes all existing Backups rows, and re-inserts one row per “.bak” file found (IsAutomatic = false and CreatedByUserId = null in every case, since the original creator identity cannot be recovered), then re-seeds the BackupScheduleSettings singleton if the restored snapshot predates that table. BackupScheduleSettings Entity: • BackupScheduleSettingsId (PK) • IsEnabled • DayOfWeek • TimeOfDay
			Step 10: An AuditLogs entry is written (Action = “Restore”, EntityId = the restored filename), any staging copy is deleted, MaintenanceState.Exit() clears the flag, and the semaphore is released. The API returns “200 OK” with “Database restored successfully. Please sign in again.”
	Step 11: Per the response message, the frontend clears the stored authentication		

	token and redirects the Super Admin ;and every other signed-in user whose next request lands after maintenance mode exits ;to the login page.		
Alternate courses:	[ALT] Step 2: Super Admin selects “Cancel” instead of “Restore” in the confirmation dialog. The dialog closes and no request is sent; the database is left untouched. [ALT] Step 4: The backup ID does not exist, or its recorded “.bak” file cannot be found on disk. RestoreAsync() throws an InvalidOperationException, which the controller returns as “400 Bad Request” with detail “Backup not found.” or “Backup file is missing from disk.”; maintenance mode is never entered. [ALT] Step 6: Another backup or restore is already in progress and the semaphore cannot be acquired immediately. RestoreAsync() throws InvalidOperationException(“A backup or restore is already in progress.”), returned as “400 Bad Request”; the in-flight operation is left undisturbed. [ALT] Step 7: The SQL Server staging directory is not writable, or a WSL/Windows path mismatch prevents the copy. The staging helper throws a descriptive configuration error; the resulting “500 Internal Server Error” surfaces “Restore failed: {exception message}”, and maintenance mode and the semaphore are still released via the finally block. [ALT] Step 8: RESTORE DATABASE fails (a corrupted “.bak” file, insufficient disk space, or a permissions error). The exception surfaces as “500 Internal Server Error” with “Restore failed: {exception message}”; the finally block still exits maintenance mode and releases the semaphore so the system remains available for the Super Admin to retry. [ALT] Step 6 (during maintenance): any user ;including the Super Admin's own subsequent requests ;hitting any endpoint, including “GET /api/backups/status”, during the maintenance window receives “503” with the configured reason; the frontend's HTTP interceptor detects this and displays a blocking banner across the entire application until maintenance ends.		
Conclusion:	The use case concludes when the target database has been fully replaced by the selected “.bak” file's contents, the Backups table and BackupScheduleSettings have been reconciled from disk, and the system has exited maintenance mode and is accepting requests again.		
Post-condition:	The application database matches the state captured in the restored “.bak” file. The Backups table lists exactly the “.bak” files present on disk at reconciliation time, each as		

	a manual entry with no recoverable creator; BackupScheduleSettings exists (re-seeded with defaults if it was missing); an AuditLogs “Restore” entry has been written; and MaintenanceState.IsInMaintenance is false again. Any user whose session depends on data that no longer matches the restored snapshot must sign in again.
Business Rules:	Only one backup or restore operation may run at a time system-wide, enforced by the same SemaphoreSlim(1,1) used for backup creation.
Implementation constraints and specifications:	None.
Assumptions:	The SQL Server service account is assumed able to read the staged “.bak” file and to have permission to execute ALTER DATABASE and RESTORE DATABASE against the target database.
Open issues:	None.

Table 12: Technical Narrative 13.2 Restore System

2.4.4.3 View Audit Log

INTEGRATED NGO MANAGEMENT SYSTEM			
AUTHOR:	Nicholas Bailey	Date: 25/07/2026 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	
		Manual Action	Automated Action
	Step 1: The super admin wants to view all audit logs.		
	Step 2: The super admin hovers over the "Admin" navigation		Step 3: The system sends the request through the Angular frontend and AuditLogService to the relevant HTTP GET API endpoints in Visual Studio

	tab and selects the "Audit Logs" option.		(AuditLogsController – GetAll and AuditLogsController – GetStats). The GetAll endpoint retrieves audit log records from the AuditLogs table in the database through the AuditLogRepository, while the GetStats endpoint retrieves audit activity statistics used for the summary section. AuditLog Entity: • AuditLogID (PK) • UserID (FK) • Timestamp • Action • EntityType • EntityID • Details
			Step 4: The system then displays the Audit Logs page using the Angular frontend, showing the retrieved audit log records and statistics returned from the API. • Page header section: ○ Page title ("Audit Logs") ○ Description text ("View all recorded system actions performed by users.") • Statistics section: ○ Total Logs ○ Last 24 Hours ○ Last 7 Days • Search section: ○ Search input field for filtering audit logs. ○ Search button to retrieve matching records.

			○ Displays the total number of logs found. • Table display section: ○ Audit logs are displayed in a table. ○ The table includes the following columns: ▪ Date ▪ User ▪ Action ▪ Entity ▪ Details • Table behaviour: ○ Each row represents a single audit log record. ○ Records are displayed in chronological order, with the most recent log entries appearing first. ○ Pagination controls allow the super admin to navigate between pages of audit records.
	Step 5: The super admin can view, search, and monitor all recorded system actions performed by users.		[ALT]
Alternate courses:	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:	The system has successfully retrieved a list of all the audit logs within the system.
Business Rules:	• Role Based Access • Ability to Backup Database.
Implementation constraints and specifications:	• Must be user-friendly and easy to use.
Assumptions:	• None
Open issues:	• None

