Lab Final Report

Only for course Teacher						
		Needs Improvement	Developing	Sufficient	Above Average	Total Mark
Allocate mark & Percentage		25%	50%	75%	100%	25
Understanding/Analysis	7					
Implementation	8					
Report Writing	10					
					Total obtained mark	
Comments						

Semester: Spring 2