Table 13: Technical Narrative 13.3 View Audit Log

2.5 Technical Context Diagram

2.5.1 Project Management Subsystem

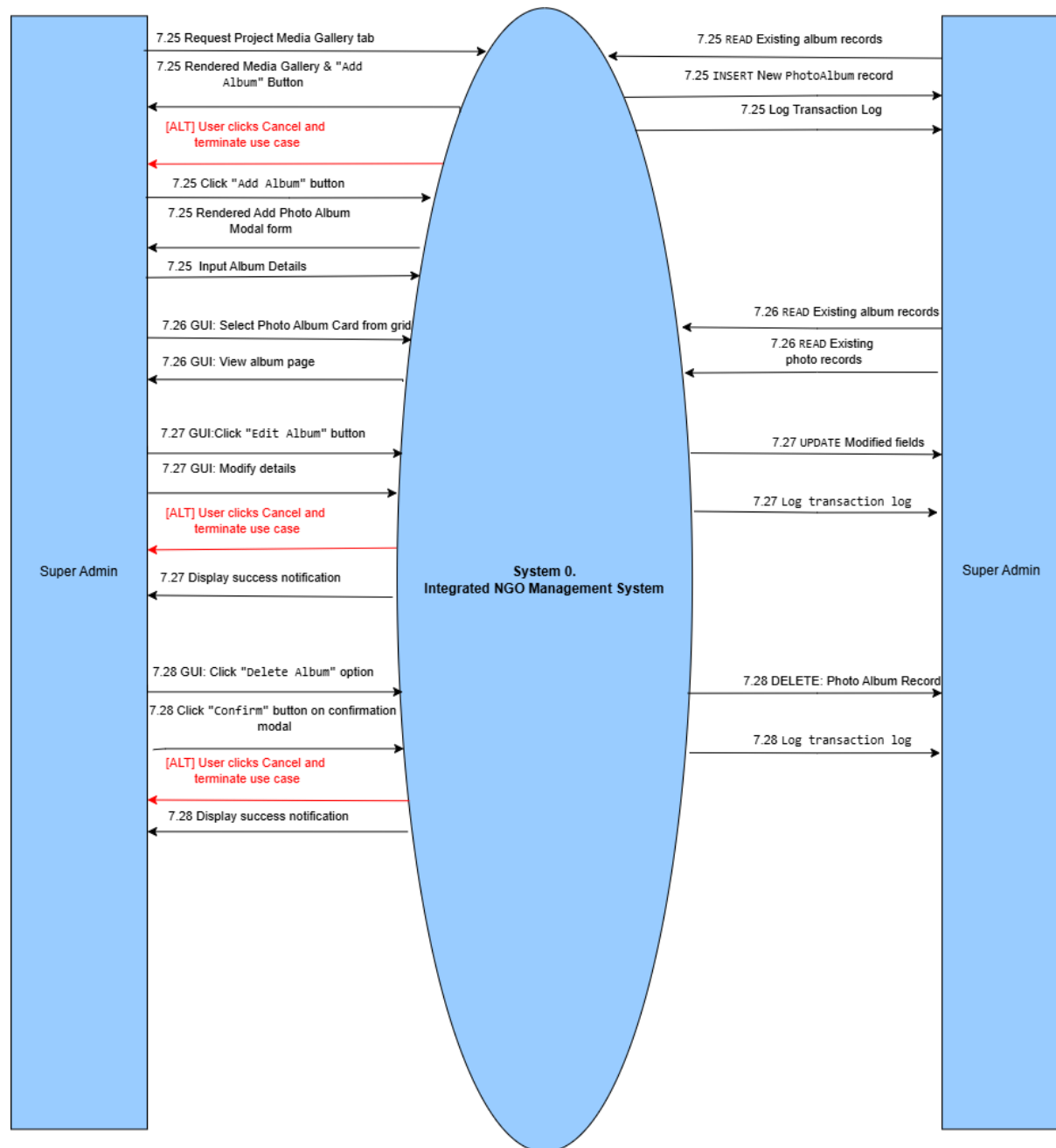


Figure 14: Project Management Technical Context Diagram

2.5.2 Report Subsystem

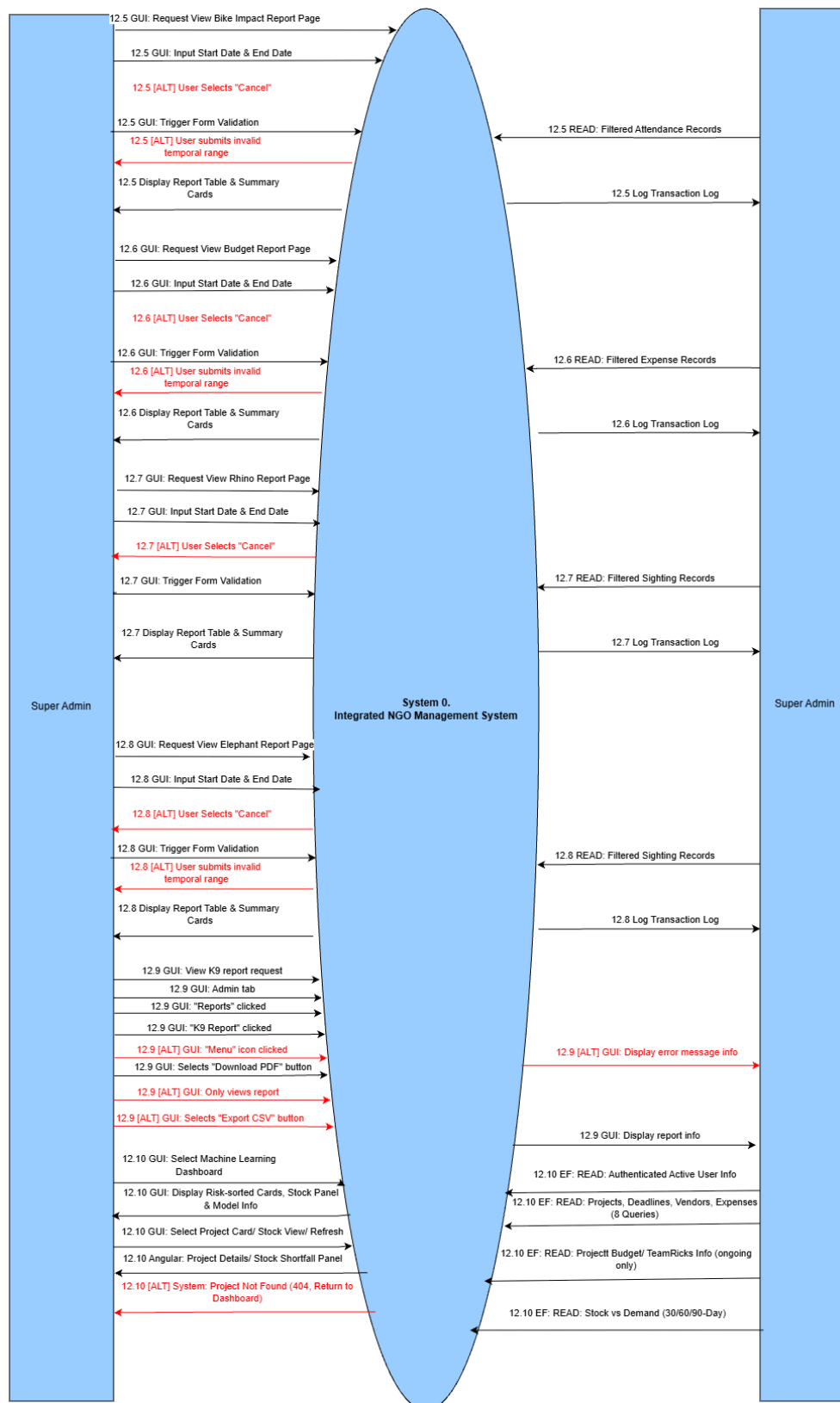


Figure 15: Report Subsystem Technical Context Diagram

2.5.3 System Subsystem

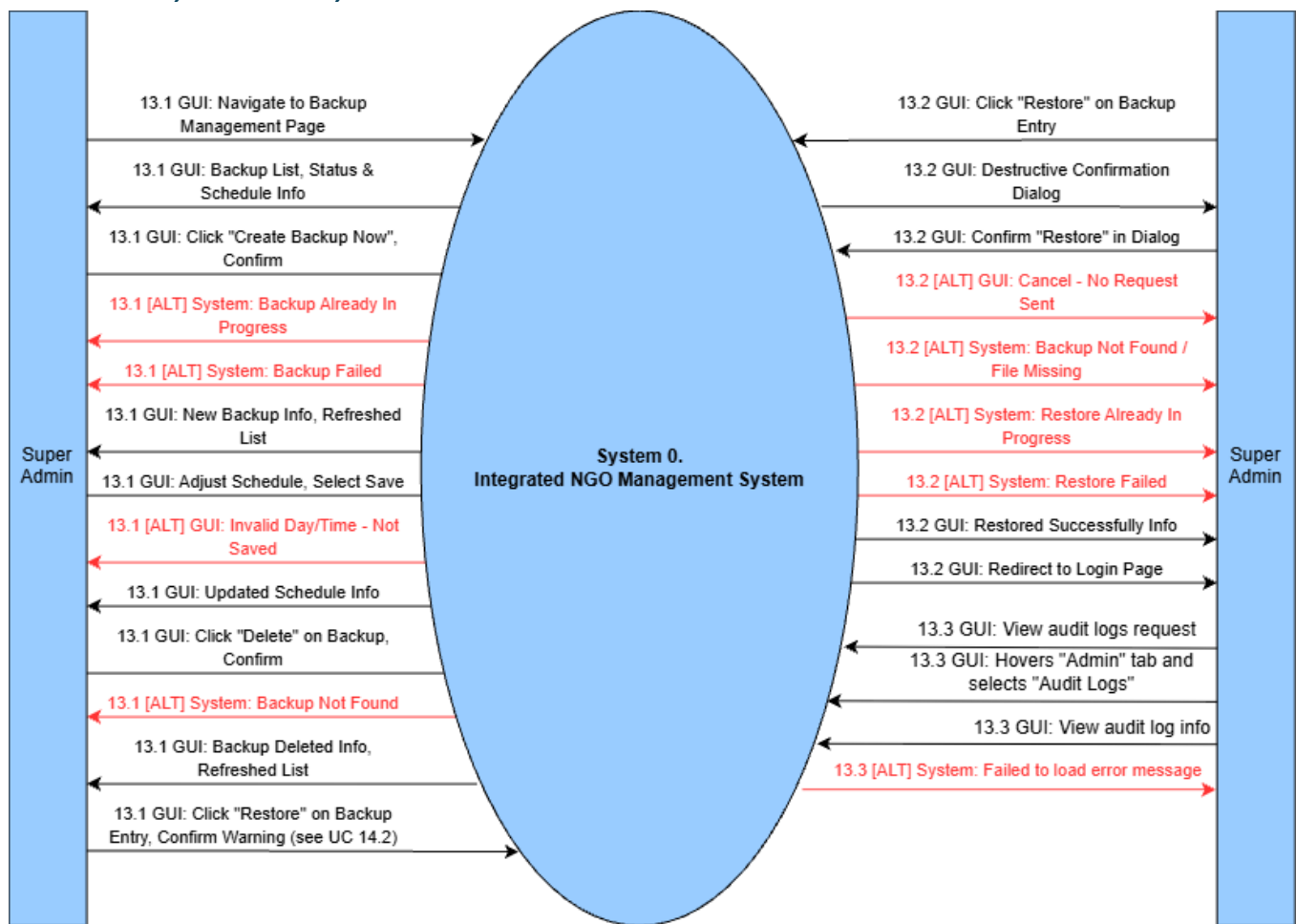


Figure 16: System Subsystem Context Diagram

2.6 Technical Primitive Level Diagram

2.6.1 Project Management Subsystem

2.6.1.1 Add Photo Album

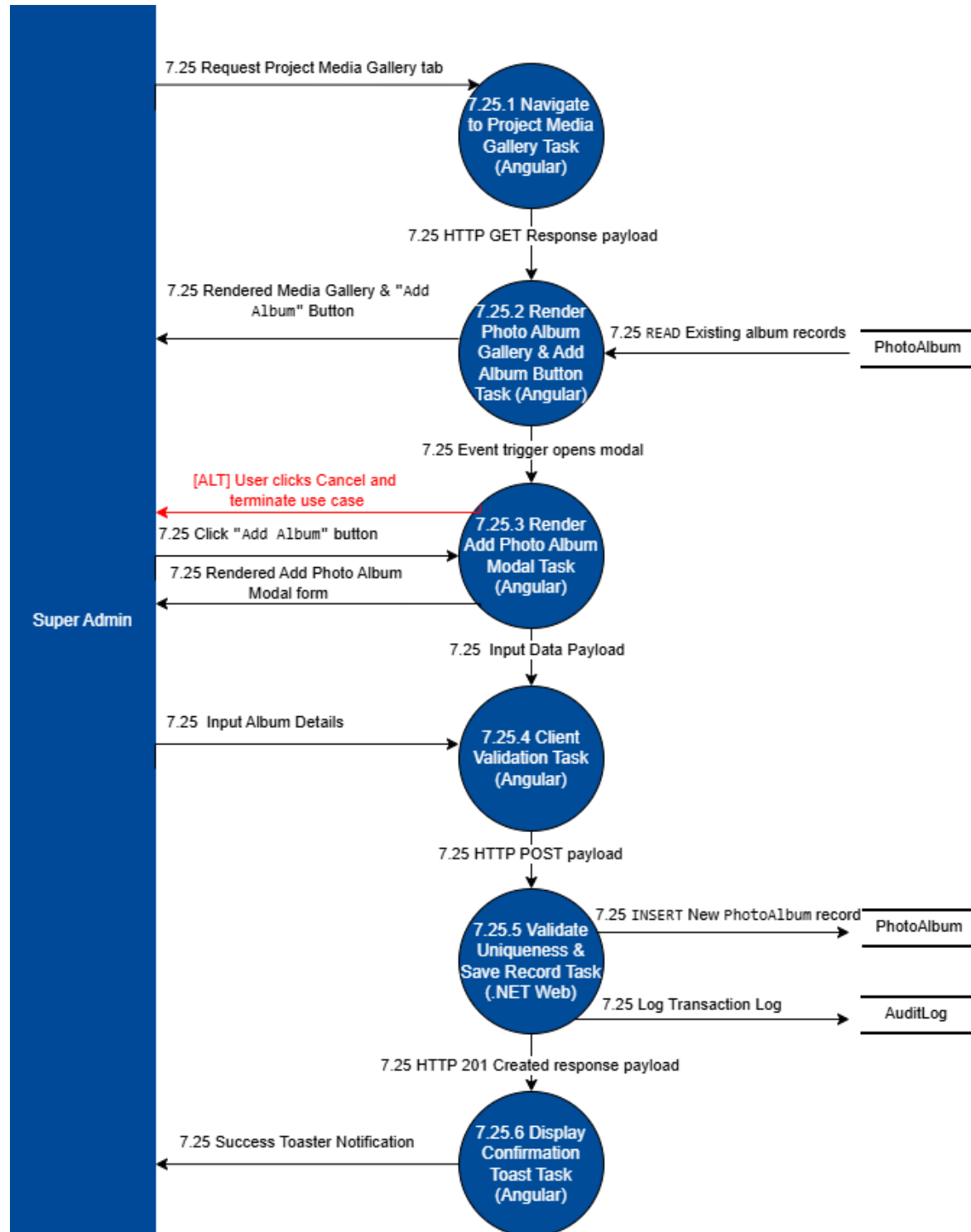


Figure 17: Use Case 7.25 Add Photo Album Technical Primitive Diagram

2.6.1.2 View Photo Album

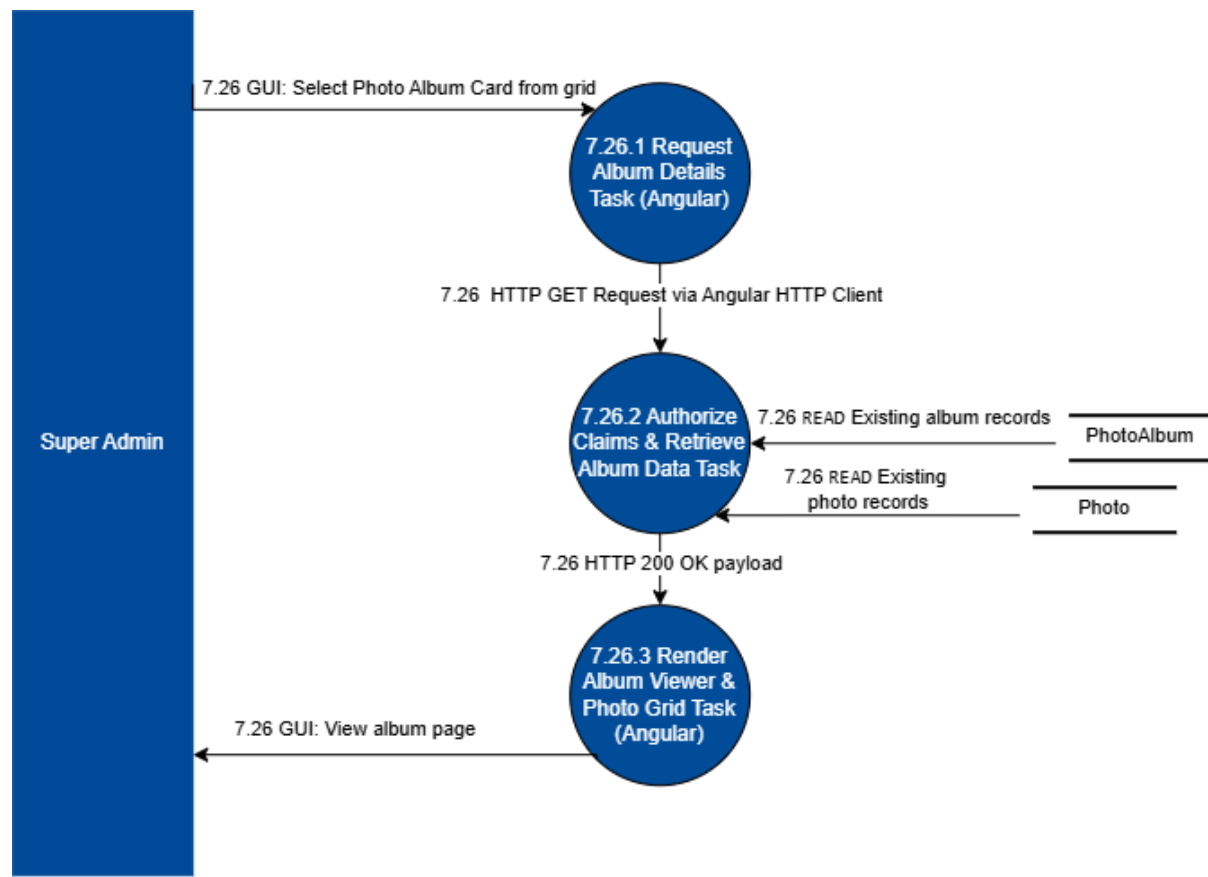


Figure 18: Use Case 7.26 View Photo Album Technical Primitive Diagram

2.6.1.3 Update Photo Album

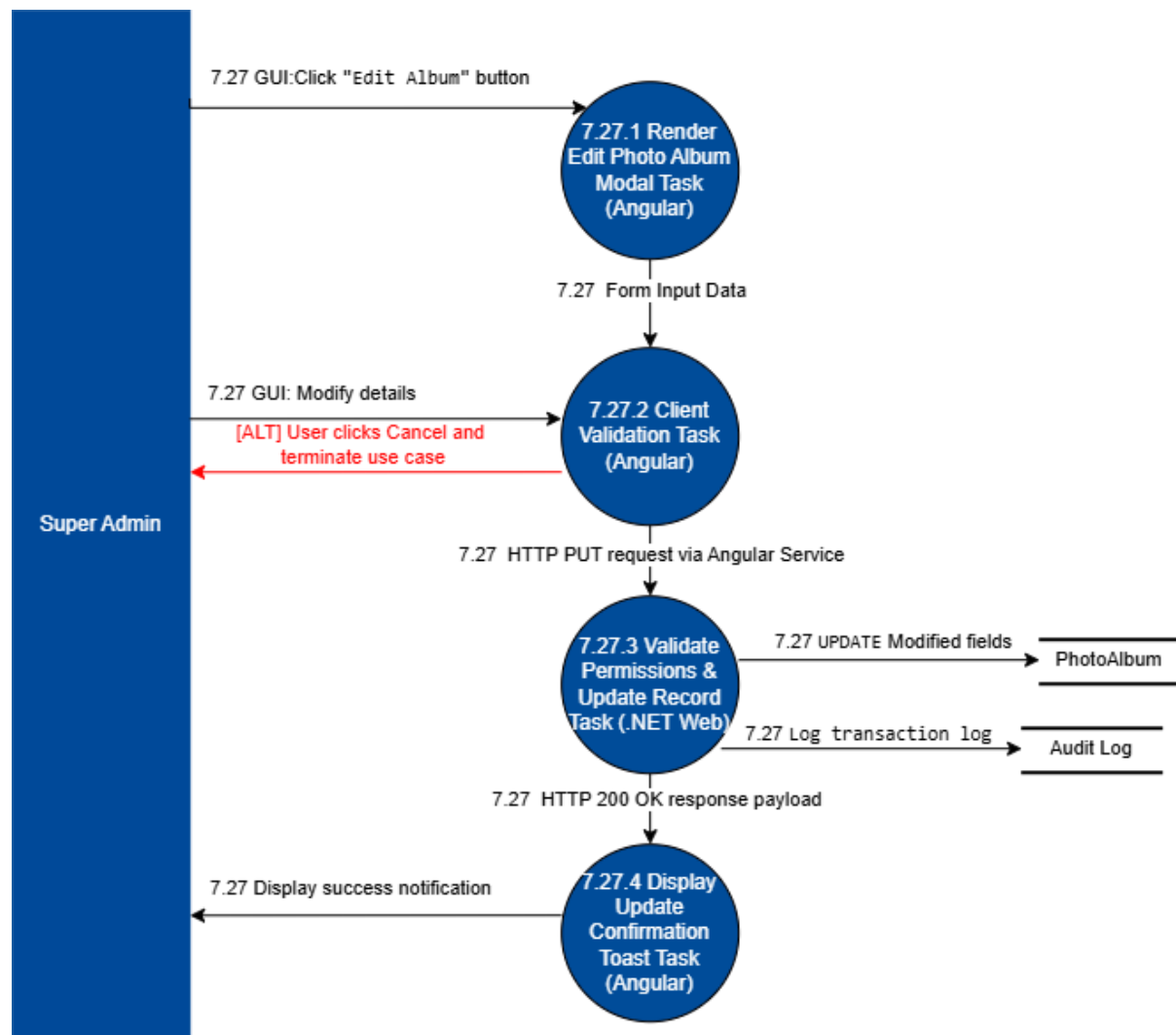


Figure 19: Use Case 7.27 Update Photo Album Technical Primitive Diagram

2.6.1.4 Remove Photo Album

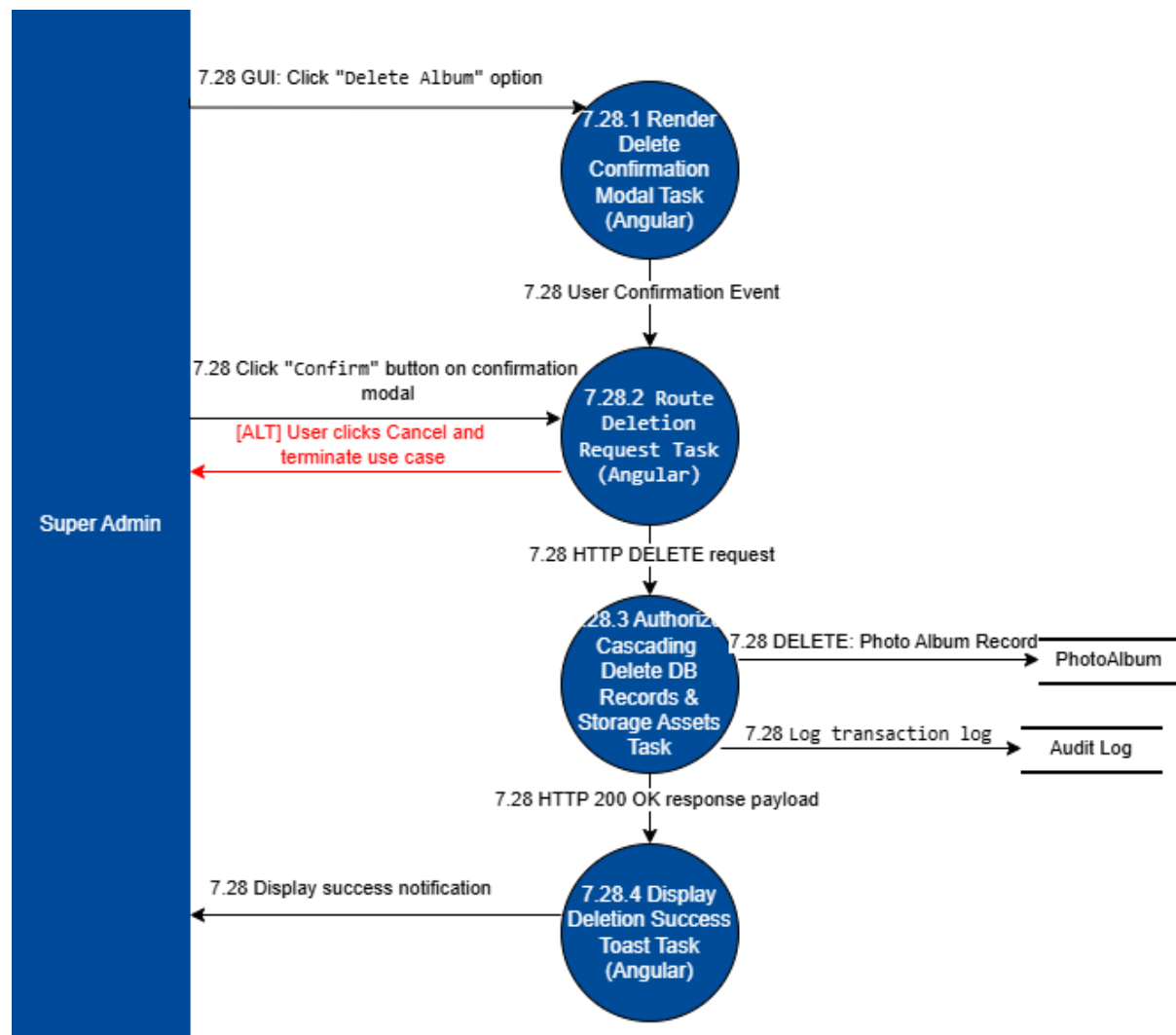


Figure 20: Use Case 7.28 Remove Photo Album Technical Primitive Diagram

2.6.2 Report Subsystem

2.6.2.1 View Bike Impact Report

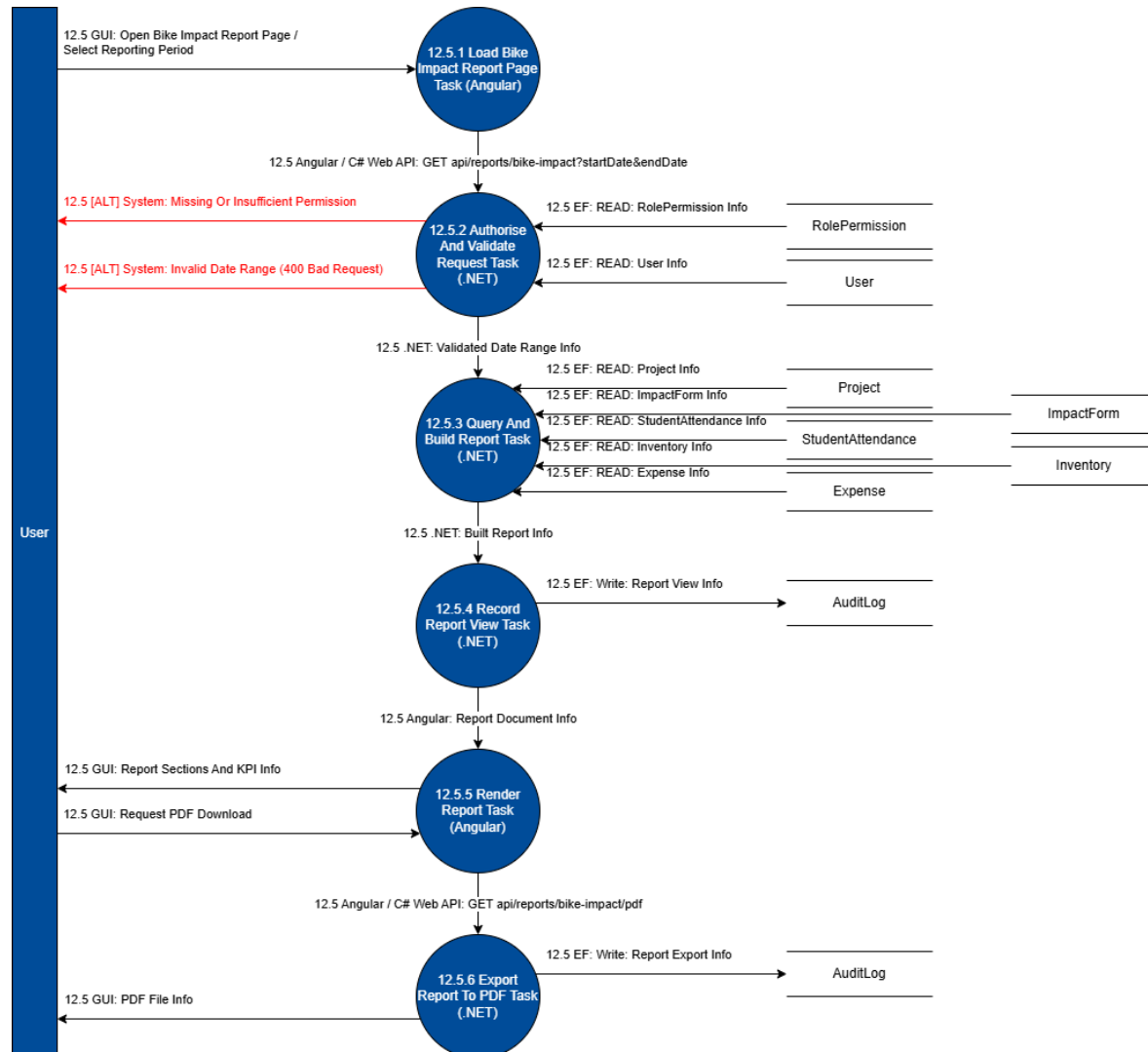


Figure 21: Use Case 12.5 View Bike Impact Report Technical Primitive Diagram

2.6.2.2 View Budget Report

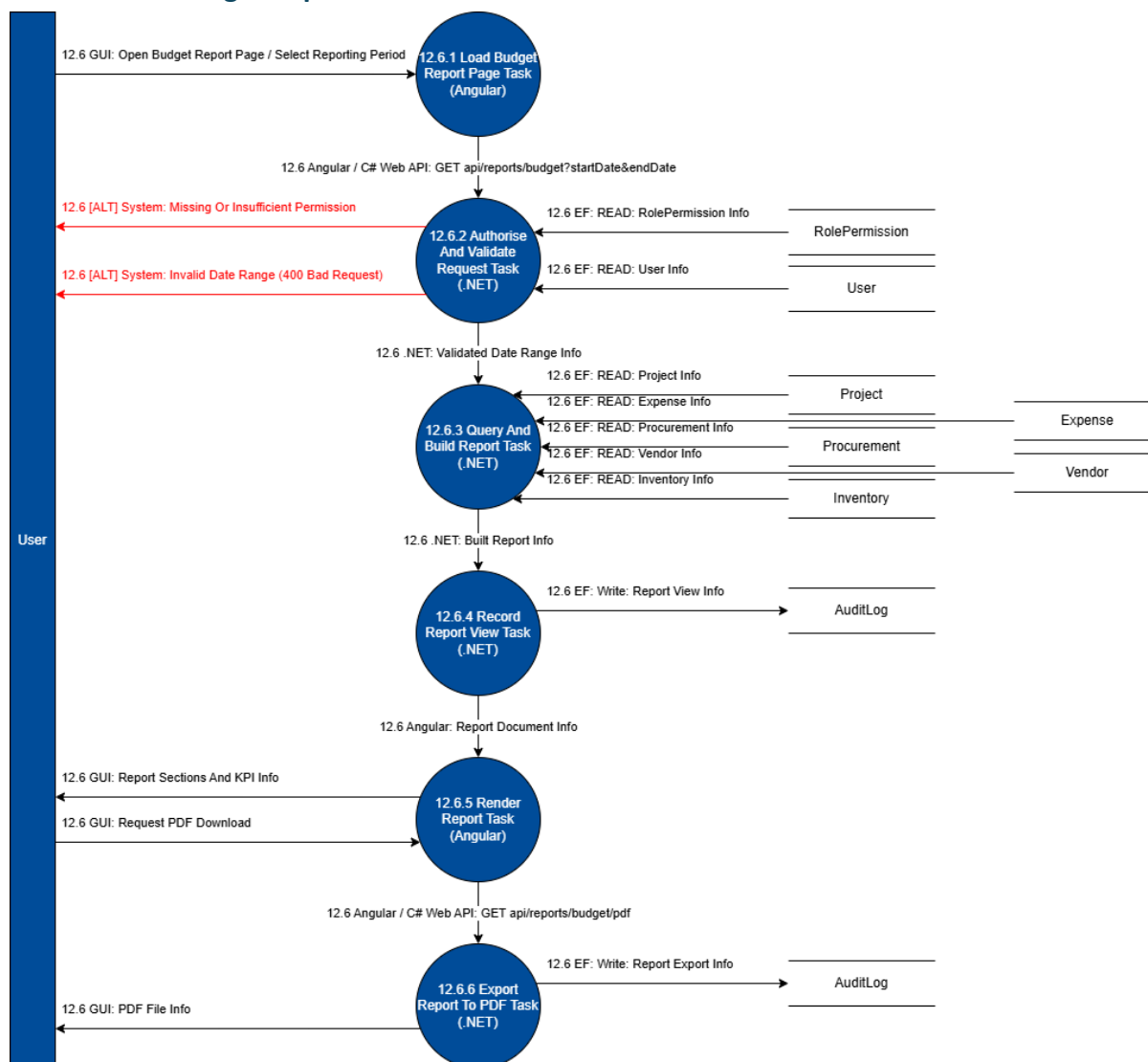


Figure 22: Use Case 12.6 View Budget Report Technical Primitive Diagram

2.6.2.3 View Rhino Report

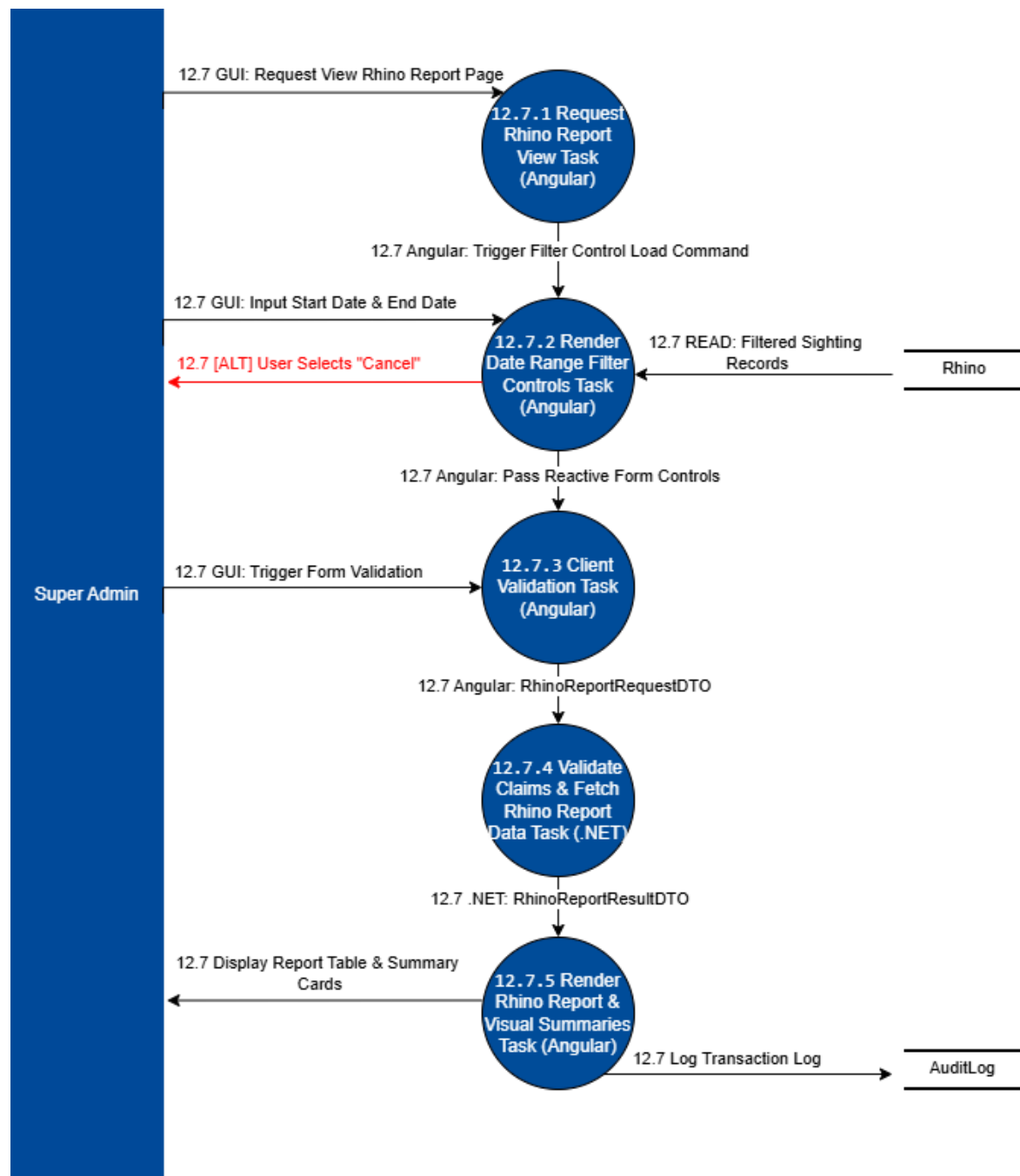


Figure 23: Use Case 12.7 View Rhino Report Technical Primitive Diagram

2.6.2.4 View Elephant Report

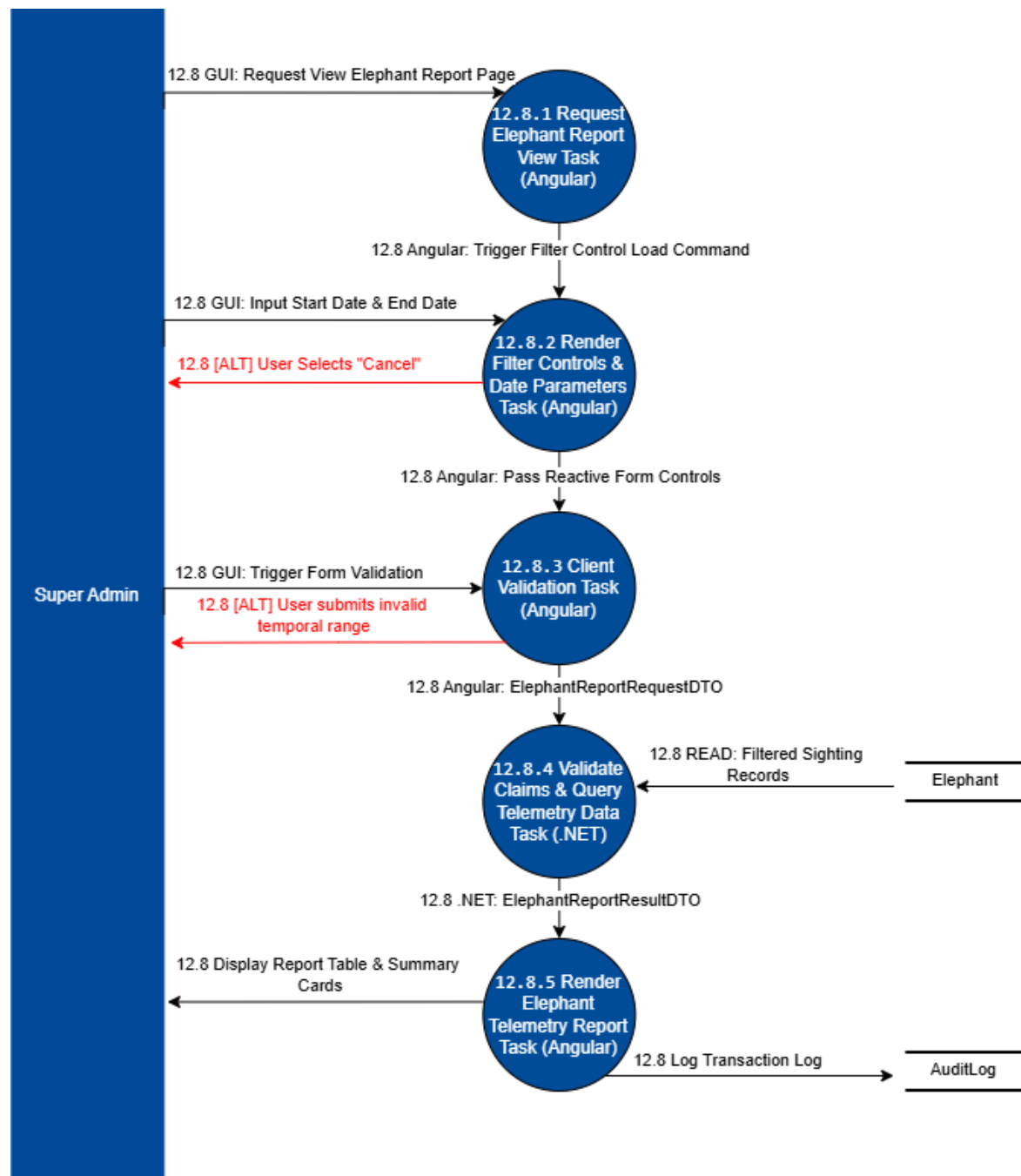


Figure 24: Use Case 12.8 View Elephant Report - Technical Primitive Diagram

2.6.2.5 View K9 Report

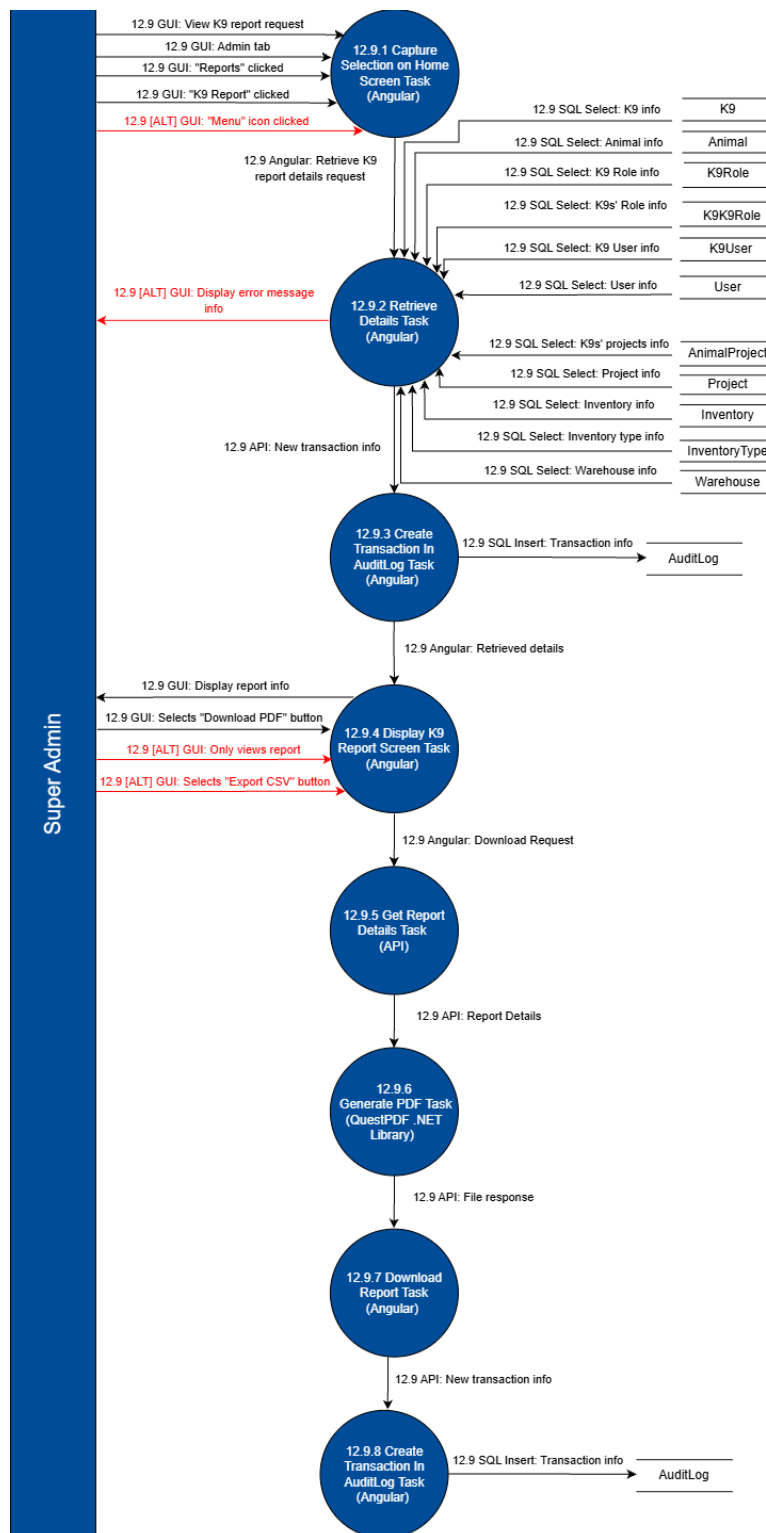


Figure 25: Use Case 12.9 View K9 Report Technical Primitive Diagram

2.6.2.6 View Machine Learning Dashboard

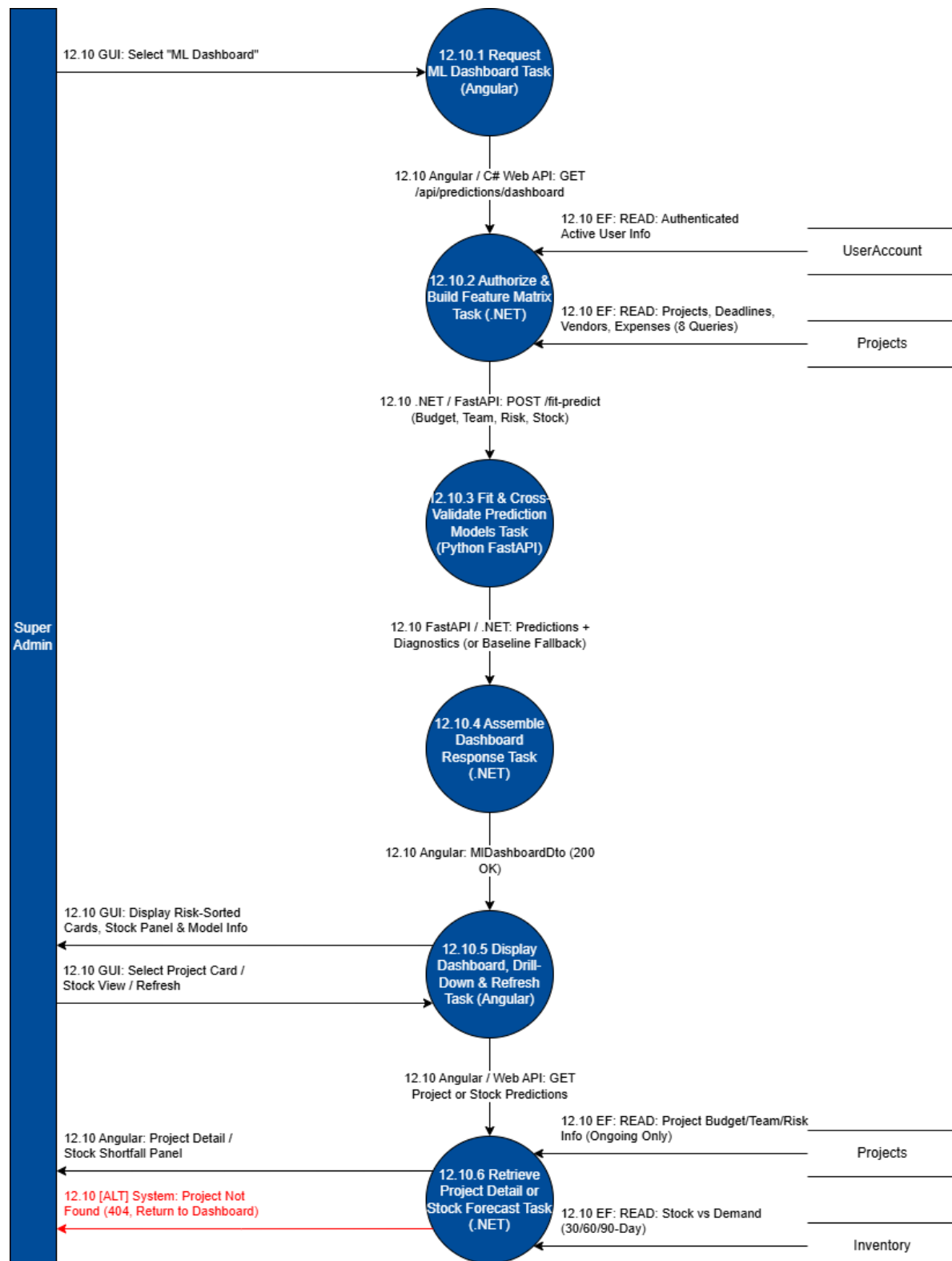


Figure 26: Use Case 12.10 View Rhino Report Technical Primitive Diagram

2.6.3 System Subsystem

2.6.3.1 Backup System

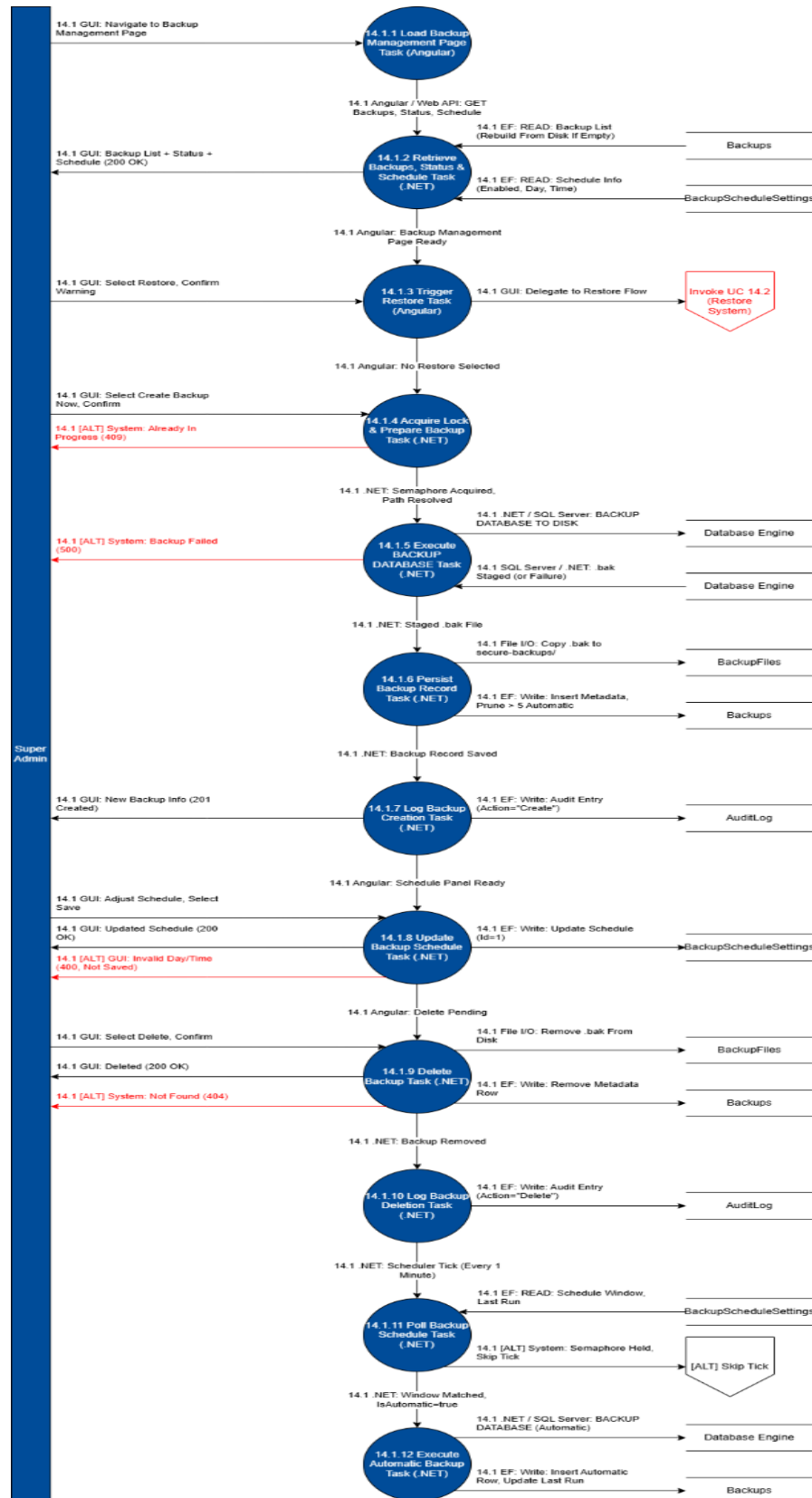


Figure 27: Use Case13.1 Backup System Technical Primitive Diagram

2.6.3.2 Restore System

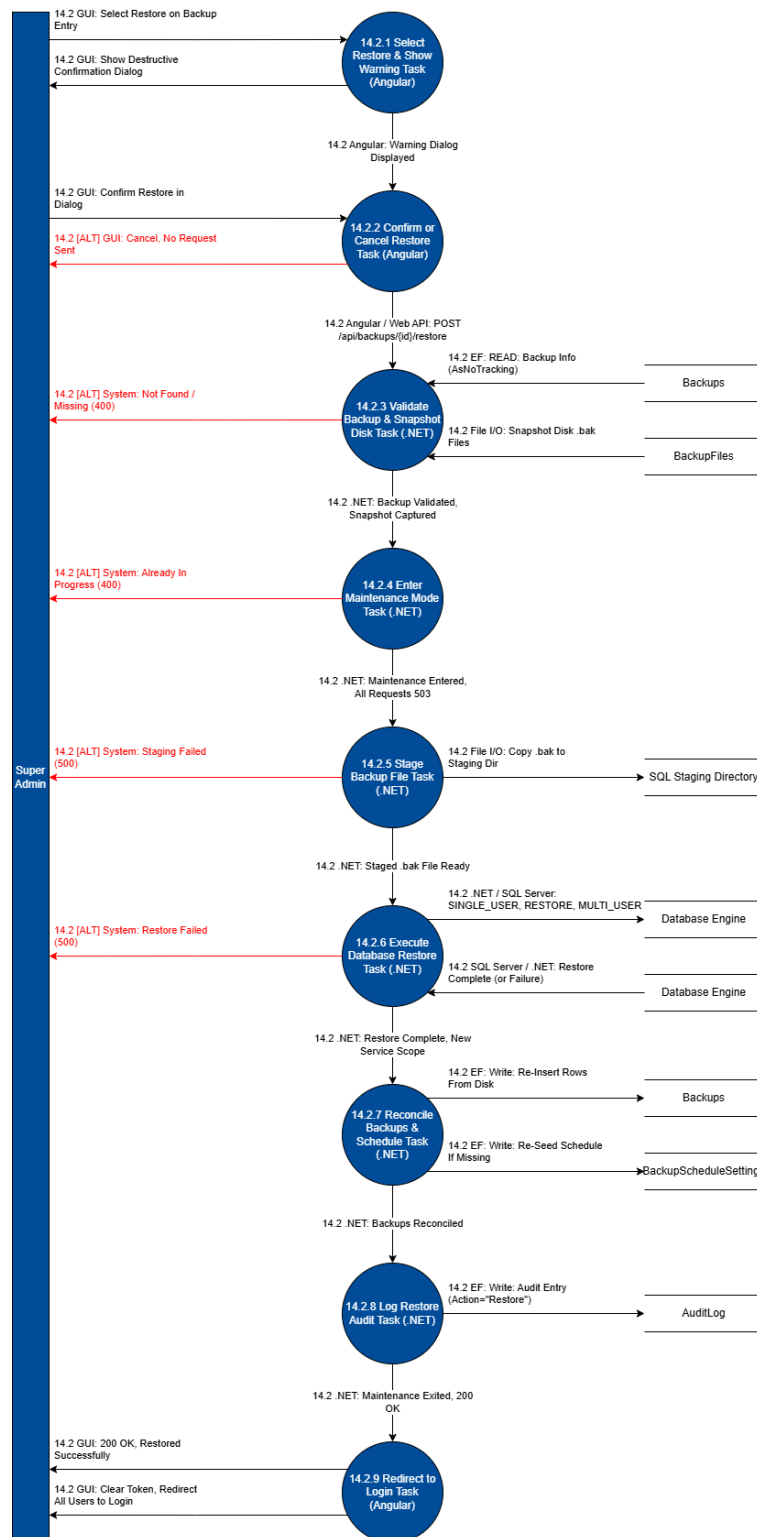


Figure 28: Use Case13.2 Restore System Technical Primitive Diagram

2.6.3.3 View Audit Log

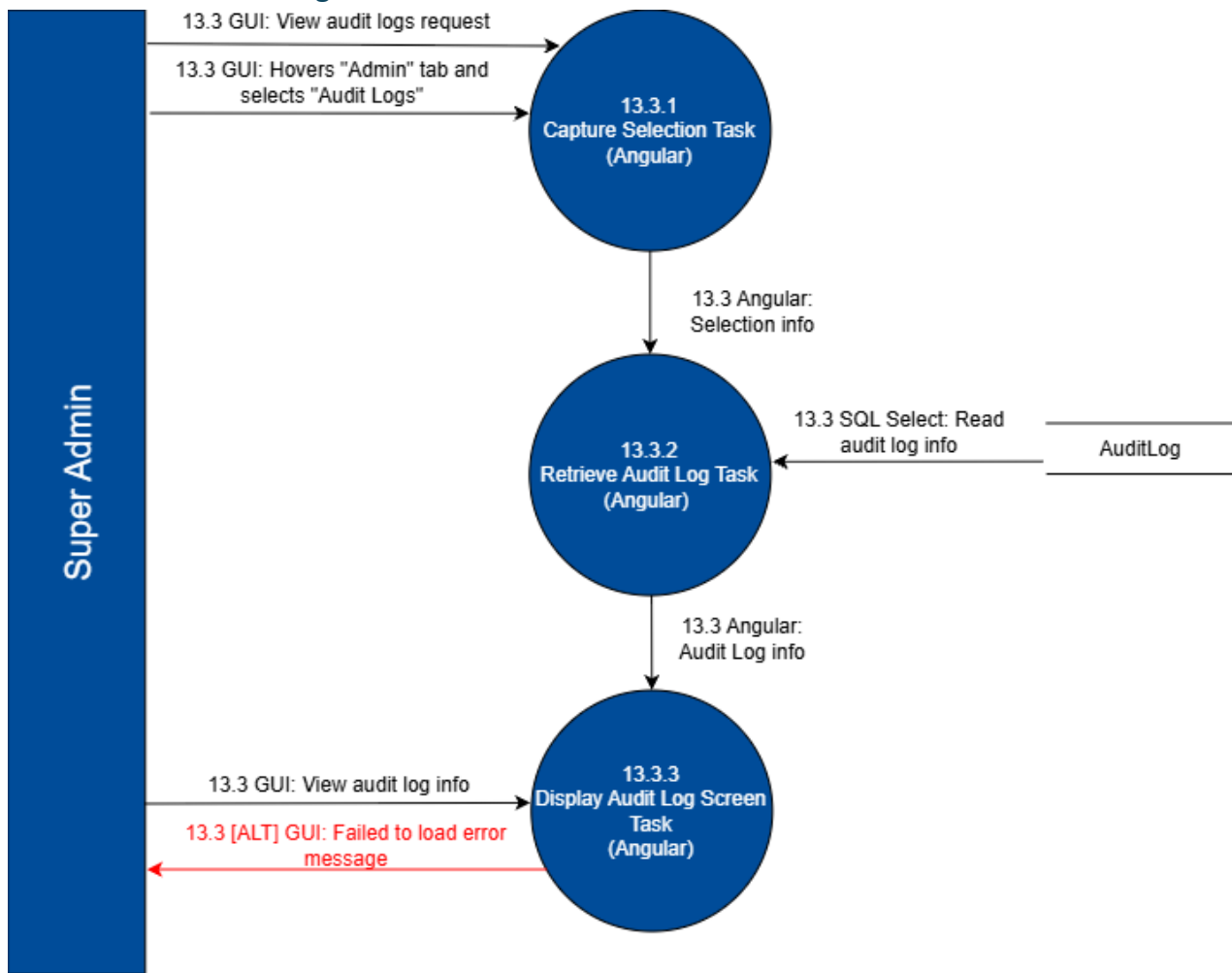


Figure 29: Use Case 13.3 View Audit Log Technical Primitive Diagram

2.7 Sequence Diagram

2.7.1 Project Management Subsystem

2.7.1.1 Add Photo Album

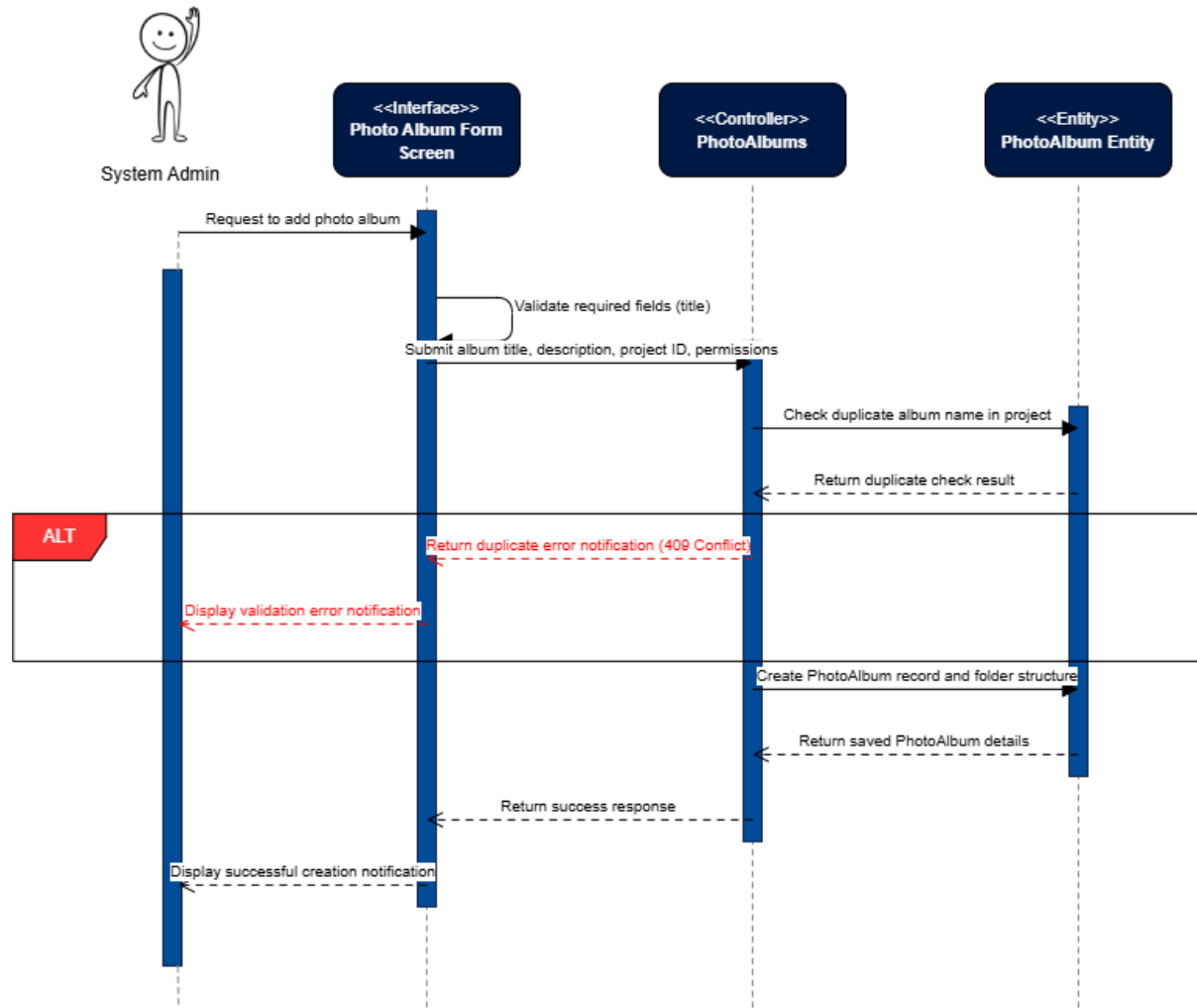


Figure 30: Use Case 7.25 Add Photo Album Sequence Diagram

2.7.1.2 View Photo Album

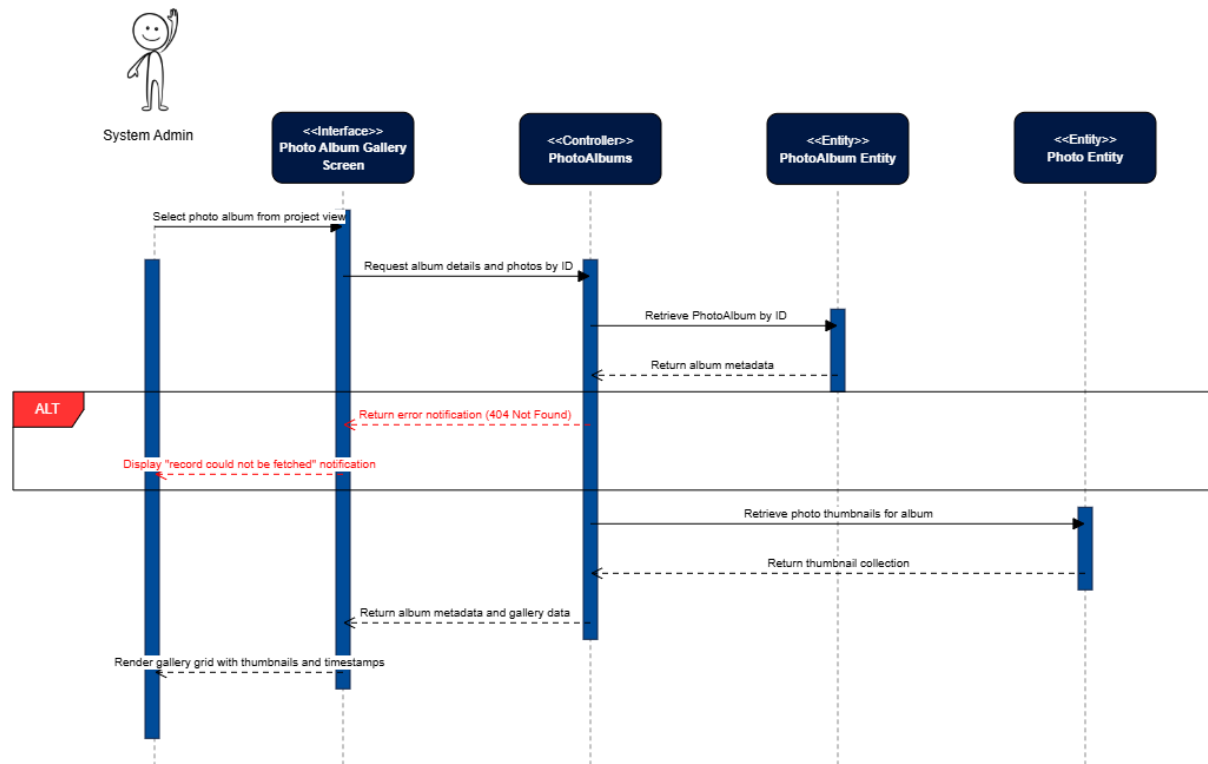


Figure 31: Use Case 7.26 View Photo Album Sequence Diagram

2.7.1.3 Update Photo Album

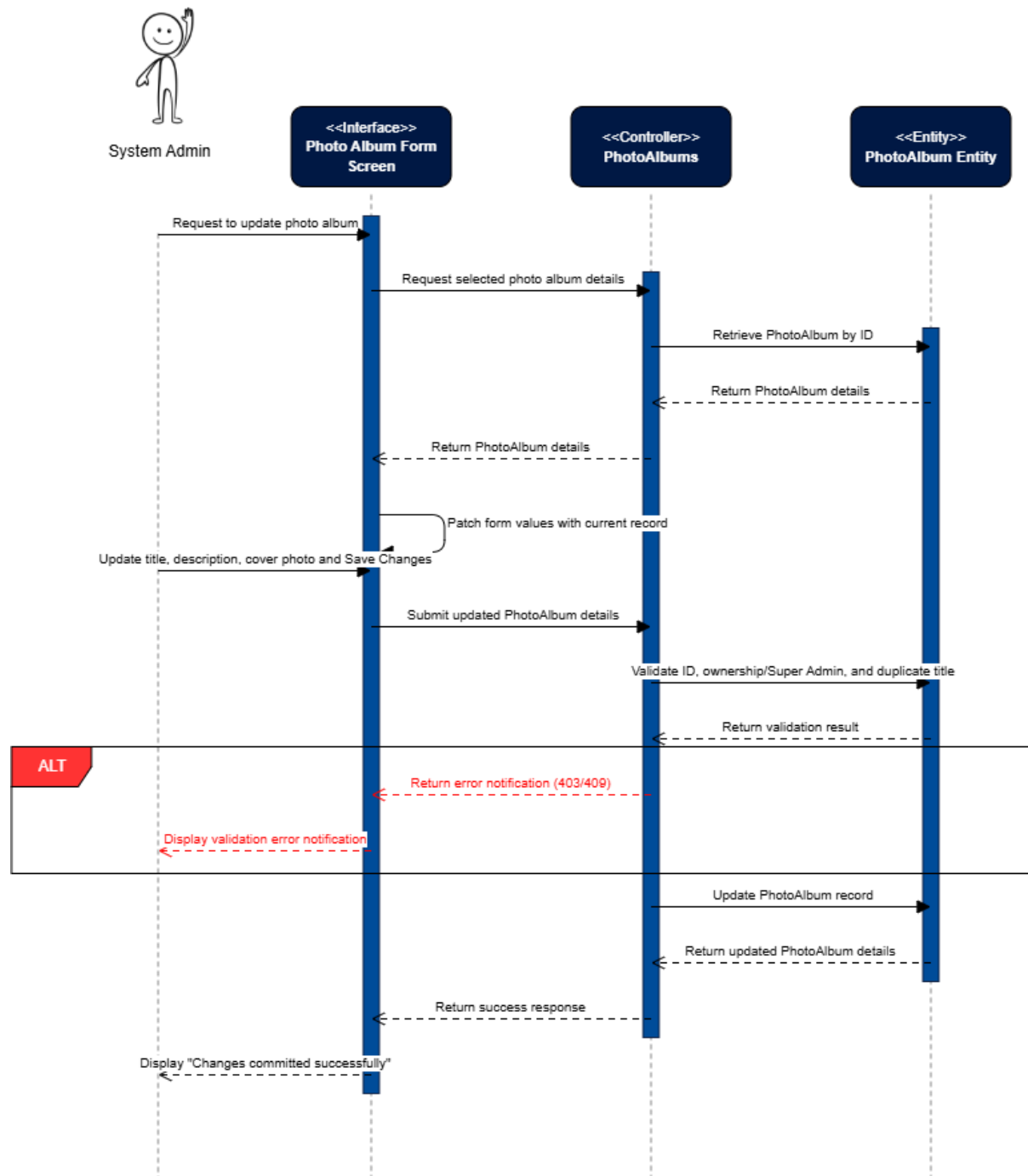


Figure 32: Use Case 7.27 Update Photo Album Sequence Diagram

2.7.1.4 Remove Photo Album

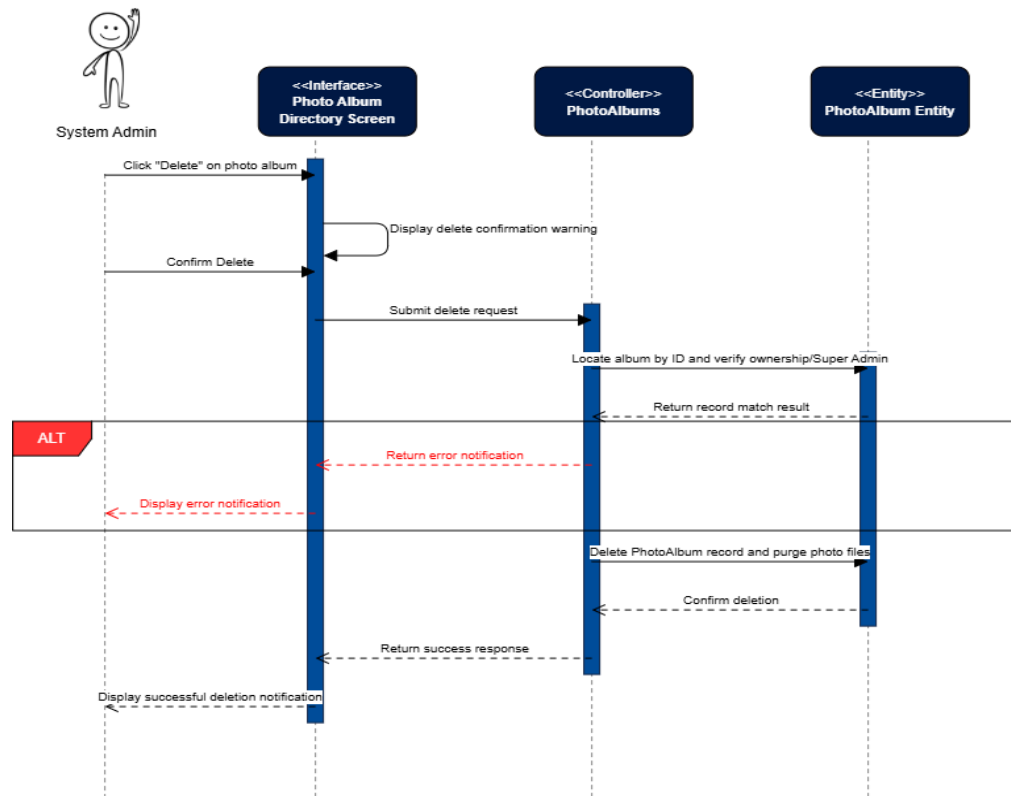


Figure 33: Use Case 7.28 Remove Photo Album Sequence Diagram

2.7.2 Report Subsystem

2.7.2.1 View Bike Impact Report

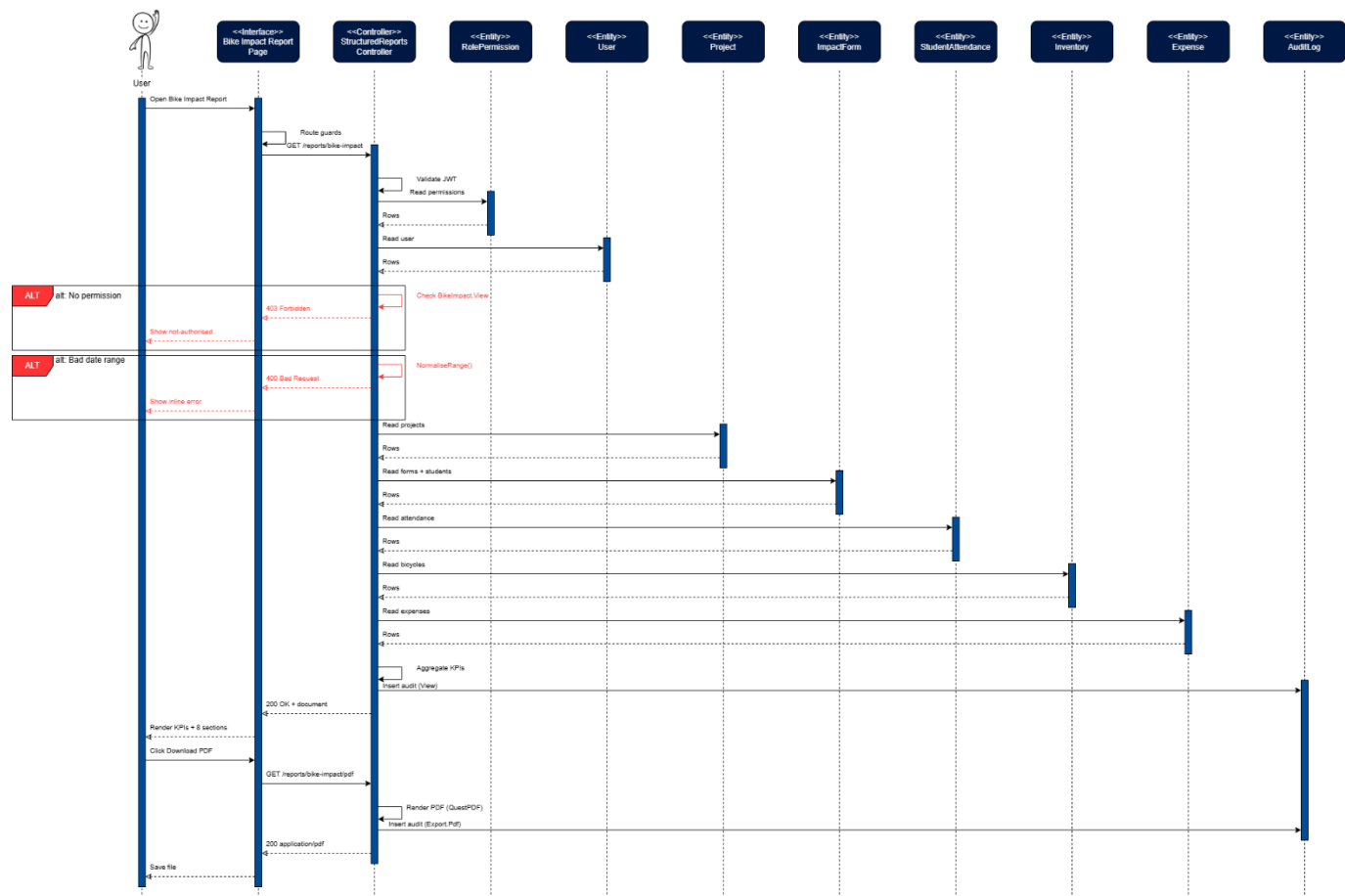


Figure 34: Use Case 12.5 View Bike Impact Report Sequence Diagram

2.7.2.2 View Budget Report

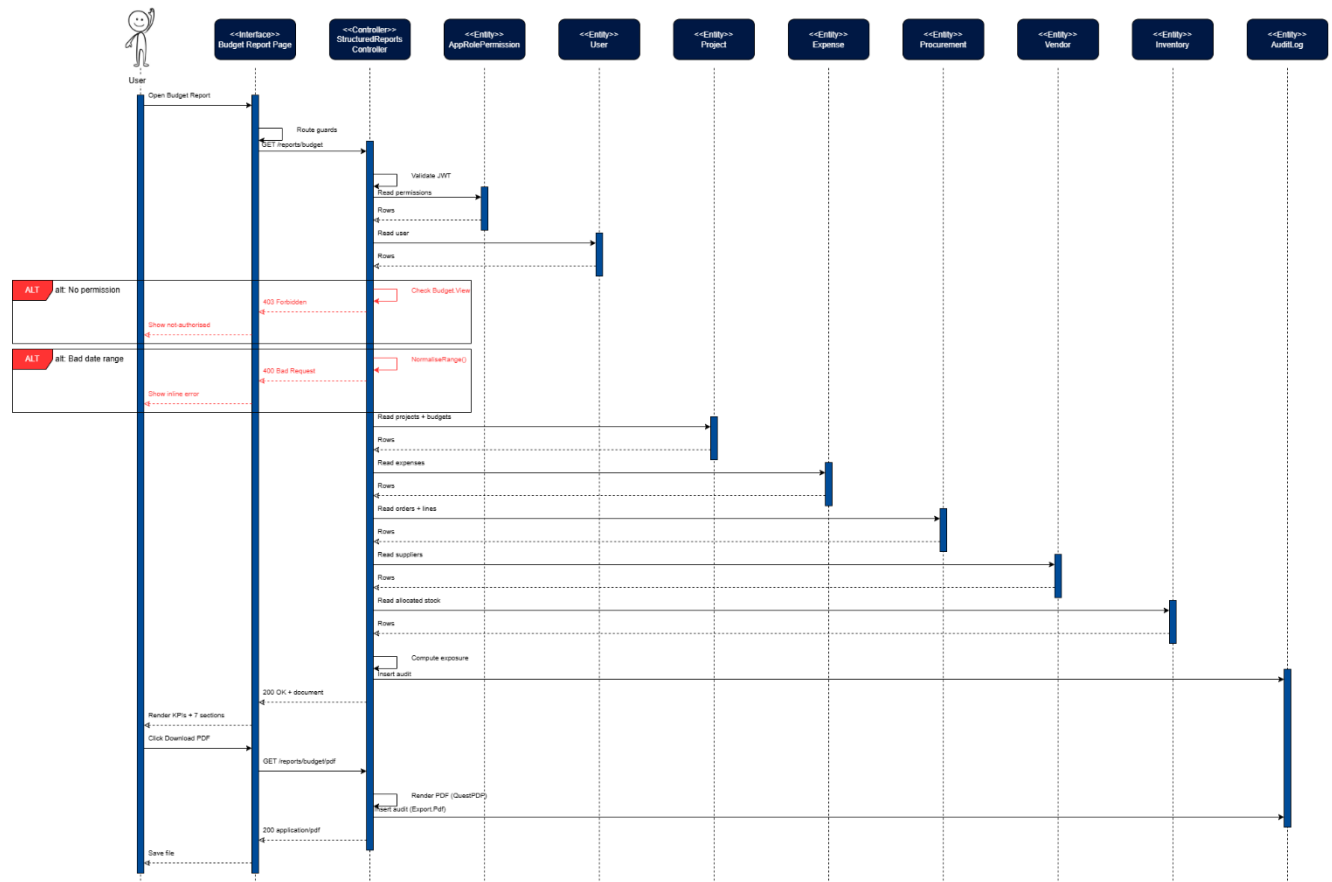


Figure 35: Use Case 12.6 View Budget Report Sequence Diagram

2.7.2.3 View Rhino Report

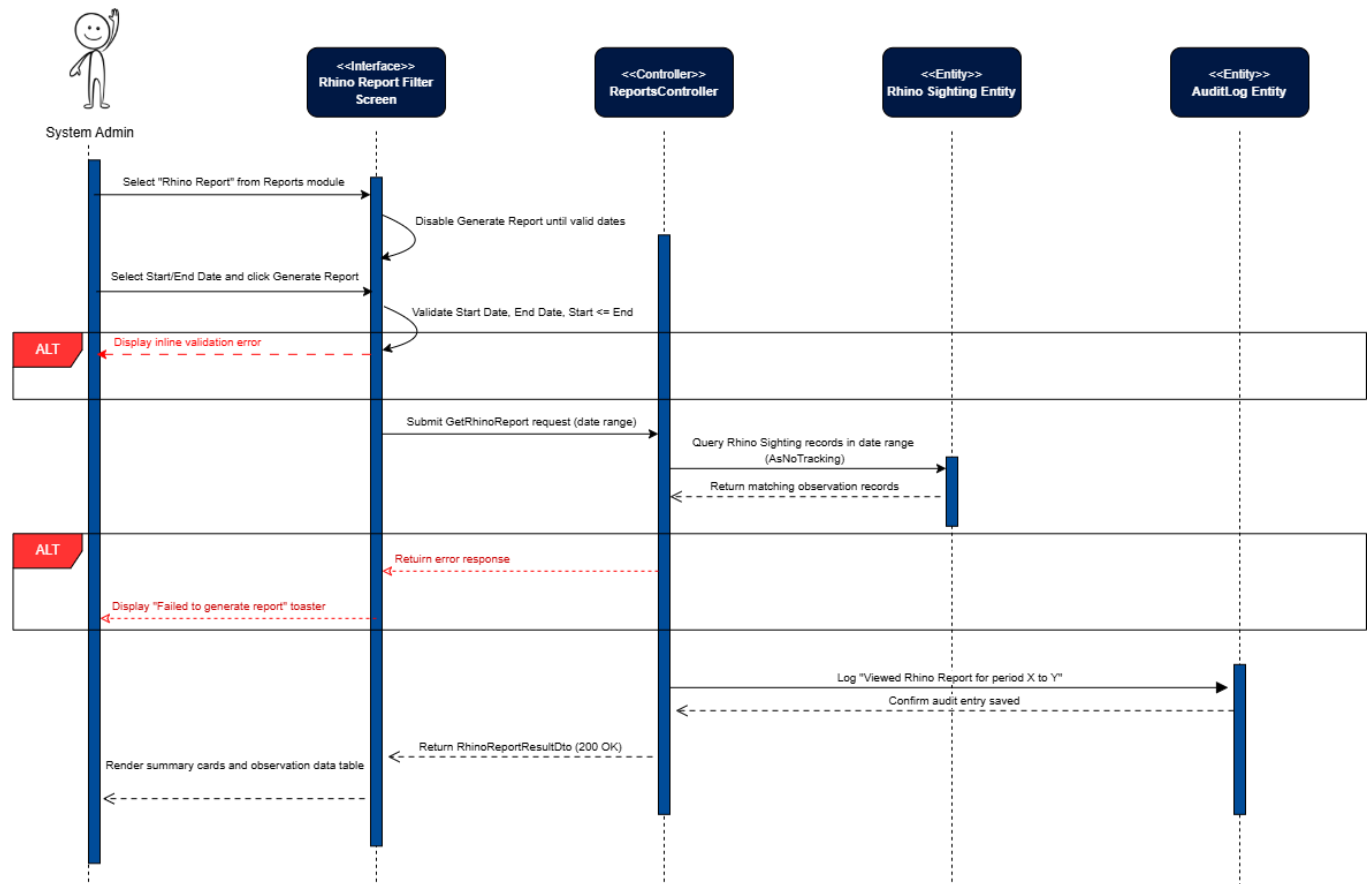


Figure 36: Use Case 12.7 View Rhino Report Sequence Diagram

2.7.2.4 View Elephant Report

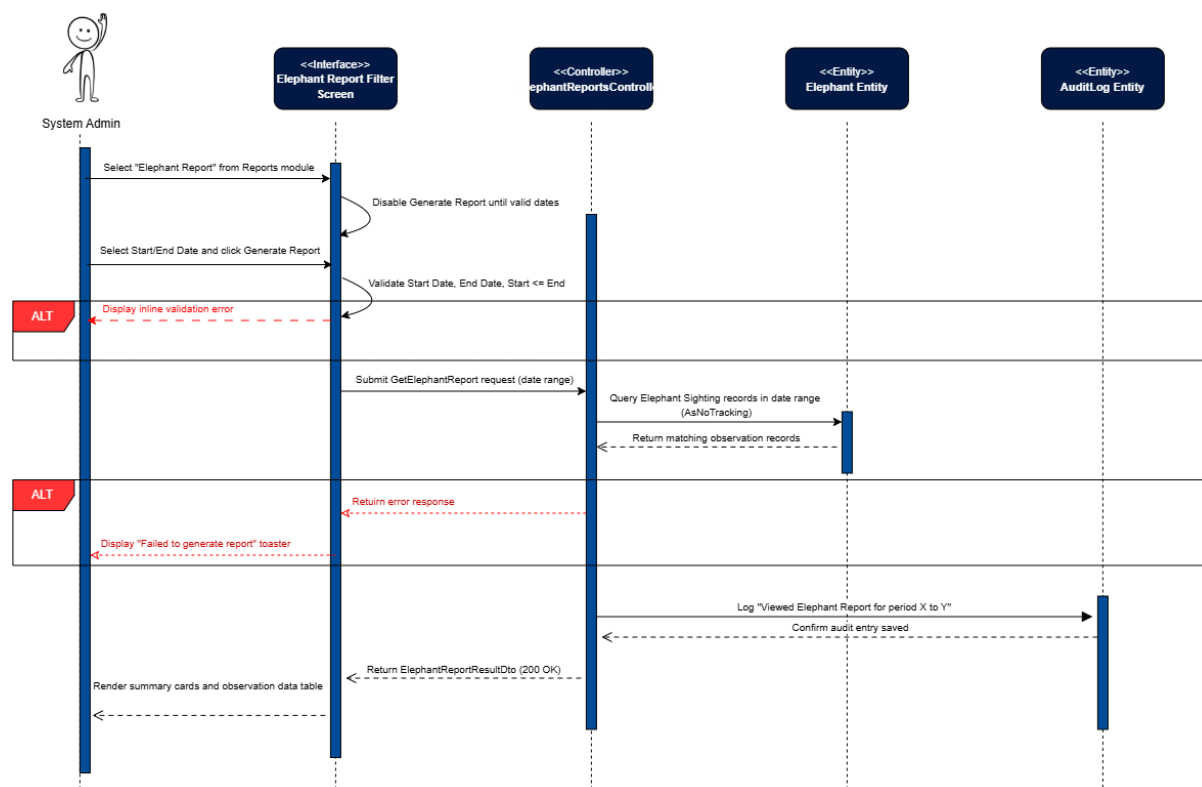


Figure 37: Use Case 12.8 View Elephant Report Sequence Diagram

[illegible]

2.7.2.5 View Machine Learning Dashboard

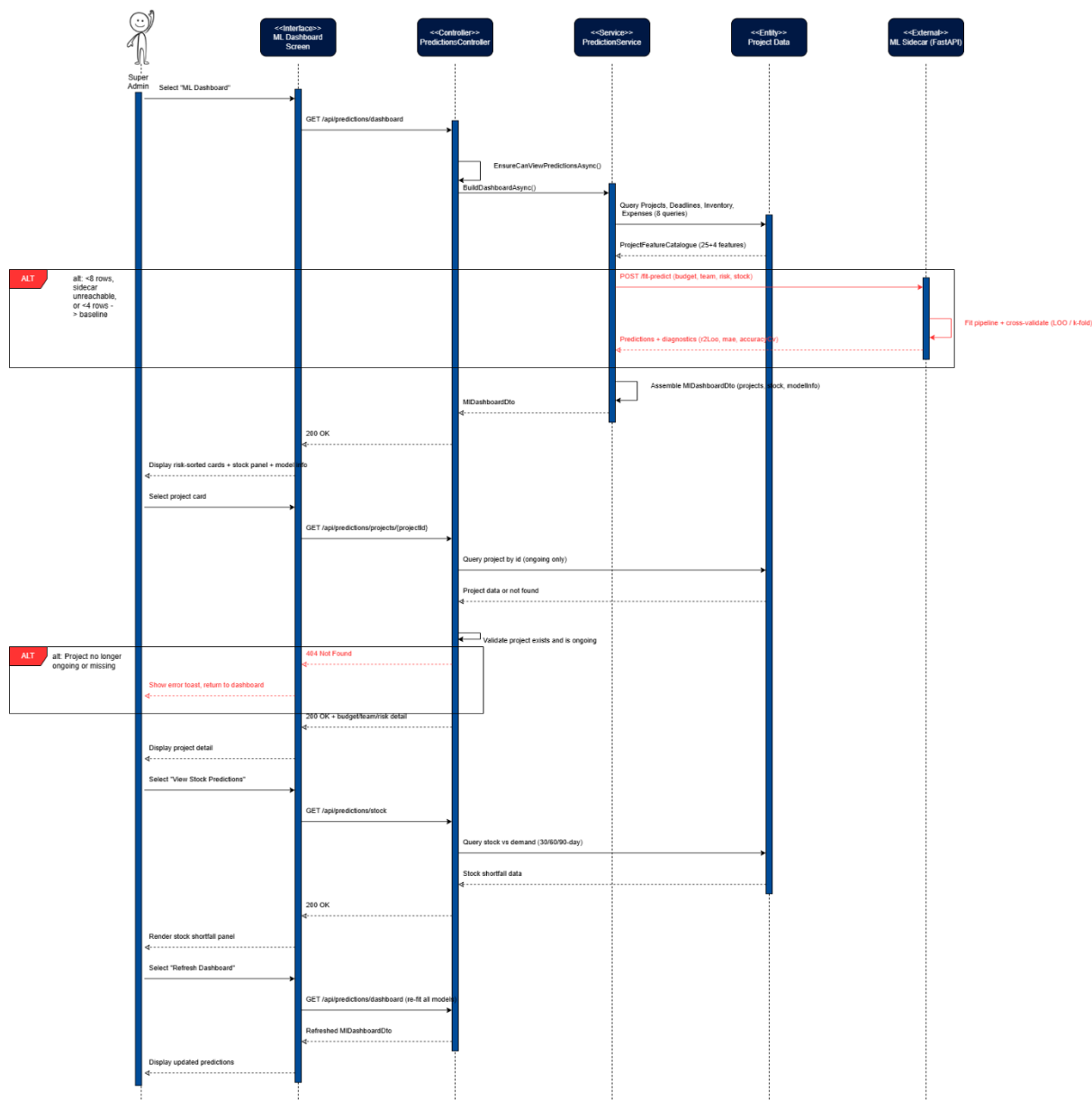


Figure 39: Use Case 12.10 View Machine Learning Dashboard Sequence Diagram

2.7.3 System Subsystem

2.7.3.1 Backup System

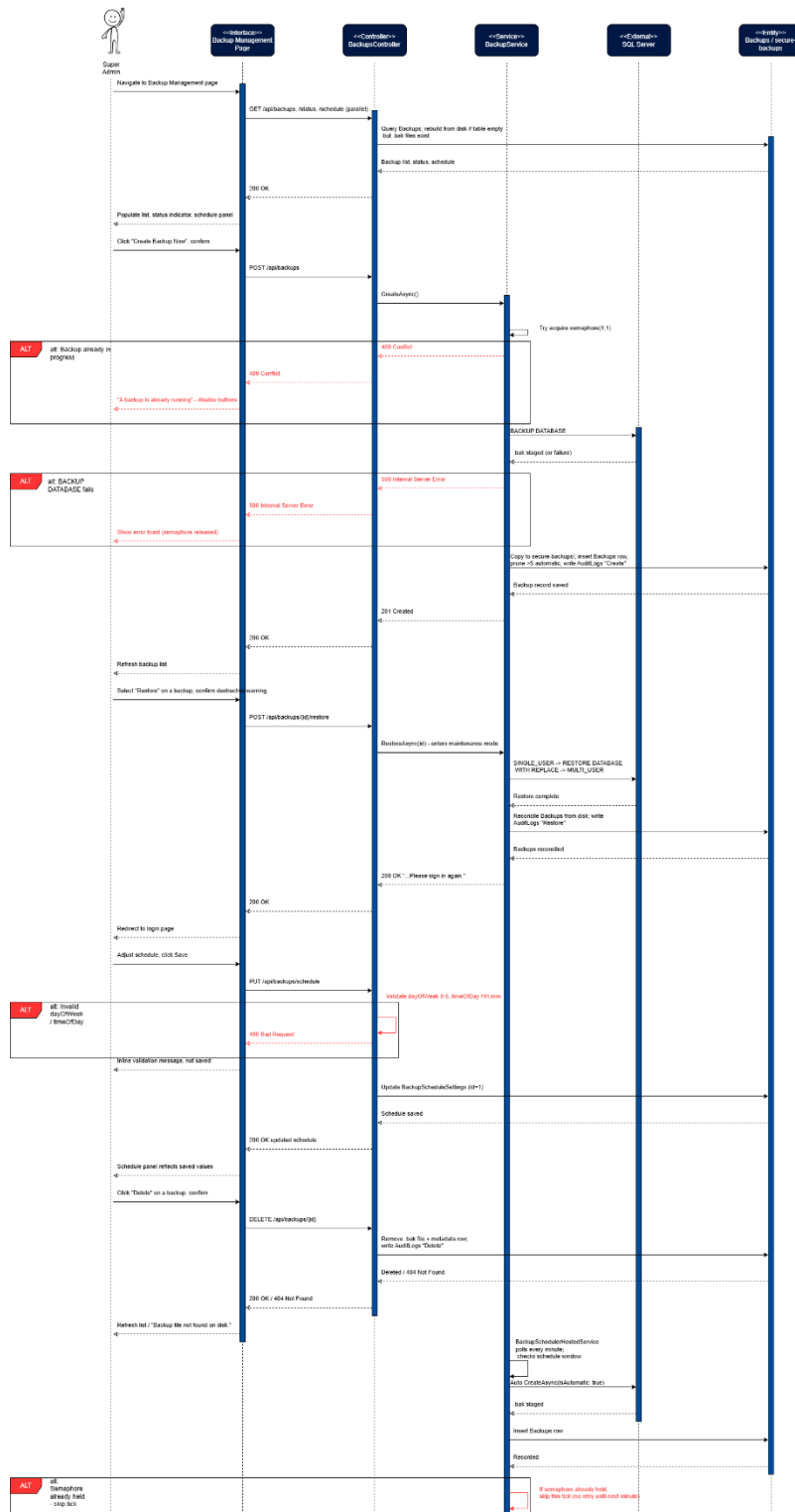


Figure 40: Use Case 13.1 Backup System Sequence Diagram

2.7.3.2 Restore System

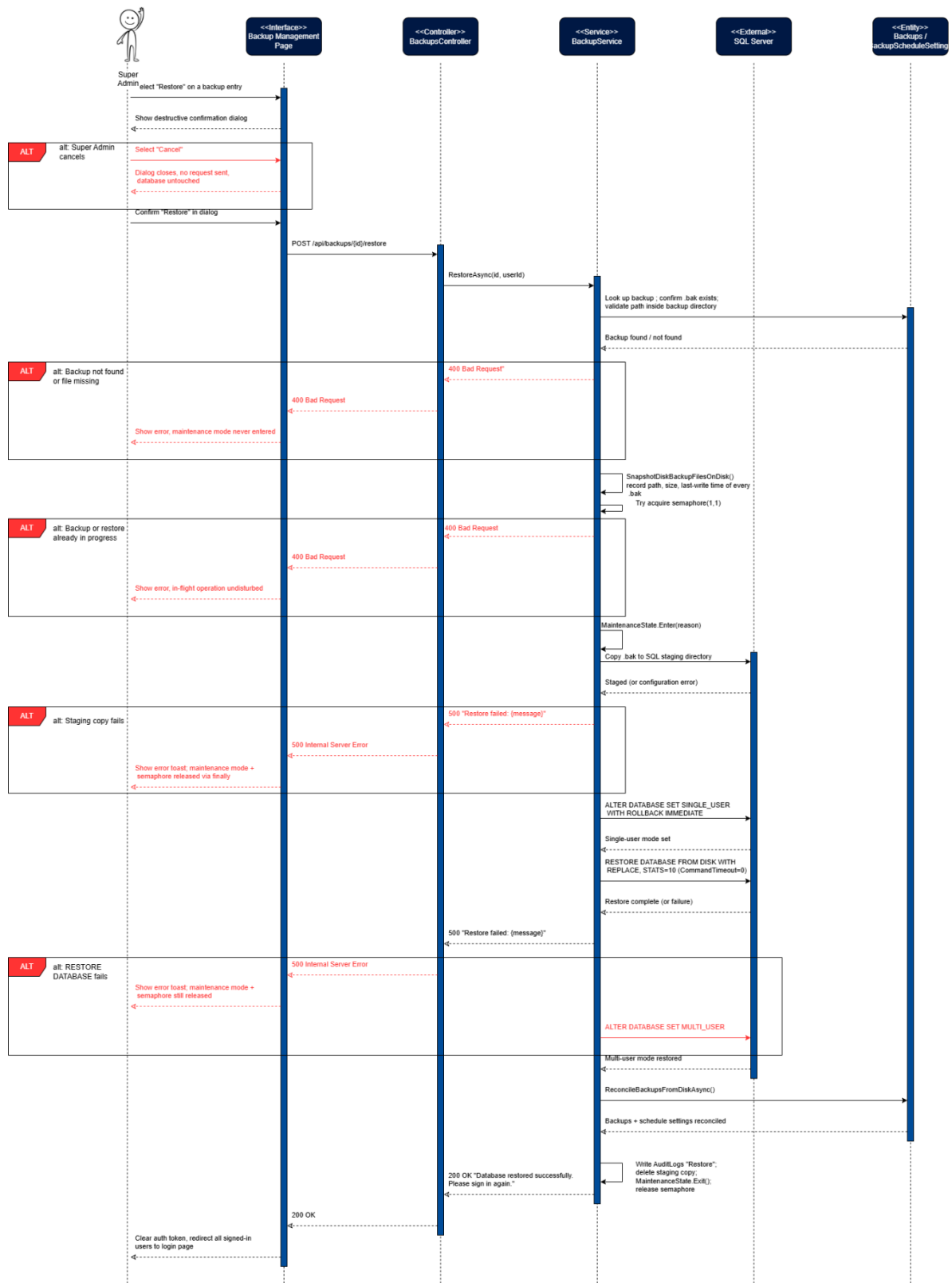


Figure 41: Use Case 13.2 Restore System Sequence Diagram

2.7.3.3 View Audit Log

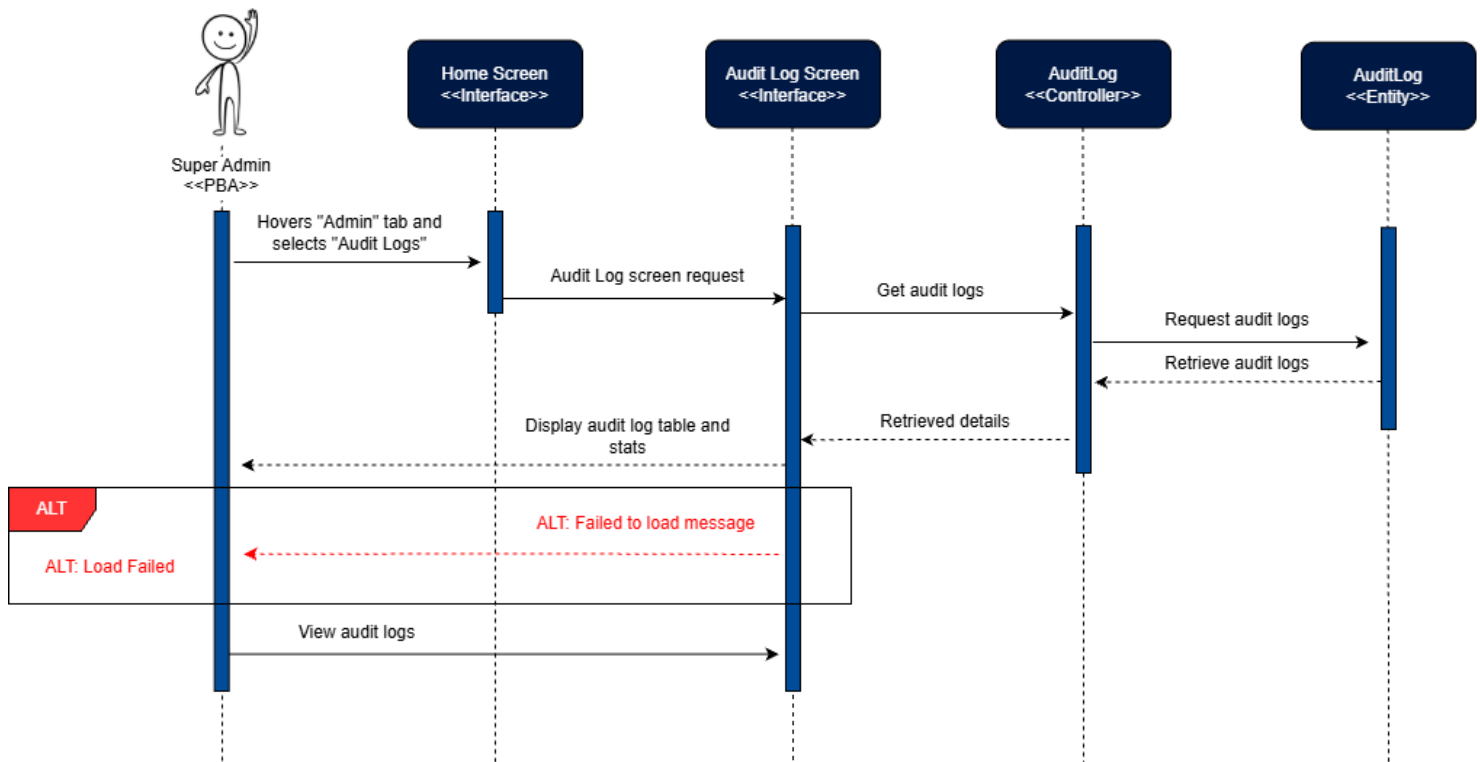


Figure 42: Use Case13.3 View Audit Log Sequence Diagram

2.8 Use Case Diagram

2.8.1 Project Management Subsystem

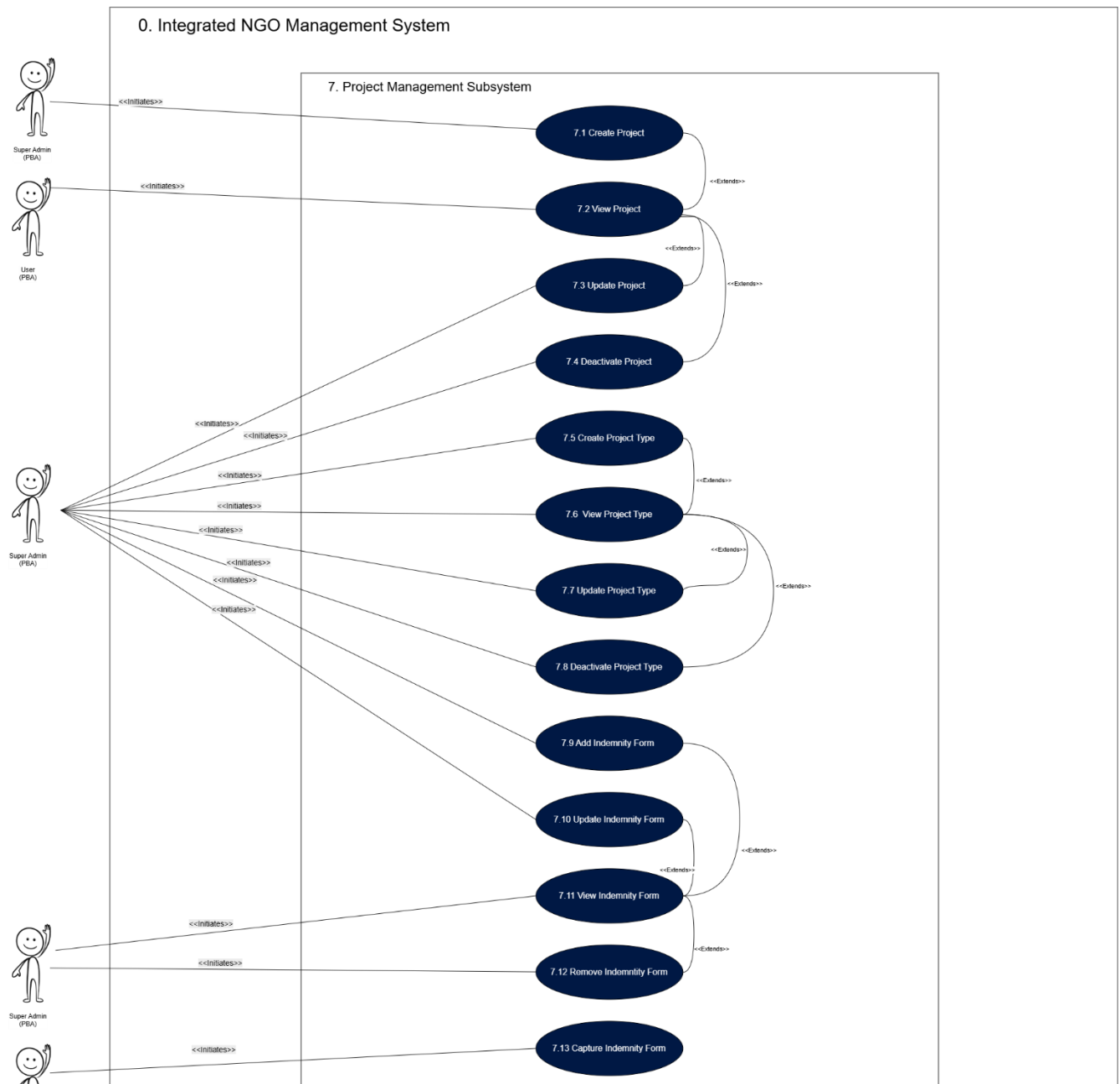


Figure 43: Project Management Subsystem Use Case Diagram

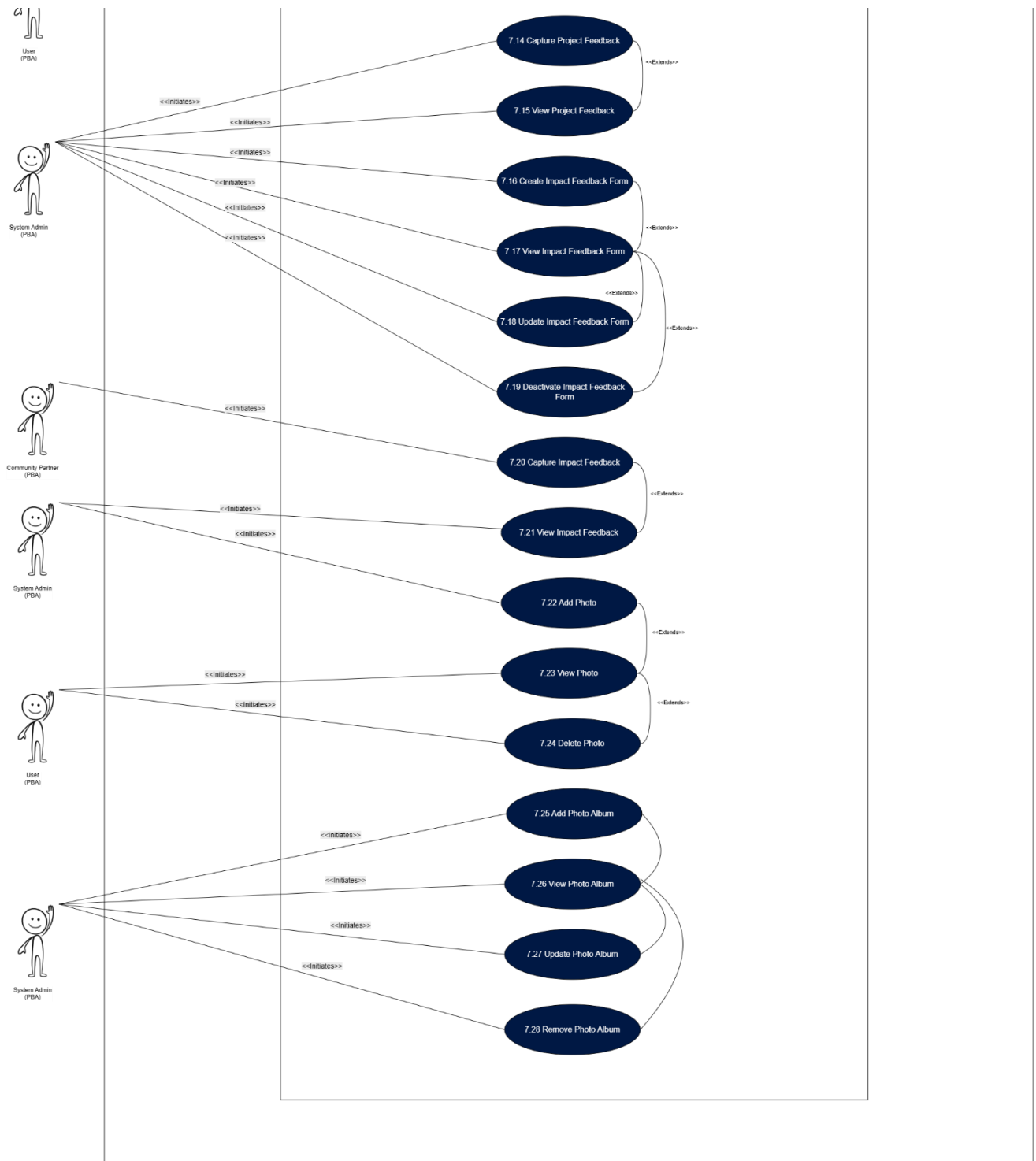


Figure 44: Project Management Subsystem Use Case Diagram (Continued)

2.8.2 Report Subsystem

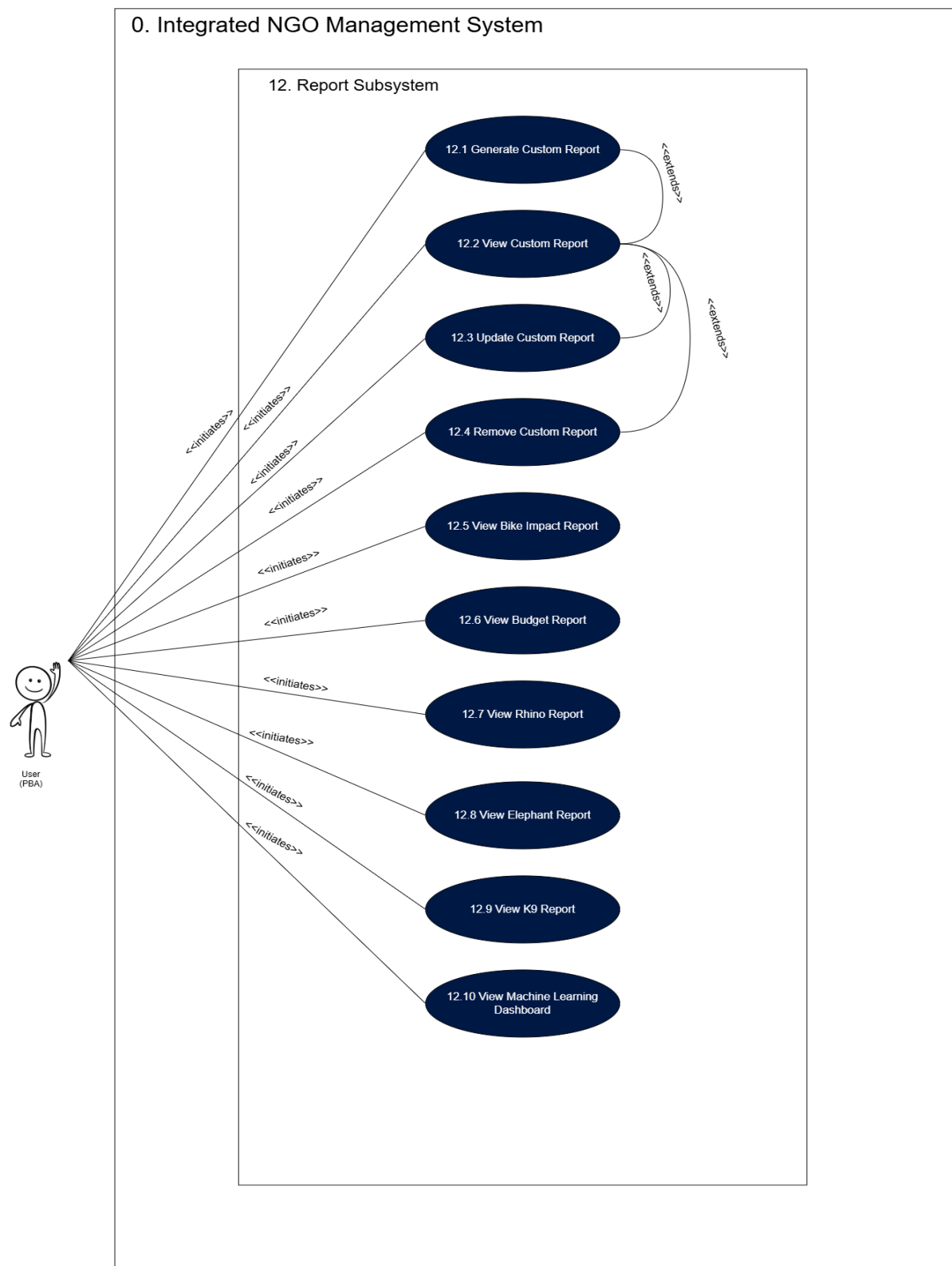


Figure 45: Report Subsystem Use Case Diagram

2.8.3 System Subsystem

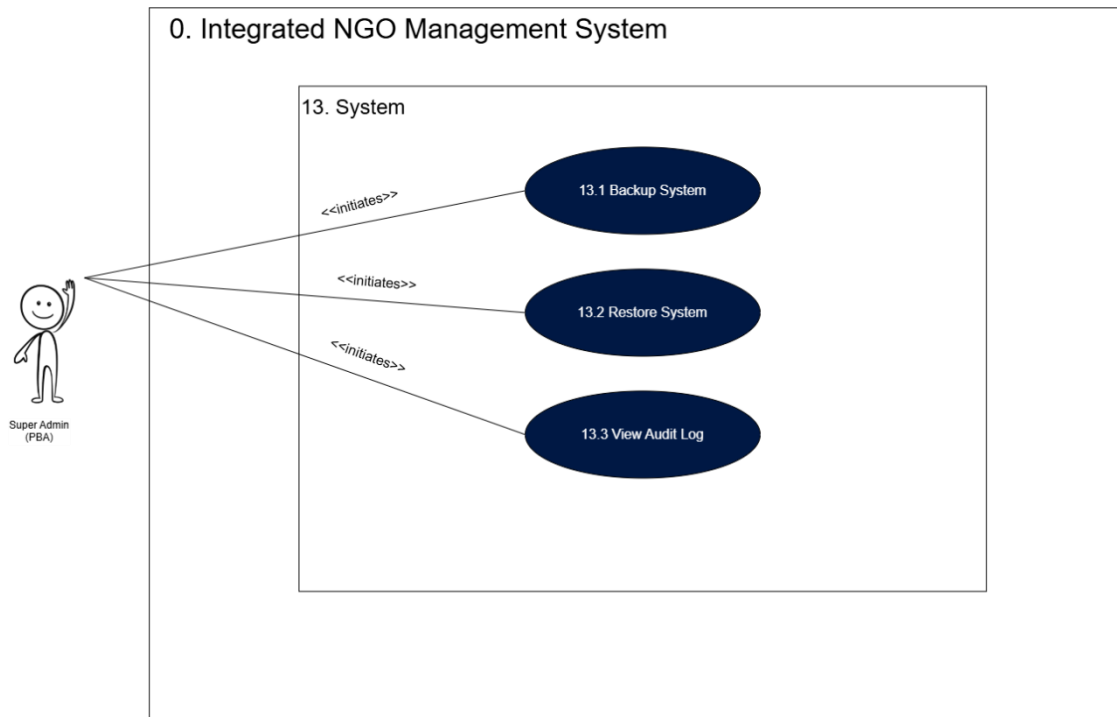


Figure 46: System Subsystem Use Case Diagram

Figure 47: Technical Entity Relationship Diagram

2.10. Updated Functional Requirements Lists

2.10.1 Introduction

This section contains a list, as well as descriptions of all functional requirements.

Tables with changes that have been made to the list are also included.

2.10.2 Updated Functional Requirements List

2.8.2.1 User Subsystem

2.8.2.1.1 User Login

2.8.2.1.2 User Logout

2.8.2.1.3 Reset Password

2.8.2.1.4 Create User

2.8.2.1.5 Update User

2.8.2.1.6 View User

2.8.2.1.7 Reactivate User

2.8.2.1.8 Deactivate User

2.8.2.1.9 View FAQ

2.8.2.2 Admin Subsystem

2.8.2.2.1 Create User Profile

2.8.2.2.2 Update User Profile

2.8.2.2.3 View User Profile

2.8.2.2.4 Deactivate User Profile

2.8.2.2.5 Add FAQ

2.8.2.2.6 Update FAQ

2.8.2.2.7 Remove FAQ

2.8.2.2.8 Add FAQ Type

2.8.2.2.9 Update FAQ Type

2.8.2.2.10 Remove FAQ Type

2.8.2.2.11 Create Role

2.8.2.2.12 View Role

2.8.2.2.13 Update Role

2.8.2.2.14 Reactivate Role

2.8.2.2.15 Deactivate Role

2.8.2.2.16 Update Role Permission

2.8.2.2.17 View Role Permission

2.8.2.3 Inventory Subsystem

2.8.2.3.1 Receive Inventory Item

2.8.2.3.2 View Inventory

2.8.2.3.3 Write-Off Inventory

2.8.2.3.4 Allocate Inventory

2.8.2.3.5 Create Inventory Item Type

2.8.2.3.7 View Inventory Item Type

2.8.2.3.7 View Inventory Item Types

2.8.2.3.8 Update Inventory Item Type

2.8.2.3.9 Deactivate Inventory Item Type

2.8.2.3.10 Reactivate Inventory Item Type

2.8.2.4 Group Subsystem

2.8.2.4.1 Create Group

2.8.2.4.2 View Groups

2.8.2.4.3 Update Group

2.8.2.4.4 Deactivate Group

2.8.2.4.5 Create Skill

2.8.2.4.6 View Skills

2.8.2.4.7 View Skill

2.8.2.4.8 Update Skill

2.8.2.4.9 Deactivate Skill

2.8.2.5 Warehouse Subsystem

2.8.2.5.1 Create Warehouse

2.8.2.5.2 View Warehouse

2.8.2.5.3 Update Warehouse

2.8.2.5.4 Reactivate Warehouse

2.8.2.5.5 Deactivate Warehouse

2.8.2.5.6 Inventory Stock Take

2.8.2.6 Vendor Subsystem

2.8.2.6.1 Create Vendor

2.8.2.6.3 View Vendor

2.8.2.6.3 View Vendors

2.8.2.6.4 Edit Vendor

2.8.2.6.5 Reactivate Vendor

2.8.2.6.6 Deactivate Vendor

2.8.2.6.7 Create Vendor Type

2.8.2.6.8 Update Vendor Type

2.8.2.6.9 View Vendor Types

2.8.2.6.10 Reactivate Vendor Type

2.8.2.6.11 Deactivate Vendor Type

2.8.2.7 Project Management Subsystem

2.8.2.7.2 Create Project

2.8.2.7.2 View Project

2.8.2.7.3 Update Project

2.8.2.7.4 Deactivate Project

2.8.2.7.5 Create Project Type

2.8.2.7.6 View Project Type

2.8.2.7.7 Update Project Type

2.8.2.7.8 Deactivate Project Type

2.8.2.7.9 Add Indemnity Form

2.8.2.7.20 Update Indemnity Form

2.8.2.7.21 View Indemnity Form

2.8.2.7.22 Remove Indemnity Form

2.8.2.7.23 Capture Indemnity Form

2.8.2.7.24 Capture Project Feedback

2.8.2.7.25 View Project Feedback

2.8.2.7.26 Create Impact Feedback Form

2.8.2.7.27 View Impact Feedback Form

2.8.2.7.28 Update Impact Feedback Form

2.8.2.7.29 Deactivate Impact Feedback Form

2.8.2.7.20 Capture Impact Form

2.8.2.7.21 View Impact Form

2.8.2.7.22 Add Photo

2.8.2.7.23 View Photo

2.8.2.7.24 Delete Photo

2.8.2.7.25 Add Photo Album

2.8.2.7.26 View Photo Album

2.8.2.7.27 Update Photo Album

2.8.2.7.28 Remove Photo Album

2.8.2.8 K9 Subsystem

2.8.2.8.1 Add Dog

2.8.2.8.2 View Dog

2.8.2.8.3 View Dogs

2.8.2.8.4 Update Dog

2.8.2.8.5 Deactivate Dog

2.8.2.8.6 Assign User to Dog

2.8.2.8.7 View User-Dog Assignment

2.8.2.8.8 Update User-Dog Assignment

2.8.2.8.9 Reactivate User-Dog Assignment

2.8.2.8.10 Deactivate User-Dog Assignment

2.8.2.8.11 Add K9 Role

2.8.2.8.12 Update K9 Role

2.8.2.8.13 View K9 Roles

2.8.2.8.14 Reactivate K9 Role

2.8.2.8.15 Deactivate K9 Role

2.8.2.8.16 View K9 Ancestry

2.8.2.9 Rhino Subsystem

2.8.2.9.1 Add Rhino

2.8.2.9.2 View Rhino

2.8.2.9.3 View Rhinos

2.8.2.9.4 Update Rhino

2.8.2.9.5 Deactivate Rhino

2.8.2.10 Elephant Subsystem

2.8.2.10.1 Add Elephant

2.8.2.10.2 View Elephant

2.8.2.10.3 View Elephants

2.8.2.10.4 Update Elephant

2.8.2.10.5 Deactivate Elephant

2.8.2.11 Message Subsystem

2.8.2.11.1 Create Message

2.8.2.11.2 View Messages

2.8.2.12 Report Subsystem

2.8.2.12.1 Generate Custom Report

2.8.2.12.2 View Custom Report

2.8.2.12.3 Update Custom Report

2.8.2.12.4 Remove Custom Report

2.8.2.12.5 View Bike Impact Report

2.8.2.12.6 View Budget Report

2.8.2.12.7 View Rhino Report

2.8.2.12.8 View Elephant Report

2.8.2.12.9 View K9 Report

2.8.2.12.10 View Machine Learning Dashboard

2.8.2.13. System Subsystem

2.8.2.13.1 Backup System

2.8.2.13.2 Restore System

2.8.2.13.3 View Audit Log

2.8.2.14 Budget Subsystem

2.8.2.14.1 View Project Budget

2.8.2.14.2 View Project Budgets

2.8.2.14.3 Log An Expense

2.8.2.14.4 Update An Expense

2.8.2.14.5 View Expense

2.8.2.14.6 Remove Expense

2.10.3 Updated Functional Requirements Lists Descriptions

2.10.3.1 User Subsystem

Functional Requirement	Explanation
Requirement number:	1.1
Requirement name (use case name):	User Login
Requirement short description:	A User logs into the system using their email address and password.
Requirement detailed description and constraints:	A user logs in to the system using their email address and password. The system compares the email address with the associated hashed password in the database; upon successful authentication the user is directed to the appropriate dashboard based on their role and associated permission level. Constraints: • The user should already have created an account on the system. If not, they must sign up. • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Secure Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 14: 1.1 User Login – Description

Functional Requirement	Explanation
Requirement number:	1.2
Requirement name (use case name):	User Logout
Requirement short description:	A user can log out of the system.
Requirement detailed description and constraints:	A user can log out of the system, thus terminating their session, ensuring that no one with physical access to their device can access the system. Once they have completed this action, they are directed to the Login page. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Secure Access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 15: 1.2 User Logout Description

Functional Requirement	Explanation
Requirement number:	1.3
Requirement name (use case name):	Reset Password
Requirement short description:	A user could reset their password if they either forgot it or suspect that their credentials might have leaked on another service.
Requirement detailed description and constraints:	A user could reset their password if they either forgot it or suspect that their credentials have leaked on another service. As the system does not save passwords in plain text, but rather a hashed format, the user will be prompted to enter their email address, to which a reset link will be sent. Once they have clicked the link, they have access to a field where they can enter a new password. The password hash is then saved in the database, replacing the previous version. Once these actions have been completed, the user will be redirected to the Login page. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature. • The password cannot be the same as the previous password, if this edge case occurs, the system will notify that the passwords are the same. • The user should already have created an account on the system. If not, they must sign up.
Business rules applicable to this requirement	• Secure Access. • Ability to recover Account.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 16: 1.3 Reset Password – Description

Functional Requirement	Explanation
Requirement number:	1.4
Requirement name (use case name):	Create User
Requirement short description:	A person must be able to create a user profile by signing up to the system.
Requirement detailed description and constraints:	A person must be able to create a user profile by signing up to the system. This entails providing login credentials (email address/password), basic personal information such as gender and age, location and selecting skills they possess. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature. • The new user must have a valid email address.
Business rules applicable to this requirement	• Secure Access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 17: 1.4 Create User – Description

Functional Requirement	Explanation
Requirement number:	1.5
Requirement name (use case name):	Update User
Requirement short description:	A user must be able to edit their personal user profile.
Requirement detailed description and constraints:	A user must be able to edit their personal user profile. This could entail changing details like location or the skills they possess. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature. • The user cannot change their age, as this is automatically updated by the system.
Business rules applicable to this requirement	• Secure Access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 18: 1.5 Update User – Description

Functional Requirement	Explanation
Requirement number:	1.6
Requirement name (use case name):	View User
Requirement short description:	A user must be able to view their personal user profile.
Requirement detailed description and constraints:	A user must be able to edit their personal user profile. Administrators should have the ability to view the personal profiles of all users This could entail changing details like location or the skills they possess. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature. • Only Administrators can view the full profile details of other users.
Business rules applicable to this requirement	• Secure Access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 19: 1.6 View User – Description

Functional Requirement	Explanation
Requirement number:	1.7
Requirement name (use case name):	Reactivate User
Requirement short description:	An administrator must be able to reactivate an account.
Requirement detailed description and constraints:	An administrator should be able to reactivate any deactivated user accounts. Once an account has been reactivated, a user can login to the system using their credentials and will appear on the list of active users again. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Secure Access. • Data Retention.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 20: 1.7 Reactivate User - Description

Functional Requirement	Explanation
Requirement number:	1.8
Requirement name (use case name):	Deactivate User
Requirement short description:	A user must be able to deactivate their personal user profile.
Requirement detailed description and constraints:	A user must be able to deactivate their personal user profile. Once they have selected the option to deactivate their profile, they will be asked to confirm. If they confirm, the profile will be deactivated, and they will be directed to the login page. Data regarding their involvement in projects will be kept for accurate records on historical data, but they will no longer appear on the users list. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Secure Access. • Data Retention.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 21: 1.8 Deactivate User - Description

Functional Requirement	Explanation
Requirement number:	1.9
Requirement name (use case name):	View FAQ
Requirement short description:	A user must be able to click a help button that takes them to a context aware FAQ page.
Requirement detailed description and constraints:	A user must be able to click a help button that takes them to a context aware FAQ page. If the button is pressed on the home page, it will direct the user to the full FAQ document, but if it is pressed while using a subsystem, it will direct the user to the appropriate section as set up by the admin using the FAQ Type use cases. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Access to a help function.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 22: 1.9 View FAQ – Description

2.10.3.2 Admin Subsystem

Functional Requirement	Explanation
Requirement number:	2.1
Requirement name (use case name):	Create User Profile
Requirement short description:	A Super Admin must be able to create a new user profile in the system.
Requirement detailed description and constraints:	Super Admin can input necessary details to create a new user. The system validates entries before asking for confirmation. Once confirmed, the user profile is saved. Constraints: • Internet connection required. • Mandatory fields must be completed before submission.
Business rules applicable to this requirement	• Only the Super Admin can create user profiles.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 23: 2.1 Create User Profile - Description

Functional Requirement	Explanation
Requirement number:	2.2
Requirement name (use case name):	Update User Profile
Requirement short description:	A Super Admin must be able to update existing user profiles.
Requirement detailed description and constraints:	The Super Admin can modify user information. System validates changes before confirmation. Once confirmed, updates are applied. Constraints: • Internet connection required. • Cannot change another user's password without proper authorization.
Business rules applicable to this requirement	• Only the Super Admin can create user profiles.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 24: 2.2 Update User Profile - Description

Functional Requirement	Explanation
Requirement number:	2.3
Requirement name (use case name):	View User Profile
Requirement short description:	The Super Admin must be able to view user profiles.
Requirement detailed description and constraints:	The Super Admin can search and view details of existing users. No changes are allowed in this view. Constraints: • Internet connection required.
Business rules applicable to this requirement	• Only the Super Admin can view any profiles.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 25: 2.3 View User Profile - Description

Functional Requirement	Explanation
Requirement number:	2.4
Requirement name (use case name):	Deactivate User Profile
Requirement short description:	The Super Admin can deactivate user profiles.
Requirement detailed description and constraints:	Super Admin can disable a profile temporarily. The system confirms the action before applying it. Constraints: • Internet connection required. • Action must be logged in the system audit trail.
Business rules applicable to this requirement	• Only the Super Admin can deactivate profiles.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 26: 2.4 Deactivate User Profile - Description

Functional Requirement	Explanation
Requirement number:	2.5
Requirement name (use case name):	Add FAQ
Requirement short description:	The Super Admin can add new FAQs to the system.
Requirement detailed description and constraints:	The Super Admin input question and answer details. System validates before confirmation. Once confirmed, FAQ is added. Constraints: • Internet connection required.
Business rules applicable to this requirement	• Only the Super Admin can add FAQs.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 27: 2.5 Add FAQ – Description

Functional Requirement	Explanation
Requirement number:	2.6
Requirement name (use case name):	Update FAQ
Requirement short description:	The Super Admin can update existing FAQs.
Requirement detailed description and constraints:	The Super Admin can edit FAQ content, which is validated before confirmation. Once confirmed, updates are applied. Constraints: • Internet connection required.
Business rules applicable to this requirement	• Only the Super Admin can update FAQs.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 28: 2.6 Update FAQ – Description

Functional Requirement	Explanation
Requirement number:	2.7
Requirement name (use case name):	Remove FAQ
Requirement short description:	The Super Admin can remove FAQs from the system.
Requirement detailed description and constraints:	The Super Admin can select FAQ to delete. System asks for confirmation before removal. Constraints: • Internet connection required.
Business rules applicable to this requirement	• Only the Super Admin can remove FAQs from the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 29: 2.7 Remove FAQ – Description

Functional Requirement	Explanation
Requirement number:	2.8
Requirement name (use case name):	Add FAQ Type
Requirement short description:	A Super Admin must be able to capture and commit a new FAQ classification category label to the global lookup directory.
Requirement detailed description and constraints:	The system must provide an interactive Angular Reactive Form template inside an overlay modal layout. The workspace operator must be allowed to input a structural configuration name string, declare a comprehensive functional description scope, and assign an active boolean data visibility property state. Constraints: • Time Constraint: The technical pipeline implementation and API route deployment must be completed and audited before the Project Day in October 2026. • User Skill Constraint: The user interface components must utilize responsive, clean, and intuitive validation feedback indicators to accommodate administrative users with minimal systemic training. • Network Constraint: The application client requires a persistent HTTPS web data connection stream to dispatch payloads to host environments. • Entity Constraint: Alphanumeric parameters mapping to the TypeName input boundary must strictly not exceed a data allocation size ceiling threshold of 100 characters.
Business rules applicable to this requirement	• Secure Access. • Classification Domain Uniqueness
Revision date and Revision number:	2026-06-24 Version 3.0
Criticality/Priority:	High

Table 30: 2.8 Add FAQ Type – Description

Functional Requirement	Explanation
Requirement number:	2.9
Requirement name (use case name):	Update FAQ Type
Requirement short description:	A Super Admin must be able to alter the alphanumeric properties or status configurations of an existing FAQ classification type entity.
Requirement detailed description and constraints:	The system must launch a contextual edit layout container window and perform an automated patchValue routine to inject current relational data properties into the active form controls. The interface must trap mutations made to the type name or scope variables and package them into an asynchronous HTTP PUT data stream. Constraints: • Time Constraint: The technical pipeline implementation and API route deployment must be completed and audited before the Project Day in October 2026. • Operational Constraint: Form modifications must process asynchronously through specialized data transfer objects (DTOs) to insulate database table layouts from the network stream layer. • Network Constraint: The client framework requires a persistent internet stream to hit the targeted server router path (PUT api/FaqType/update/{id}).
Business rules applicable to this requirement	• Secure Access. • Data Overwrite Consistency
Revision date and Revision number:	2026-06-24 Version 3.0
Criticality/Priority:	High

Table 31: 2.9 Update FAQ Type – Description

Functional Requirement	Explanation
Requirement number:	2.10
Requirement name (use case name):	Remove FAQ Type
Requirement short description:	A Super Admin must be able to remove an FAQ classification type category row from the system database when it is no longer required.
Requirement detailed description and constraints:	The frontend application must intercept row-level deletion clicks and display an explicit modal verification confirmation overlay block. Upon user affirmation, the framework must compile an asynchronous HTTP DELETE transaction addressing the primary key parameter sequence route (DELETE api/FaqType/remove/{id}). The backend must validate database dependency maps prior to dropping rows from disk storage files. Constraints: • Time Constraint: The technical pipeline implementation and API route deployment must be completed and audited before the Project Day in October 2026. • User Error Protection Constraint: Destructive command routines must never process silent; an explicit user-facing verification warning prompt layout component must always disrupt execution loops. • Network Constraint: The system requires stable network connectivity to safely process transaction signatures without corrupting indexing tables.
Business rules applicable to this requirement	• Secure Access.
Revision date and Revision number:	2026-06-24 Version 3.0
Criticality/Priority:	High

Table 32: 2.10 Remove FAQ Type – Description

Functional Requirement	Explanation
Requirement number:	2.11
Requirement name (use case name):	Create Role
Requirement short description:	The Super Admin can create new roles in the system.
Requirement detailed description and constraints:	The Super Admin enter role name and permissions. System validates input before confirmation. Once confirmed, role is created. Constraints: • Internet connection required.
Business rules applicable to this requirement	• The Super Admin can create roles.
Revision date and Revision number:	2026-06-24 Version 4.0
Criticality/Priority:	High

Table 33: 2.10 Create Role – Description

Functional Requirement	Explanation
Requirement number:	2.12
Requirement name (use case name):	View Role
Requirement short description:	The Super Admin can view details of the existing roles.
Requirement detailed description and constraints:	The Super Admin can search and see role details. No changes allowed in view mode. Constraints: • Internet connection required.
Business rules applicable to this requirement	• Only the Super Admin can view roles.
Revision date and Revision number:	2026-06-24 Version 4.0
Criticality/Priority:	High

Table 34: 2.9 View Role – Description

Functional Requirement	Explanation
Requirement number:	2.13
Requirement name (use case name):	Update Role
Requirement short description:	The Super Admin can update existing roles.
Requirement detailed description and constraints:	The Super Admin can modify role names or permissions. System validates changes before confirmation. Constraints: • Internet connection required.
Business rules applicable to this requirement	• The Super Admin can update roles.
Revision date and Revision number:	2026-06-24 Version 4.0
Criticality/Priority:	High

Table 35: 2.10 Update Role – Description

Functional Requirement	Explanation
Requirement number:	2.14
Requirement name (use case name):	Reactivate Role
Requirement short description:	The Super Admin can reactivate deactivated roles.
Requirement detailed description and constraints:	The Super Admin selects a deactivated role. System confirms before reactivation. Constraints: • Internet connection required.
Business rules applicable to this requirement	• Only the Super Admin can reactivate a role.
Revision date and Revision number:	2026-06-24 Version 4.0
Criticality/Priority:	High

Table 36:2.11 Reactivate Role – Description

Functional Requirement	Explanation
Requirement number:	2.15
Requirement name (use case name):	Deactivate Role
Requirement short description:	The Super Admin can deactivate roles.
Requirement detailed description and constraints:	The Super Admin selects a role. System asks for confirmation before applying deactivation. Constraints: • Internet connection required. • Action must be logged.
Business rules applicable to this requirement	• Only the Super Admin can deactivate roles.
Revision date and Revision number:	2026-06-24 Version 4.0
Criticality/Priority:	High

Table 37: 2.12 Deactivate Role – Description

Functional Requirement	Explanation
Requirement number:	2.16
Requirement name (use case name):	Update Role Permission
Requirement short description:	The Super Admin modifies existing role permissions. System validates changes before confirmation.
Requirement detailed description and constraints:	The Super Admin modifies existing role permissions. System validates changes before confirmation. Constraints: • Internet connection required.
Business rules applicable to this requirement	• The Super Admin can update role permissions.
Revision date and Revision number:	2026-06-24 Version 4.0
Criticality/Priority:	High

Table 38: 2.13 Update Role Permission – Description

Functional Requirement	Explanation
Requirement number:	2.17
Requirement name (use case name):	View Role Permission
Requirement short description:	The Super Admin can view permissions assigned to roles.
Requirement detailed description and constraints:	The Super Admin can search and view role permissions. No edits allowed. Constraints: • Internet connection required.
Business rules applicable to this requirement	• The Super Admin can view role permissions.
Revision date and Revision number:	2026-06-24 Version 4.0
Criticality/Priority:	High

Table 39: 2.14 View Role Permission – Description

2.10.3.3 Inventory Subsystem

Functional Requirement	Explanation
Requirement number:	3.1
Requirement name (use case name):	Create Procurement
Requirement short description:	A user must be able to add an item(s) to the system's inventory.
Requirement detailed description and constraints:	A user must be able to add an item(s) to the system's inventory once the item is received from a vendor. The system will allow the admin to enter details regarding the inventory received. The system will validate the details entered before asking for confirmation. Once confirmed the system will add the inventory to the system. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with the required permissions can receive inventory.
Revision date and Revision number:	2026-07-19 Version 4.0
Criticality/Priority:	Medium

Table 40: 3.1 Receive Inventory Item – Description

Functional Requirement	Explanation
Requirement number:	3.2
Requirement name (use case name):	View Procurement
Requirement short description:	A user must be able to view a procurements detail on the system.
Requirement detailed description and constraints:	This use case describes the event in which the user views a procurement on the system. The system will retrieve and display the procurements detail on the system. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with the required permissions can view procurements.
Revision date and Revision number:	2026-07-19 Version 4.0
Criticality/Priority:	Medium

Table 41: 3.2 View Procurement – Description

Functional Requirement	Explanation
Requirement number:	3.3
Requirement name (use case name):	View Inventory
Requirement short description:	A user must be able to view all inventory on the system.
Requirement detailed description and constraints:	This use case describes the event in which the user views the inventory on the system. The system will retrieve and display all the inventory on the system as well as allowing the user to filter. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with the required permissions can view inventory.
Revision date and Revision number:	2026-07-19 Version 4.0
Criticality/Priority:	Medium

Table 42: 3.2 View Inventory – Description

Functional Requirement	Explanation
Requirement number:	3.4
Requirement name (use case name):	Write-Off Inventory.
Requirement short description:	A user with appropriate permissions must be able to write off an inventory item(s).
Requirement detailed description and constraints:	A user with appropriate permissions must be able to edit the details of an inventory item(s). The system will display all the inventory on the system and allow the user to write-off inventory. The system will ask for confirmation before the inventory items are written off. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with the required permissions can write-off inventory.
Revision date and Revision number:	2026-07-19 Version 4.0
Criticality/Priority:	Medium

Table 43: 3.3 Write-Off Inventory – Description

Functional Requirement	Explanation
Requirement number:	3.5
Requirement name (use case name):	Allocate Inventory
Requirement short description:	A user with the appropriate role must be able to allocate inventory to a project.
Requirement detailed description and constraints:	A user with the appropriate role must be able to allocate inventory to a project. The system will display all the inventory and allow the user to allocate inventory to an active project. The system will validate inputted data and ask for confirmation before allocating inventory. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with the required permissions can allocate inventory.
Revision date and Revision number:	2026-07-19 Version 4.0
Criticality/Priority:	Medium

Table 44: 3.4 Allocate Inventory – Description

Functional Requirement	Explanation
Requirement number:	3.6
Requirement name (use case name):	Create Inventory Item Type
Requirement short description:	A user with the appropriate permissions must be able to create a new inventory item type.
Requirement detailed description and constraints:	A user with the appropriate permissions must be able to create a new inventory item type that will then be selectable when adding inventory items. Information including name, description and security level. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can create an inventory item type.
Revision date and Revision number:	2026-07-19 Version 4.0
Criticality/Priority:	Low

Table 45: 3.5 Create Inventory Item Type – Description

Functional Requirement	Explanation
Requirement number:	3.7
Requirement name (use case name):	View Inventory Item Types
Requirement short description:	A user must be able to view all inventory item types.
Requirement detailed description and constraints:	A user must be able to view all inventory item types. The system will retrieve and display all the inventory item types on the system before displaying them to the user. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can view all inventory item types.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Low

Table 46:3.7 View Inventory Item Types – Description

Functional Requirement	Explanation
Requirement number:	3.8
Requirement name (use case name):	Update Inventory Item Type
Requirement short description:	A user with the appropriate permissions must be able to update an inventory item type.
Requirement detailed description and constraints:	A user with the appropriate permissions must be update inventory item type that will then be selectable when adding inventory items. Information including name, description and security level can be changed. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can update an inventory item type.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Low

Table 47: 3.8 Update Inventory Item Type – Description

Functional Requirement	Explanation
Requirement number:	3.9
Requirement name (use case name):	Deactivate Inventory Item Type
Requirement short description:	A user with the appropriate permissions must be able to deactivate an inventory item type.
Requirement detailed description and constraints:	A user with the appropriate permissions must be able to deactivate an inventory item type if the type is no longer needed by the organization due to a change in policy or project scope. Data regarding previous use of the inventory item type will be kept for accurate record keeping, but the type will no longer appear when selecting an inventory item type. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can deactivate an inventory item type.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Low

Table 48: 3.9 Deactivate Inventory Item Type - Description

Functional Requirement	Explanation
Requirement number:	3.10
Requirement name (use case name):	Reactivate Inventory Item Type
Requirement short description:	A user with the appropriate permissions must be able to reactivate an inventory item type.
Requirement detailed description and constraints:	A user with the appropriate permissions must be able to reactivate an inventory item type. The type will appear again when selecting an inventory item type. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can reactivate an inventory item type.
Revision date and Revision number:	2026-07-19 Version 4.0
Criticality/Priority:	Low

Table 49: 3.10 Reactivate Inventory Item Type - Description

2.10.3.4 Group Subsystem

Functional Requirement	Explanation
Requirement number:	4.1
Requirement name (use case name):	Create Group
Requirement short description:	The system must allow a user with appropriate permissions to create a group for a project.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to create a group that is associated with a specific project. A project can have multiple groups within it and group members can see information related to them as well as communicate with each other. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Each group must be linked with an existing project on the system. • Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 50: 4.1 Create Group – Description

Functional Requirement	Explanation
Requirement number:	4.2
Requirement name (use case name):	View Groups
Requirement short description:	The system must allow users to view the details of the group they are in for a project.
Requirement detailed description and constraints:	The system must allow users on the system to view a group's details. When a user selects to view a group, the system will display information such as the group name, the project it belongs to and the other members. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Users can only view groups they are assigned too. • Users must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 51:4.2 View Group – Description

Functional Requirement	Explanation
Requirement number:	4.3
Requirement name (use case name):	Update Group
Requirement short description:	The system must allow a user with appropriate permissions to update the details of a group.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to update a group's details within a project. The editable details will include the group name and ability to add or remove members. Once the user clicks save, the system will validate the updated data and save the changes. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 52: 4.3 Update Group – Description

Functional Requirement	Explanation
Requirement number:	4.4
Requirement name (use case name):	Deactivate Group
Requirement short description:	The system must allow a user with appropriate permissions to deactivate a group.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to deactivate a group within a project when it is no longer used or required. The system will confirm the deactivation and upon confirmation, the group will be deactivated. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 53: 4.4 Deactivate Group - Description

Functional Requirement	Explanation
Requirement number:	4.5
Requirement name (use case name):	Create Skill
Requirement short description:	The system must allow a user with appropriate permissions to create a new skill.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to create new skills that can be assigned to volunteers and staff members. These skills can be used to help assign volunteers or staff members to appropriate teams. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 54: 4.5 Create Skill - Description

Functional Requirement	Explanation
Requirement number:	4.6
Requirement name (use case name):	View Skills
Requirement short description:	The system must allow users to view skills stored on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to view the skills that have been created on the system. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Users must be logged into the system. • Skills must be displayed accurately.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 55: 4.6 View Skills - Description

Functional Requirement	Explanation
Requirement number:	4.7
Requirement name (use case name):	View Skill
Requirement short description:	The system must allow users to view a skill stored on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to view a skill that is on the system. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Users must be logged into the system. • Skills must be displayed accurately.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 56: 4.7 View Skill - Description

Functional Requirement	Explanation
Requirement number:	4.8
Requirement name (use case name):	Update Skill
Requirement short description:	The system must allow a user with appropriate permissions to update skills stored on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to update the skills on the system such as correcting the name or spelling. Once the user clicks save, the system will validate the updated data and save the changes. This will ensure that skill information remains accurate. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 57: 4.8 Update Skill - Description

Functional Requirement	Explanation
Requirement number:	4.9
Requirement name (use case name):	Deactivate Skill
Requirement short description:	The system must allow a user with appropriate permissions to deactivate a skill stored on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to update the skills on the system such as correcting the name or spelling. Once the user clicks save, the system will validate the updated data and save the changes. This will ensure that skill information remains accurate. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 58: 4.9 Deactivate Skill - Description

2.10.3.5 Warehouse Subsystem

Functional Requirement	Explanation
Requirement number:	5.1
Requirement name (use case name):	Create Warehouse
Requirement short description:	The Super Admin can create new warehouses.
Requirement detailed description and constraints:	The Super Admin can input warehouse name, location, and capacity. System validates details before confirmation. Constraints: • Internet connection required. • Location field cannot be empty. • Capacity must be a positive number.
Business rules applicable to this requirement	• Only users with warehouse admin permissions can create warehouses.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 59 :5.1 Create Warehouse - Description

Functional Requirement	Explanation
Requirement number:	5.2
Requirement name (use case name):	View Warehouse
Requirement short description:	The Super Admin can view warehouse details.
Requirement detailed description and constraints:	The Super Admin can search and view warehouses. No edits allowed in view mode. Constraints: • Internet connection required. • Access limited to users with warehouse admin privileges.
Business rules applicable to this requirement	• Only warehouse admin users can view warehouses.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 60: 5.2 View Warehouse - Description

Functional Requirement	Explanation
Requirement number:	5.3
Requirement name (use case name):	Update Warehouse
Requirement short description:	The Super Admin can update warehouse details.
Requirement detailed description and constraints:	The Super Admin modifies warehouse information. System validates changes before confirmation. Constraints: • Internet connection required. • Capacity updates must be positive numbers. • Changes must be logged.
Business rules applicable to this requirement	• Only warehouse admin users can update warehouses.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 61: 5.3 Update Warehouse - Description

Functional Requirement	Explanation
Requirement number:	5.4
Requirement name (use case name):	Reactivate Warehouse
Requirement short description:	The Super Admin can reactivate deactivated warehouses.
Requirement detailed description and constraints:	The Super Admin selects a deactivated warehouse. System asks for confirmation before reactivation. Constraints: • Internet connection required. • Action must be logged.
Business rules applicable to this requirement	• Only the warehouse admin user can reactivate warehouses.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 62 5.4: Reactivate Warehouse - Description

Functional Requirement	Explanation
Requirement number:	5.5
Requirement name (use case name):	Deactivate Warehouse
Requirement short description:	The Super Admin can deactivate warehouses.
Requirement detailed description and constraints:	The Super Admin selects a warehouse. System asks for confirmation before applying deactivation. Constraints: • Internet connection required. • Action must be logged.
Business rules applicable to this requirement	• Only warehouse admin users can deactivate warehouses.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 63: 5.5 Deactivate Warehouse - Description

Functional Requirement	Explanation
Requirement number:	5.6
Requirement name (use case name):	Inventory Stock Take.
Requirement short description:	A user with appropriate permissions must be able to perform a stock take.
Requirement detailed description and constraints:	A user with appropriate permissions must be able to perform a stock take. The system will allow the user to select a warehouse and view its inventory. The user will perform a stock take and enter the stock count. The system will determine the difference and capture the difference to adjust the inventory. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with the required permissions can receive perform a stock take.
Revision date and Revision number:	2026-03-29 Version 3.0
Criticality/Priority:	High

Table 64: Inventory Stock Take - Description

2.10.3.6 Vendor Subsystem

Functional Requirement	Explanation
Requirement number:	6.1
Requirement name (use case name):	Create Vendor
Requirement short description:	A user must be able to add a vendor to the system.
Requirement detailed description and constraints:	A user must be able to add a vendor to the system by adding information including vendor name, type, email, phone, contact person, credit facility (if applicable), address, rating and photo. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can add a vendor.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 65: 6.1 Create Vendor - Description

Functional Requirement	Explanation
Requirement number:	6.2
Requirement name (use case name):	View Vendor
Requirement short description:	A user must be able to view any given vendor.
Requirement detailed description and constraints:	A user must be able to view any given vendor. The system will then retrieve and display all relevant information regarding the vendor. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can view a vendor.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 66 6.2: View Vendor - Description

Functional Requirement	Explanation
Requirement number:	6.3
Requirement name (use case name):	View Vendors
Requirement short description:	A user must be able to view all vendors.
Requirement detailed description and constraints:	A user must be able to view all vendors on the system. The system will retrieve and display all vendors and their information for the user. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can view all vendor.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 67: 6.3 View Vendors - Description

Functional Requirement	Explanation
Requirement number:	6.4
Requirement name (use case name):	Edit Vendor
Requirement short description:	A user must be able to edit vendor information
Requirement detailed description and constraints:	A user must be able to edit a vendor on the system by changing information including vendor name, type, email, phone, contact person, credit facility (if applicable), address, rating and photo. Constraints: • The user requires an Internet connection to access this feature. • The vendor must already exist on the system.
Business rules applicable to this requirement	• Only a user with appropriate permissions can edit a vendor.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 68: 6.4 Edit Vendor - Description

Functional Requirement	Explanation
Requirement number:	6.5
Requirement name (use case name):	Reactivate Vendor
Requirement short description:	A user must be able to reactivate a vendor.
Requirement detailed description and constraints:	A user must be able to reactivate a vendor on the system. The system will display the vendor and allow the user to reactivate the vendor. The system will ask the user to confirm the reactivation before the vendor is reactivated Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can reactivate a vendor.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 69: 6.5 Reactivate Vendor - Description

Functional Requirement	Explanation
Requirement number:	6.6
Requirement name (use case name):	Deactivate Vendor
Requirement short description:	A user must be able to deactivate any given vendor.
Requirement detailed description and constraints:	A user must be able to deactivate any given vendor in the case that the vendor is no longer in use, or the vendor no longer exists. The system will display the vendor and allow the user to deactivate the vendor. The system will ask the user to confirm the deactivation before the vendor is deactivated. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Only a user with appropriate permissions can deactivate a vendor.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 70: 6.6 Deactivate Vendor - Description

Functional Requirement	Explanation
Requirement number:	6.7
Requirement name (use case name):	Create Vendor Type
Requirement short description:	A user with the appropriate permissions must be able to create a type that vendors can be categorised under.
Requirement detailed description and constraints:	A user with the appropriate permissions must be able to create a type that vendors can be categorised under. These types will be used when updating and creating vendors. It also ensures that vendors can be easily filtered. The details will include type name, description and status. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Vendor management capabilities. • Use of Preferred vendors. • Role based access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 71: 6.7 Create Vendor Type - Description

Functional Requirement	Explanation
Requirement number:	6.8
Requirement name (use case name):	Update Vendor Type
Requirement short description:	A user with the appropriate permissions must be able to update a vendor type.
Requirement detailed description and constraints:	A user with the appropriate permissions must be able to update a vendor type. This includes changes to the type name and description. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature. • Vendor type must already exist on the system.
Business rules applicable to this requirement	• Vendor management capabilities. • Use of Preferred vendors. • Role based access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 72: 6.8 Update Vendor Type - Description

Functional Requirement	Explanation
Requirement number:	6.9
Requirement name (use case name):	View Vendor Types
Requirement short description:	A user with the appropriate permissions must be able to view the vendor types.
Requirement detailed description and constraints:	A user must be able to view the vendor types and then view all the available information regarding the vendor types. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Vendor management capabilities. • Use of Preferred vendors. • Role based access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 73: 6.8 View Vendor Types - Description

Functional Requirement	Explanation
Requirement number:	6.10
Requirement name (use case name):	Reactivate Vendor Type
Requirement short description:	A user with the appropriate permissions must be able to view the vendor types.
Requirement detailed description and constraints:	A user with the appropriate permissions must be able to reactivate a vendor type when it is required again by the organization due to a change in policy or project scope. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature. • Vendor type must already exist on the system.
Business rules applicable to this requirement	• Vendor management capabilities. • Use of Preferred vendors. • Role based access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 74: 6.10 Reactivate Vendor Type – Description

Functional Requirement	Explanation
Requirement number:	6.11
Requirement name (use case name):	Deactivate Vendor Type
Requirement short description:	A user with the appropriate permissions must be able to deactivate a vendor type.
Requirement detailed description and constraints:	A user with the appropriate permissions must be able to deactivate a vendor type if the type is no longer needed by the organization due to a change in policy or project scope. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature. • Vendor type must already exist on the system.
Business rules applicable to this requirement	• Vendor management capabilities. • Use of Preferred vendors. • Role based access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 75: 6.11 Deactivate Vendor Type - Description

2.10.3.7 Project Management Subsystem

Functional Requirement	Explanation
Requirement number:	7.1
Requirement name (use case name):	Create Project
Requirement short description:	A Super-Admin creates a new project representing an NGO goal by inputting project details and team assignments.
Requirement detailed description and constraints:	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. Constraints: • 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. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 76: 7.1 Create Project - Description

Functional Requirement	Explanation
Requirement number:	7.2
Requirement name (use case name):	View Project
Requirement short description:	A user clicks on a specific project to see all its details.
Requirement detailed description and constraints:	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. The system retrieves the details and displays them in sections relative to the user's role. Constraints: • The user must already have a pre-existing account on the system. The user is logged in. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 77: 7.2 View Project - Description

Functional Requirement	Explanation
Requirement number:	7.3
Requirement name (use case name):	Update Project
Requirement short description:	A Super-Admin will edit an existing project already saved in the system.
Requirement detailed description and constraints:	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. The system will capture all the updated information and alert all the relevant staff. Constraints: • 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. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 78: 7.3 Update Project - Description

Functional Requirement	Explanation
Requirement number:	7.4
Requirement name (use case name):	Deactivate Project
Requirement short description:	A Super Admin can deactivate projects after they are completed or no longer support goals.
Requirement detailed description and constraints:	A Super Admin can deactivate projects after they are completed or if the project no longer supports the NPO's organisational goals. The system will load the form and ask the user for confirmation before deactivating the project and notifying the admin of the status change. Constraints: • The project is already created and saved on the system. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 79: 7.4 Deactivate Project - Description

Functional Requirement	Explanation
Requirement number:	7.5
Requirement name (use case name):	Create Project Type
Requirement short description:	The Super Admin creates a new category for projects.
Requirement detailed description and constraints:	The Super Admin would like to create a new category for projects that will represent a type of project. The system will display a form where the admin can type the name of the new project type, which the system then verifies and saves. Constraints: • The user must already have a pre-existing account on the system. • The user has Super Admin privileges. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 80: 7.5 Create Project Type - Description

Functional Requirement	Explanation
Requirement number:	7.6
Requirement name (use case name):	View Project Type
Requirement short description:	The Super Admin can see, edit, or deactivate project types.
Requirement detailed description and constraints:	The Super Admin can click "View Project types" to be able to see, edit, or deactivate project types to better categorise the projects being completed. The system retrieves recorded project types and displays them in a table for the Super Admin to view. Constraints: • The user must already have a pre-existing account on the system. • The user must have Super Admin privileges. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 81: 7.6 View Project Type - Description

Functional Requirement	Explanation
Requirement number:	7.7
Requirement name (use case name):	Update Project Type
Requirement short description:	A Super Admin wants to change the name of a project type.
Requirement detailed description and constraints:	A Super Admin wants to change the name of a project type so users can better understand what those types of projects represent. The system retrieves the recorded project type and displays the name as an editable field for the admin to update. Constraints: • The project type has already been added and captured onto the system. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 82: 7.7 Update Project Type - Description

Functional Requirement	Explanation
Requirement number:	7.8
Requirement name (use case name):	Deactivate Project Type
Requirement short description:	The Super Admin deactivates a project type no longer being used.
Requirement detailed description and constraints:	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. The system loads the project type name for confirmation before deactivating it. Constraints: • The project type has already been captured on the system. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 83: 7.8 Deactivate Project Type - Description

Functional Requirement	Explanation
Requirement number:	7.9
Requirement name (use case name):	Add Indemnity Form
Requirement short description:	The Super Admin can add a new Indemnity form to projects.
Requirement detailed description and constraints:	The Super Admin can add a new Indemnity form that can be added to projects to make form completions and capturing easier. The system shows fields to place a name for the form and a field to upload a pdf file or paste text. Constraints: • The admin is signed in, and the form hasn't been added yet. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 84: 7.9 Add Indemnity Form - Description

Functional Requirement	Explanation
Requirement number:	7.10
Requirement name (use case name):	Update Indemnity Form
Requirement short description:	The Super Admin can update an existing Indemnity form.
Requirement detailed description and constraints:	The Super Admin can update an existing Indemnity form that can be added to projects to make form completions and capturing easier. The system displays editable fields where the admin can change the name, text, or pdf file of the form. Constraints: • The admin is signed in, and the form hasn't been added yet. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 85: 7.10 Update Indemnity Form - Description

Functional Requirement	Explanation
Requirement number:	7.11
Requirement name (use case name):	View Indemnity Form
Requirement short description:	The Super Admin can see, edit, add or deactivate Indemnity Forms.
Requirement detailed description and constraints:	The Super Admin can click "View Indemnity Form" to be able to see, edit, add or deactivate Indemnity Forms to better help the projects. The system retrieves the recorded forms and displays them in a table for the admin to manage. Constraints: • The user must already have a pre-existing account on the system. • The user must have Super Admin privileges. • Must be completed before project day. • Must be user-friendly and easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 86: 7.11 View Indemnity Form - Description

Functional Requirement	Explanation
Requirement number:	7.12
Requirement name (use case name):	Remove Indemnity Form
Requirement short description:	The Super Admin can remove an existing indemnity form no longer being used.
Requirement detailed description and constraints:	The Super Admin can remove an existing indemnity form that is no longer being used to reduce clutter. The system loads the form content for deactivation confirmation and notifies the user once the form is removed from the table. Constraints: • The admin is signed in, and the form hasn't been added yet. • Must be completed before project day. • Must be user-friendly and easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 87: 7.12 Remove Indemnity Form - Description

Functional Requirement	Explanation
Requirement number:	7.13
Requirement name (use case name):	Capture Indemnity Form
Requirement short description:	A Super Admin can attach an indemnity form to a project.
Requirement detailed description and constraints:	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. The system captures the selection and requires signatures from staff. Constraints: • The Super Admin must already have a pre-existing account on the system. • Must be completed before project day. • Must be user-friendly and easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 88: 7.13 Capture Indemnity Form - Description

Functional Requirement	Explanation
Requirement number:	7.14
Requirement name (use case name):	Capture Project Feedback
Requirement short description:	A Super Admin can attach a project feedback form to a project.
Requirement detailed description and constraints:	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 sent a form 1 week after the project commences asking them for a rating and comments. The system captures the selection for all involved staff. Constraints: • The project assigned to the feedback form must have commenced. • Must be completed before project day. • Must be user-friendly and easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 89: 7.14 Capture Project Feedback - Description

Functional Requirement	Explanation
Requirement number:	7.15
Requirement name (use case name):	View Project Feedback
Requirement short description:	A Super Admin can view project feedback form ratings and comments.
Requirement detailed description and constraints:	A Super Admin can view all the past projects and see the feedback forms ratings and comments for each project. The system retrieves the records for feedback and displays it neatly for the admin to read. Constraints: • The projects have been saved on the system and not been deactivated. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 90: 7.15 View Project Feedback - Description

Functional Requirement	Explanation
Requirement number:	7.16
Requirement name (use case name):	Create Impact Feedback Form
Requirement short description:	The Super Admin can add a new impact form for community members.
Requirement detailed description and constraints:	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. The system shows fields to name the form and buttons to add and define question types. Constraints: • The admin creating the form has permissions to create impact forms already. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 91: 7.16 Create Impact Feedback Form - Description

Functional Requirement	Explanation
Requirement number:	7.17
Requirement name (use case name):	View Impact Forms
Requirement short description:	A Super Admin can view project impact form questions and answers.
Requirement detailed description and constraints:	A Super Admin can view all the past projects and see the impact forms questions and answers and see what project they are assigned to. The system retrieves records for impact data and displays it neatly for the admin. Constraints: • The projects have been saved on the system and not been deactivated. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 92: 7.17 View Impact Forms - Description

Functional Requirement	Explanation
Requirement number:	7.18
Requirement name (use case name):	Update Impact Form
Requirement short description:	The Super Admin can update an existing impact form.
Requirement detailed description and constraints:	The Super Admin can update an existing impact form that can be added to projects to make form completions and capturing easier. The system shows fields for the form name and the questions currently on the form as editable content. Constraints: • The admin is signed in, and the form hasn't been added yet. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 93: 7.18 Update Impact Form - Description

Functional Requirement	Explanation
Requirement number:	7.19
Requirement name (use case name):	Deactivate Impact Form
Requirement short description:	The admin can deactivate an impact form from the list of all forms.
Requirement detailed description and constraints:	The admin can deactivate an impact form from the list of all the forms being displayed; this can remove unused forms and clear clutter from the system. The system loads the form name for deactivation confirmation before removing it from view. Constraints: • The impact form has already been created and is displayed on the list. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 94: 7.19 Deactivate Impact Form - Description

Functional Requirement	Explanation
Requirement number:	7.20
Requirement name (use case name):	Capture Impact Form
Requirement short description:	The Super Admin can include an impact form and send it to a community partner.
Requirement detailed description and constraints:	When creating or editing a project the Super Admin can check the "Include impact form" button and choose from the list or create a new one. The system will capture an email address or phone number to send the form to the community partner. Constraints: • The impact form has already been captured in the system. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 95: 7.20 Capture Impact Form - Description

Functional Requirement	Explanation
Requirement number:	7.21
Requirement name (use case name):	View Impact Feedback
Requirement short description:	A Super Admin can view all of the project's impact results.
Requirement detailed description and constraints:	A Super Admin can view all of the project's impact results and see the impact form questions and answers, the assigned project, and which community partner answered. The system retrieves and displays these records neatly. Constraints: • The impact forms have been saved on the system and not been deactivated. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 96: 7.21 View Impact Feedback - Description

Functional Requirement	Explanation
Requirement number:	7.22
Requirement name (use case name):	Add Photo
Requirement short description:	A user can upload multiple photos to a project's album.
Requirement detailed description and constraints:	A user can click on a project to see all photos from that project, then click the add photos button to upload multiple photos to that album. The system loads a page where images can be dragged, dropped, or uploaded from the device. Constraints: • The project's photo group has already been added, and the user has access to that project. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 97: 7.22 Add Photo - Description

Functional Requirement	Explanation
Requirement number:	7.23
Requirement name (use case name):	View Photo
Requirement short description:	A user can see all photos from a specific project.
Requirement detailed description and constraints:	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 a representative project photo. Constraints: • All the project's photo group has already been added, and the user has access to those projects. • Must be completed before project day. • Must be user-friendly and easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 98: 7.23 View Photo - Description

Functional Requirement	Explanation
Requirement number:	7.24
Requirement name (use case name):	Delete Photo
Requirement short description:	A user can remove photos from a project's image folder.
Requirement detailed description and constraints:	A user can click on a project to see all photos and then click the delete photos button; the system then displays an "X" button on photos they uploaded to remove them. The system captures the selection and updates the album. Constraints: • The project's photo group has already been added, and the user has access to that project. • The user is deleting photo's they uploaded or are a Super Admin. • Must be completed before project day. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 99: 7.24 Delete Photo - Description

Functional Requirement	Explanation
Requirement number:	7.25
Requirement name (use case name):	Add Photo Album
Requirement short description:	A user with appropriate permissions can create a new photo album for a project.
Requirement detailed description and constraints:	The system will allow an authorized user to create a new photo album under a specific project. The user provides an album title, optional description, and assigns media permissions or categories. Upon confirmation, the system creates the album folder structure and records the creation entry. Constraints: • The target project must already exist in the system and be active. • Users must be logged in. • Album names must be unique within the same project. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-07-22 Version 1.0
Criticality/Priority:	High

Table 100: 7.25 Add Photo Album - Description

Functional Requirement	Explanation
Requirement number:	7.26
Requirement name (use case name):	View Photo Album
Requirement short description:	A user with appropriate permissions can view a project's photo album and its contents.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to view photo albums associated with a project. The user selects a project photo album to view a gallery grid of thumbnail images, album metadata, and photo upload timestamps. Constraints: • The user must have read access to the associated project. • Users must be logged in. • Must load images efficiently. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-07-22 Version 1.0
Criticality/Priority:	High

Table 101: 7.26 View Photo Album - Description

Functional Requirement	Explanation
Requirement number:	7.27
Requirement name (use case name):	Update Photo Album
Requirement short description:	A user with appropriate permissions can update the details and organization of an existing photo album.
Requirement detailed description and constraints:	The system will allow an authorized user to edit the metadata of an existing photo album (such as the album title, description, or cover photo) and manage its photo organization. Upon saving changes, the system validates the inputs and updates the album context. Constraints: • The photo album must already exist. • The user is the creator of the album or is a Super Admin. • Users must be logged in. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-07-22 Version 1.0
Criticality/Priority:	High

Table 102: 7.27 Update Photo Album - - Description

Functional Requirement	Explanation
Requirement number:	7.28
Requirement name (use case name):	Remove Photo Album
Requirement short description:	A user with appropriate permissions can delete an entire photo album and its associated files.
Requirement detailed description and constraints:	The system will allow an authorized user to delete an existing photo album from a project. The user selects the album, clicks the delete button, and confirms the deletion via a confirmation modal. The system then removes the album record and purges all underlying photo files associated with that album. Constraints: • The photo album must already exist. • The user created the album or is a Super Admin. • Users must be logged in. • Deletion requires user confirmation to prevent accidental data loss. • Must be easy to use.
Business rules applicable to this requirement	• Secure Access • Role Based Access
Revision date and Revision number:	2026-07-22 Version 1.0
Criticality/Priority:	High

Table 103: 2.28 Remove Photo Album – Description

2.10.3.8 K9 Subsystem

Functional Requirement	Explanation
Requirement number:	8.1
Requirement name (use case name):	Add Dog
Requirement short description:	The system must allow a user with appropriate permissions to add a dog to the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to capture and store the details of new dogs used in K9 operations. The details will include the dog's name, breed, age, gender, role, status, handler and parents. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • Each dog must be assigned a handler when applicable. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 104: 8.1 Add Dog - Description

Functional Requirement	Explanation
Requirement number:	8.2
Requirement name (use case name):	View Dog
Requirement short description:	The system must allow a user with appropriate permissions to view a dog on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to view the details of each dog used in K9 operations. This will help users to monitor and manage the K9 unit. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • Dog information must be stored and displayed accurately. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 105: 8.2 View Dog - Description

Functional Requirement	Explanation
Requirement number:	8.3
Requirement name (use case name):	View Dogs
Requirement short description:	The system must allow a user with appropriate permissions to view the dogs on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to view the dogs used in K9 operations. This will help users to monitor and manage the K9 unit. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 106: 8.3 View All Dogs- Description

Functional Requirement	Explanation
Requirement number:	8.4
Requirement name (use case name):	Update Dog
Requirement short description:	The system must allow a user with appropriate permissions to update a dog on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to update the details of an existing dog used in the K9 operations. The editable details will include the dog's name, breed, age, gender, role, and parents. Once the user clicks save, the system will validate the updated data and save the changes. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 107: 8.4 Update Dog - Description

Functional Requirement	Explanation
Requirement number:	8.5
Requirement name (use case name):	Deactivate Dog
Requirement short description:	The system must allow a user with appropriate permissions to deactivate a dog on the system.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to deactivate a dog on the system. This could occur when the dog becomes retired and no longer participating in anti-poaching activities. The system will confirm to deactivate the dog and upon confirmation, the dog will be deactivated. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 108: 8.5 Deactivate Dog - Description

Functional Requirement	Explanation
Requirement number:	8.6
Requirement name (use case name):	Assign User to Dog
Requirement short description:	The system must allow a user with appropriate permissions to assign a user to a dog.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to assign a registered handler to a dog on the system. This will help track who is responsible for the dog and working with the dog during patrols. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • The handler must be registered on the system. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 109: 8.6 Assign User to Dog – Description

Functional Requirement	Explanation
Requirement number:	8.7
Requirement name (use case name):	View User-Dog Assignment
Requirement short description:	The system must allow a user with appropriate permissions to view the assignment of users to dogs.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to view which handlers are assigned to which dogs on the system. This will track responsibilities, ensure accountability, and manage K9 operations effectively. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • All assignments must be stored and displayed accurately. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 110: 8.7 View User-Dog Assignment – Description

Functional Requirement	Explanation
Requirement number:	8.8
Requirement name (use case name):	Update User-Dog Assignment
Requirement short description:	The system must allow a user with appropriate permissions to update the assignment of users to dogs.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to update a dog's handler from one handler to another if needed. Once the user clicks save, the system will validate the updated data and save the changes. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • The new handler must be registered on the system. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 111: 8.8 Update User-Dog Assignment - Description

Functional Requirement	Explanation
Requirement number:	8.9
Requirement name (use case name):	Reactivate User-Dog Assignment
Requirement short description:	The system must allow a user with appropriate permissions to reactivate the assignment of users to dogs.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to reactivate a previous assignment between a handler and a dog. The system will confirm the reactivation and upon confirmation, the user-dog assignment will be reactivated. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 112: 8.9 Reactivate User-Dog Assignment - Description

Functional Requirement	Explanation
Requirement number:	8.10
Requirement name (use case name):	Deactivate User-Dog Assignment
Requirement short description:	The system must allow a user with appropriate permissions to deactivate the assignment of users to dogs.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to deactivate an existing assignment between a handler and a dog. The system will confirm the deactivation and upon confirmation, the user-dog assignment will be deactivated. This will remove the handler from being responsible for the dog. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 113: 8.10 Deactivate User-Dog Assignment - Description

Functional Requirement	Explanation
Requirement number:	8.11
Requirement name (use case name):	Add K9 Role
Requirement short description:	The system must allow a user with appropriate permissions to add a K9 role.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to add a new K9 role. These roles will define the responsibilities of each dog including tracking or attacking. These roles help to assign dogs to activities that match their abilities. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 114: 8.11 Add K9 Role - Description

Functional Requirement	Explanation
Requirement number:	8.12
Requirement name (use case name):	Update K9 Role
Requirement short description:	The system must allow a user with appropriate permissions to update a K9 role.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to update an existing K9's role on the system. The editable details will include the name or spelling of the role. Once the user clicks save, the system will validate the updated data and save the changes. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 115: 8.12 Update K9 Role - Description

Functional Requirement	Explanation
Requirement number:	8.13
Requirement name (use case name):	View K9 Roles
Requirement short description:	The system must allow a user with appropriate permissions to view K9 roles.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to view a list of K9 roles on the system. These roles describe the responsibilities of each dog such as a tracking dog or attacking dog. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • K9 roles must be displayed accurately. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 116: 8.13 View K9 Role - Description

Functional Requirement	Explanation
Requirement number:	8.14
Requirement name (use case name):	Reactivate K9 Role
Requirement short description:	The system must allow a user with appropriate permissions to reactivate a K9 role.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to reactivate a K9 role on the system when it is required again. The system will confirm the reactivation and upon confirmation, the K9 role will be reactivated. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 117: 8.14 Reactivate K9 Role - Description

Functional Requirement	Explanation
Requirement number:	8.15
Requirement name (use case name):	Deactivate K9 Role
Requirement short description:	The system must allow a user with appropriate permissions to remove a K9 role.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to deactivate an existing K9 role from the system when it is no longer required. The system will confirm the deactivation and upon confirmation, the K9 role will be deactivated. Constraints: - Time constraint: the use case must be completed before the Project Day in October 2026. - The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	- Role based access. - User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 118: 8.15 Deactivate K9 Role – Description

Functional Requirement	Explanation
Requirement number:	8.16
Requirement name (use case name):	View K9 Ancestry
Requirement short description:	The system must allow a user with appropriate permissions to view the ancestry of a K9.
Requirement detailed description and constraints:	The system must allow a user with appropriate permissions to view a dog's family tree showing the parentage and lineage of the dog. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Role based access. • User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 119: 8.16 View K9 Ancestry – Description

2.10.3.9 Rhino Subsystem

Functional Requirement	Explanation
Requirement number:	9.1
Requirement name (use case name):	Add Rhino
Requirement short description:	The system allows users with appropriate permissions to add a rhino.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to store information about rhinos. The user will enter the required details including name, species, weight, date of birth, gender, horn status, status, last seen, identification notes, approximate location (kept vague for security reasons) and photo. Constraints: • Fields must pass input validation before being captured. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can add a rhino. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 120: 9.1 Add Rhino - Description

Functional Requirement	Explanation
Requirement number:	9.2
Requirement name (use case name):	View Rhino
Requirement short description:	The system allows users with appropriate permissions to view a rhino.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to access and view information about a rhino stored on the system. Constraints: • The rhino must already exist on the system. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can view a rhino's details. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 121:9.2 View Rhino - Description

Functional Requirement	Explanation
Requirement number:	9.3
Requirement name (use case name):	View Rhinos
Requirement short description:	The system allows users with appropriate permissions to view all rhinos.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to access and view all rhinos stored on the system. Constraints: • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can view all rhinos. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 122: 9.3 View Rhinos - Description

Functional Requirement	Explanation
Requirement number:	9.4
Requirement name (use case name):	Update Rhino
Requirement short description:	The system allows users with appropriate permissions to update rhino details.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to update rhino details that are stored on the system. The system will display the rhino's current details and allow the user to change them. This includes name, species, weight, date of birth, gender, horn status, status, last seen, identification notes, approximate location (kept vague for security reasons) and photo. Constraints: • The rhino must already exist on the system. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can update a rhino. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 123: 9.4 Update Rhino - Description

Functional Requirement	Explanation
Requirement number:	9.5
Requirement name (use case name):	Deactivate Rhino
Requirement short description:	The system allows users with appropriate permissions to remove a rhino; historic data will be kept.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to remove a rhino from the system. The user will get a confirmation prompt before the rhino is removed from the system. Historic rhino data will be kept. Constraints: • The rhino must already exist on the system. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can deactivate a rhino. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	Medium

Table 124: 9.5 9.5 Deactivate Rhino - Description

2.10.3.10 Elephant Subsystem

Functional Requirement	Explanation
Requirement number:	10.1
Requirement name (use case name):	Add Elephant
Requirement short description:	The system allows users with appropriate permissions to add an elephant.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to store information about elephants. The user will enter the required details including name, species, weight, date of birth, gender, tusk status, status, last seen, identification notes, approximate location (kept vague for security reasons) and photo. Constraints: • Fields must pass input validation before being captured. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can add an elephant. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 125: 10.1 Add Elephant - Description

Functional Requirement	Explanation
Requirement number:	10.2
Requirement name (use case name):	View Elephant
Requirement short description:	The system allows users with appropriate permissions to view an elephant's details.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to access and view information about an elephant stored on the system. Constraints: • The elephant must already exist on the system. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can view an elephant's details. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 126: 10.2 View Elephant - Description

Functional Requirement	Explanation
Requirement number:	10.3
Requirement name (use case name):	View Elephants
Requirement short description:	The system allows users with appropriate permissions to view all elephants.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to access and view all elephants stored on the system. Constraints: • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can view all elephants. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 127: 10.3 View Elephants - Description

Functional Requirement	Explanation
Requirement number:	10.4
Requirement name (use case name):	Update Elephant
Requirement short description:	The system allows users with appropriate permissions to update an elephant's details.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to update elephant details that are stored on the system. The system will display the elephant's current details and allow the user to change them. This includes the name, species, weight, date of birth, gender, tusk status, status, last seen, identification notes, approximate location (kept vague for security reasons) and photo. Constraints: • The elephant must already exist on the system. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can update an elephant. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 128: 10.4 Update Elephant – Description

Functional Requirement	Explanation
Requirement number:	10.5
Requirement name (use case name):	Deactivate Elephant
Requirement short description:	The system allows users with appropriate permissions to deactivate an elephant; historic data will be kept.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to remove an elephant or elephant from the system. The user will get a confirmation prompt before the rhino or elephant is removed from the system. Historic data will be kept. Constraints: • Only users with appropriate permissions can remove an elephant. • The elephant must already exist on the system. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Only a user with appropriate permissions can remove an elephant. • User must be logged in.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 129: 10.5 Deactivate Elephant - Description

2.10.3.11 Messages Subsystem

Functional Requirement	Explanation
Requirement number:	11.1
Requirement name (use case name):	Create Message
Requirement short description:	The system must allow a user to send a message on a group.
Requirement detailed description and constraints:	The system must allow a user to send a message on a group to which they are assigned. The user will type out the message they wish to send and then send the message to a group. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 130: 11.1 Create Message – Description

Functional Requirement	Explanation
Requirement number:	11.2
Requirement name (use case name):	View Messages
Requirement short description:	The system must allow a user to view the messages on a group.
Requirement detailed description and constraints:	The system must allow a user to view a message on a group to which they are assigned. The user selects a group and will be able to see the messages on the group. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• User must be logged into the system.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 131: 11.2 View Message – Description

2.10.3.12 Report Subsystem

Functional Requirement	Explanation
Requirement number:	12.1
Requirement name (use case name):	Generate Custom Report
Requirement short description:	The system allows a user with appropriate permissions to generate a custom report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to generate a custom report. The user can select fields, and the system will generate a report using those fields. Constraints: • Only users with appropriate permissions can generate custom reports. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 132: 12.1 Generate Custom Report - Description

Functional Requirement	Explanation
Requirement number:	12.2
Requirement name (use case name):	View Custom Report
Requirement short description:	The system allows a user with appropriate permissions to view a custom report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to view a custom report. The user can select and view a custom report that has been created. Constraints: • Only users with appropriate permissions can view custom reports. • Report must already exist. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 133: 12.2 View Custom Report - Description

Functional Requirement	Explanation
Requirement number:	12.3
Requirement name (use case name):	Update Custom Report
Requirement short description:	The system allows a user with appropriate permissions to update a custom report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to update a custom report. The system will display the report's current fields and allow the user to change them. Constraints: • Only users with appropriate permissions can update custom reports. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 134: 12.3 Update Custom Report - Description

Functional Requirement	Explanation
Requirement number:	12.4
Requirement name (use case name):	Remove Custom Report
Requirement short description:	The system allows a user with appropriate permissions to remove a custom report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to remove a custom report and its configuration from the database. Constraints: - Only users with appropriate permissions can remove custom reports. - Users must be logged in.
Business rules applicable to this requirement	Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 135: 12.4 Remove Custom Report - Description

Functional Requirement	Explanation
Requirement number:	12.5
Requirement name (use case name):	View Bike Impact Report
Requirement short description:	The system allows a user with appropriate permissions to view a bike impact report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to view a bike impact report. The user Selects a date range, and the pre-structured report is generated for them to view. Constraints: • Only users with appropriate permissions can view bike impact reports. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 136: 12.5 View Bike Impact Report - Description

Functional Requirement	Explanation
Requirement number:	12.6
Requirement name (use case name):	View Budget Report
Requirement short description:	The system allows a user with appropriate permissions to view a budget report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to view a budget report. The user Selects a date range, and the pre-structured report is generated for them to view. Constraints: • Only users with appropriate permissions can view budget reports. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 137: 12.6 View Budget Report – Description

Functional Requirement	Explanation
Requirement number:	12.7
Requirement name (use case name):	View Rhino Report
Requirement short description:	The system allows a user with appropriate permissions to view a rhino report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to view a rhino report. The user Selects a date range, and the pre-structured report is generated for them to view. Constraints: • Only users with appropriate permissions can view rhino reports. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 138: 12.7 View Rhino Report - Description

Functional Requirement	Explanation
Requirement number:	12.10
Requirement name (use case name):	View Elephant Report
Requirement short description:	The system allows a user with appropriate permissions to view an elephant report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to view an elephant report. The user Selects a date range, and the pre-structured report is generated for them to view. Constraints: • Only users with appropriate permissions can view elephant reports. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 139: 12.10 View Elephant Report - Description

Functional Requirement	Explanation
Requirement number:	12.9
Requirement name (use case name):	View K9 Report
Requirement short description:	The system allows a user with appropriate permissions to view a K9 report.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions to view a K9 report. The user Selects a date range, and the pre-structured report is generated for them to view. Constraints: • Only users with appropriate permissions can view K9 reports. • Users must be logged in.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-09 Version 4.0
Criticality/Priority:	High

Table 140: 12.9 View K9 Report – Description

Functional Requirement	Explanation
Requirement number:	12.10
Requirement name (use case name):	View Machine Learning Dashboard
Requirement short description:	A user with the appropriate role should be able to view all current predictions.
Requirement detailed description and constraints:	A user with the appropriate role should be able to view all current predictions. Predictions will appear dynamically as users complete tasks and will also be viewable in the Machine Learning Dashboard. Once a user opens the dashboard, the model will give all predictions based on current data. This will include team predictions for ongoing projects, budget predictions for ongoing projects and risk predictions for all ongoing projects Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	• Intelligent Business Insights.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	Medium

Table 141: 12.10 View Machine Learning Dashboard - Description

2.10.3.13 System Subsystem

Functional Requirement	Explanation
Requirement number:	13.1
Requirement name (use case name):	Backup System
Requirement short description:	The system must allow a user with appropriate permissions to back up the system.
Requirement detailed description and constraints:	The system will allow users with appropriate permissions make a backup of the system data to ensure that information can be recovered in case of a system failure or data loss. The system backup can be done manually or automatically at scheduled intervals. Constraints: Only users with appropriate permissions can perform system backups.
Business rules applicable to this requirement	- Role-based access. - Ability to backup database.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	High

Table 142: 13.1 Backup System - Description

Functional Requirement	Explanation
Requirement number:	13.2
Requirement name (use case name):	Restore System
Requirement short description:	The user should be able to restore data on the system.
Requirement detailed description and constraints:	The user should be able to restore data on the system to a previously backed up version. All users will be logged out by the system before the operation takes place. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature. • Only a Super Administrator can access this feature.
Business rules applicable to this requirement	• Role-based access. • Ability to backup database.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	Medium

Table 143: 13.2 Restore System - Description

Functional Requirement	Explanation
Requirement number:	13.3
Requirement name (use case name):	View Audit Log
Requirement short description:	An administrator should be able to view the audit log
Requirement detailed description and constraints:	An administrator should be able to view the audit log to see who made which important changes when. This allows them to easily identify the source of misuse. Constraints: • Time constraint: the use case must be completed before the Project Day in October 2026. • User skill constraint: the use case must be easy to understand and use, as some users will have limited technology abilities. • The user requires an Internet connection to access this feature. • Only an Administrator can access this feature.
Business rules applicable to this requirement	• Role-based access.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	Medium

Table 144: 13.3 View Audit Log - Description

2.10.3.14 Budget Subsystem

Functional Requirement	Explanation
Requirement number:	14.1
Requirement name (use case name):	View Project Budget
Requirement short description:	A user with the appropriate permissions should be able to view a project budget.
Requirement detailed description and constraints:	A user with the appropriate permissions should be able to view a project budget. The system will allow the user to select and view a budget's details including expenses. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Financial data such as this is privileged, and thus only persons with appropriate roles may view it.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	High

Table 145: 14.1 View Project Budget – Description

Functional Requirement	Explanation
Requirement number:	14.2
Requirement name (use case name):	View Project Budgets
Requirement short description:	A user with the appropriate permissions should be able to view all project budgets.
Requirement detailed description and constraints:	A user with the appropriate permissions should be able to view all project budgets. The system will retrieve all project budgets on the system and display them to the user. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Financial data such as this is privileged, and thus only persons with appropriate roles may view it.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	High

Table 146: 14.2 View Project Budgets – Description

Functional Requirement	Explanation
Requirement number:	14.3
Requirement name (use case name):	Log An Expense
Requirement short description:	A user with appropriate permissions should be able to log an expense.
Requirement detailed description and constraints:	A user with appropriate permissions should be able to log an expense. The system will display the project's budget and allow the user to add an expense by entering the expenses details and the system will validate the inputted data before asking for confirmation Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Financial data such as this is privileged, and thus only persons with appropriate roles may view it.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	High

Table 147: 14.3 Log An Expense – Description

Functional Requirement	Explanation
Requirement number:	14.4
Requirement name (use case name):	Update An Expense
Requirement short description:	A user with appropriate permissions should be able to update an expense.
Requirement detailed description and constraints:	A user with appropriate permissions should be able to update an expense in the edge case that information is entered incorrectly. The system will display the expense's details and allow the user to update its details. The user will enter the updated details, and the system will validate the updated details before asking the user to confirm the updated details Constraints: • The user requires an Internet connection to access this feature. • The expense should already exist on the system.
Business rules applicable to this requirement	• Financial data such as this is privileged, and thus only persons with appropriate roles may view it.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	High

Table 148: 14.4 Update An Expense – Description

Functional Requirement	Explanation
Requirement number:	14.5
Requirement name (use case name):	View An Expense
Requirement short description:	A user with appropriate permissions should be able to view an expense.
Requirement detailed description and constraints:	A user with appropriate permissions should be able to view an expense. The system will display the project budget and allow the user to select an expense to view. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Financial data such as this is privileged, and thus only persons with appropriate roles may view it.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	High

Table 149: 14.5 View An Expense – Description

Functional Requirement	Explanation
Requirement number:	14.6
Requirement name (use case name):	Remove An Expense
Requirement short description:	A user with the appropriate permissions should be able to remove an expense.
Requirement detailed description and constraints:	A user with the appropriate permissions should be able to remove an expense. The system will display the projects budget and expenses and allow the user to remove an expense. The system will ask for confirmation before the expense is removed. Constraints: The user requires an Internet connection to access this feature.
Business rules applicable to this requirement	Financial data such as this is privileged, and thus only persons with appropriate roles may view it.
Revision date and Revision number:	2026-04-28 Version 4.0
Criticality/Priority:	High

Table 150: 16.6 Remove an Expense – Description

2.10.4 Functional Requirements List Detailed Changes Description

2.10.4.1 User Subsystem

User Subsystem		
Use Case Name	Modification Change	Reason for Change
1.1 User Login	1.1 User Login	-
1.2 User Logout	1.2 User Logout	-
1.3 Reset Password	1.3 Reset Password	-
1.4 Create User	1.4 Create User	-
1.5 Update User	1.5 Update User	-
1.6 View User	1.6 View User	-
1.7 Reactivate User	1.7 Reactivate User	-
1.8 Deactivate User	1.8 Deactivate User	-
1.9 View FAQ	1.9 View FAQ	-

Table 151: User Subsystem Modifications

2.10.4.2 Admin Subsystem

Admin Subsystem		
Use Case Name	Modification Change	Reason for Change
2.1 Create User Profile	2.1 Create User Profile	-
2.2 Update User Profile	2.2 Update User Profile	-
2.3 View User Profile	2.3 View User Profile	-
2.4 Deactivate User Profile	2.4 Deactivate User Profile	-
2.5 Add FAQ	2.5 Add FAQ	-
2.6 Update FAQ	2.6 Update FAQ	-
2.7 Remove FAQ	2.7 Remove FAQ	-
-	2.10 Add FAQ Type	Organizes data by creating scalable topic categories instead of a single unmanageable list.
-	2.9 Update FAQ Type	Keeps lookups accurate by instantly cascading updated terminology down to all related FAQs.
-	2.10 Remove FAQ Type	Keeps the system clean by removing obsolete categories while preventing accidental data corruption.
2.10 Create Role	2.11 Create Role	Numbering change
2.9 View Role	2.12 View Role	Numbering change
2.10 Update Role	2.13 Update Role	Numbering change
2.11 Reactivate Role	2.14 Reactivate Role	Numbering change
2.12 Deactivate Role	2.15 Deactivate Role	Numbering change
2.13 Update Role Permission	2.16 Update Role Permission	Numbering change
2.14 View Role Permission	2.17 View Role Permission	Numbering change

Table 152: Admin Subsystem Modifications

2.10.4.3 Inventory Subsystem

Inventory Subsystem		
Use Case Name	Modification Change	Reason for Change
3.1 Receive Inventory Item	3.1 Create Procurement	Name change for better clarity on use case function.
-	3.2 View Procurement	Added to allow admins to view a procurement's details and which procurement an inventory item belongs to.
3.2 View Inventory	3.3 View Inventory	-
3.3 Write Off Inventory	3.4 Write Off Inventory	-
3.4 Allocate Inventory	3.5 Allocate Inventory	-
3.5 Create Inventory Item Type	3.6 Create Inventory Item Type	-
3.6 View Inventory Item Type	Removed	Inventory types are viewed in a table and not individually.
3.7 View Inventory Item Types	3.7 View Inventory Item Types	-
3.8 Update Inventory Item Type	3.8 Update Inventory Item Type	-
3.9 Deactivate Inventory Item Type	3.9 Deactivate Inventory Item Type	-
-	3.10 Reactivate Inventory Item Type	Added to allow super admins the ability to reactivate inventory item types.

Table 153 Inventory Subsystem Modifications

2.10.4.4 Group Subsystem

Group Subsystem		
Use Case Name	Modification Change	Reason for Change
4.1 Create Group	4.1 Create Group	-
4.2 View Groups	4.2 View Groups	-
4.3 Update Group	4.3 Update Group	-
4.4 Deactivate Group	4.4 Deactivate Group	-
4.5 Create Skill	4.5 Create Skill	-
4.6 View Skills	4.6 View Skills	-
4.7 View Skills	4.7 View Skill	-
4.8 Update Skill	4.8 Update Skill	-
4.9 Deactivate Skill	4.9 Deactivate Skill	-

Table 154 Group Subsystem Modifications

2.10.4.5 Warehouse Subsystem

Warehouse Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
5.1 Create Warehouse	5.1 Create Warehouse	-
5.2 View Warehouse	5.2 View Warehouse	-
5.3 Update Warehouse	5.3 Update Warehouse	-
5.4 Reactivate Warehouse	5.4 Reactivate Warehouse	-
5.5 Deactivate Warehouse	5.5 Deactivate Warehouse	-
5.6 Inventory Stock Take	5.6 Inventory Stock Take	-

Table 155: Warehouse Subsystem Modifications

2.10.4.6 Vendor Subsystem

Vendor Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
6.1 Create Vendor	6.1 Create Vendor	-
6.2 View Vendor	6.2 View Vendor	-
6.3 View Vendors	6.3 View Vendors	-
6.4 Edit Vendor	6.4 Edit Vendor	-
6.5 Reactivate Vendor	6.5 Reactivate Vendor	-
6.6 Deactivate Vendor	6.6 Deactivate Vendor	-
6.7 Create Vendor Type	6.7 Create Vendor Type	-
6.8 Update Vendor Type	6.8 Update Vendor Type	-
6.9 View Vendor Types	6.9 View Vendor Types	-
6.10 Reactivate Vendor Type	6.10 Reactivate Vendor Type	-
6.11 Deactivate Vendor Type	6.11 Deactivate Vendor Type	-

Table 156 Vendor Subsystem Modifications

2.10.4.7 Project Management Subsystem

Project Management Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
7.1 Create Project	7.1 Create Project	-
7.2 View Project	7.2 View Project	-
7.3 Update Project	7.3 Update Project	-
7.4 Deactivate Project	7.4 Deactivate Project	-
7.5 Create Project Type	7.5 Create Project Type	-
7.6 View Project Type	7.6 View Project Type	-
7.7 Update Project Type	7.7 Update Project Type	-
7.8 Deactivate Project Type	7.8 Deactivate Project Type	-
7.9 Add Indemnity Form	7.9 Add Indemnity Form	-
7.10 Update Indemnity Form	7.10 Update Indemnity Form	-
7.11 View Indemnity Form	7.11 View Indemnity Form	-
7.12 Remove Indemnity Form	7.12 Remove Indemnity Form	-
7.13 Capture Indemnity Form	7.13 Capture Indemnity Form	-
7.14 Capture Project Feedback	7.14 Capture Project Feedback	-
7.15 View Project Feedback	7.15 View Project Feedback	-

Project Management Subsystem		
Use Case Name	Modification Change	Reason for Change
7.16 Create Impact Feedback Form	7.16 Create Impact Form	-
7.17 View Impact Feedback Form	7.17 View Impact Form	-
7.18 Update Impact Feedback Form	7.18 Update Impact Form	-
7.19 Deactivate Impact Feedback Form	7.19 Deactivate Impact Form	-
7.20 Capture Impact Feedback	7.20 Capture Impact Form	-
7.21 View Impact Feedback	7.21 View Impact Form	-
7.22 Add Photo	7.22 Add Photo	-
7.23 View Photo	7.23 View Photo	-
7.24 Delete Photo	7.24 Delete Photo	-
-	7.25 Add Photo Album	To organize photos into structured, manageable project collections.
-	7.26 View Photo Album	To organize photos into structured, manageable project collections.
-	7.27 Update Photo Album	To organize photos into structured, manageable project collections.
-	7.28 Remove Photo Album	To organize photos into structured, manageable project collections.

Table 157 Vendor Subsystem Modifications

2.10.4.8 K9 Subsystem

K9 Subsystem		
Use Case Name	Modification Change	Reason for Change
8.1 Add Dog	8.1 Add Dog	-
8.2 View Dog	8.2 View Dog	-
8.3 View Dogs	8.3 View Dogs	-
8.4 Update Dog	8.4 Update Dog	-
8.5 Deactivate Dog	8.5 Deactivate Dog	-
8.6 Assign User to Dog	8.6 Assign User to Dog	-
8.7 View User-Dog Assignment	8.7 View User-Dog Assignment	-
8.8 Update User-Dog Assignment	8.8 Update User-Dog Assignment	-
8.9 Reactivate User-Dog Assignment	8.9 Reactivate User-Dog Assignment	-
8.10 Deactivate User-Dog Assignment	8.10 Deactivate User-Dog Assignment	-
8.11 Add K9 Role	8.11 Add K9 Role	-
8.12 Update K9 Role	8.12 Update K9 Role	-

K9 Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
8.13 View K9 Role	8.13 View K9 Role	-
8.14 Reactivate K9 Role	8.14 Reactivate K9 Role	-
8.15 Deactivate K9 Role	8.15 Deactivate K9 Role	-
8.16 View Ancestry	8.16 View Ancestry	-

Table 158 K9 Subsystem Modifications

2.10.4.9 Rhino Subsystem

Rhino Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
9.1 Add Rhino	9.1 Add Rhino	-
9.2 View Rhino	9.2 View Rhino	-
9.3 View Rhinos	9.3 View Rhinos	-
9.4 Update Rhino	9.4 Update Rhino	-
9.5 Deactivate Rhino	9.5 Deactivate Rhino	-

Table 159 Rhino Subsystem Modifications

2.10.4.10 Elephant Subsystem

Elephant Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
10.1 Add Elephant	10.1 Add Elephant	-
10.2 View Elephant	10.2 View Elephant	-
10.3 View Elephants	10.3 View Elephants	-
10.4 Update Elephant	10.4 Update Elephant	-
10.5 Deactivate Elephant	10.5 Deactivate Elephant	-

Table 160 Elephant Subsystem Modifications

2.10.4.11 Message Subsystem

Message Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
11.1 Create Message	11.1 Create Message	-
11.2 View Messages	11.2 View Messages	-

Table 161 Message Subsystem Modifications

2.10.4.12 Report Subsystem

Report Subsystem		
<i>Use Case Name</i>	<i>Modification Change</i>	<i>Reason for Change</i>
12.1 Generate Custom Report	12.1 Generate Custom Report	-
12.2 View Custom Report	12.2 View Custom Report	-
12.3 Update Custom Report	12.3 Update Custom Report	-
12.4 Remove Custom Report	12.4 Remove Custom Report	-

12.5 View Bike Impact Report	12.5 View Bike Impact Report	-
12.6 View Budget Report	12.6 View Budget Report	-
12.7 View Rhino Report	12.7 View Rhino Report	-
12.10 View Elephant Report	12.10 View Elephant Report	-
12.9 View K9 Report	12.9 View K9 Report	-
12.10 View Machine Learning Dashboard	12.10 View Machine Learning Dashboard	-

Table 162: Report Subsystem Modifications

2.10.4.13 System Subsystem

System Subsystem		
Use Case Name	Modification Change	Reason for Change
13.1 Backup System	13.1 Backup System	-
13.2 Restore System	13.2 Restore System	-
13.3 View Audit Log	13.3 View Audit Log	-

Table 163: System Subsystem Modifications

2.10.4.14 Budget Subsystem

Budget Subsystem		
Use Case Name	Modification Change	Reason for Change
14.1 View Project Budget	14.1 View Project Budget	-
14.2 View Project Budgets	14.2 View Project Budgets	-
14.3 Log An Expense	14.3 Log An Expense	-
14.4 Update Expense	14.4 Update Expense	-
14.5 View Expense	14.5 View Expense	-
14.6 Remove Expense	14.6 Remove Expense	-

Table 164 Budget Subsystem Modifications

2.11 State Diagrams

2.11.1 Inventory State Diagram

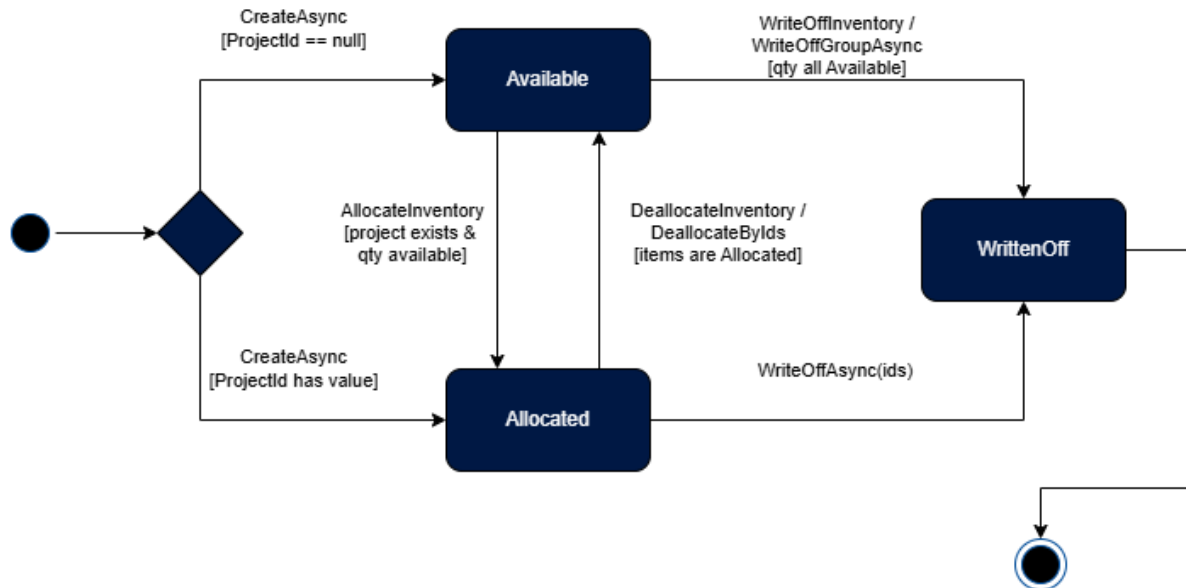


Figure 48: Inventory State Diagram

2.11.2 Deadline State Diagram

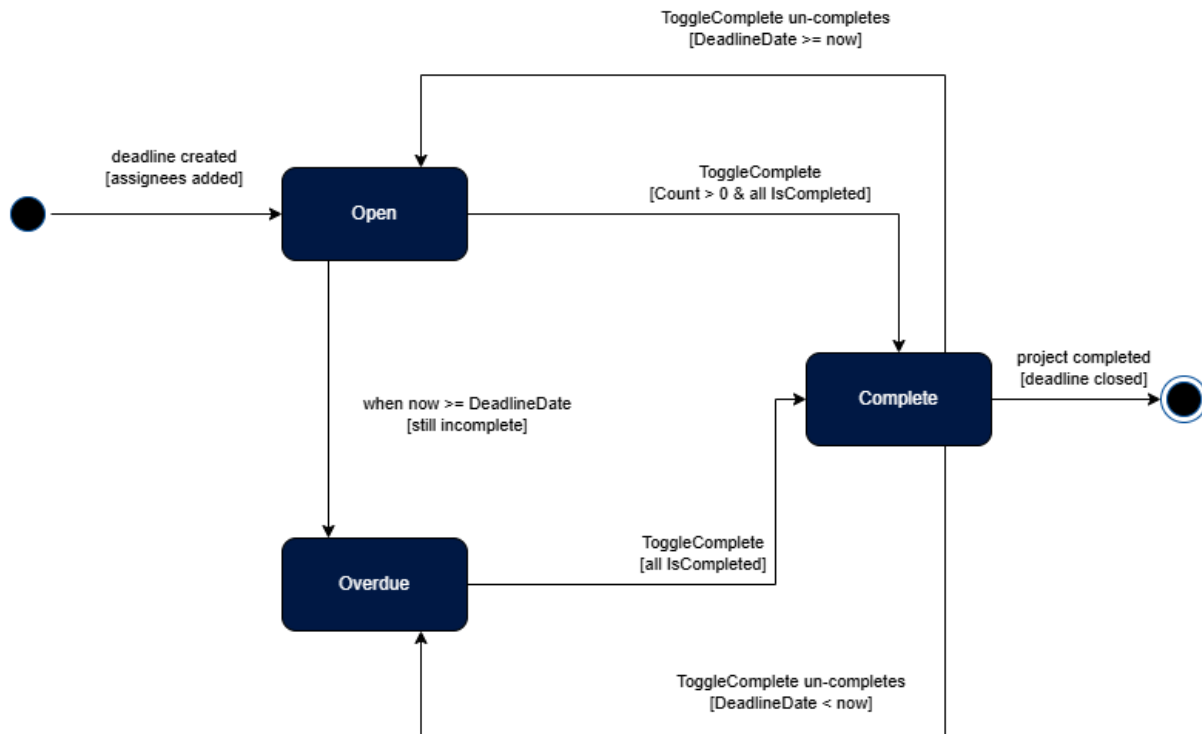


Figure 49: Deadline State Diagram

2.11.3 Animal State Diagram

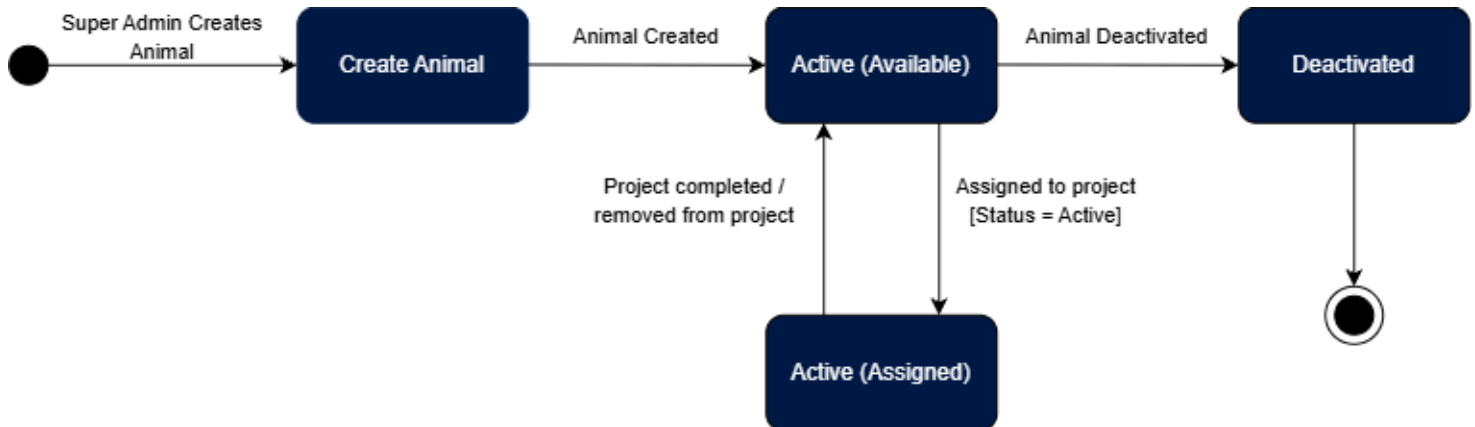


Figure 50: Animal State Diagram

3. Document Conclusion

The documentation presented showcased the screen designs for 13 uses assigned to Iteration 6. It also included technical narratives, technical primitive diagrams, technical context diagrams, sequence diagrams, a technical entity relationship diagram, a use case diagram, the updated functional requirement lists.

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: 

Date: 27/07/2026

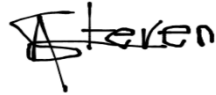
5. Sign-off by Team

Team member:

Name: Alexander Steven

Date: 30 July 2026

Signature:



Team member:

Name: Amahle Khoza

Date: 30 July 2026

Signature:



Team member:

Name: Christopher Bailey

Date: 30 July 2026

Signature:



Team member:

Name: Nicholas Bailey

Date: 30 July 2026

Signature:



Team member:

Name: Stephan Bredenhann

Date: 30 July 2026

Signature:

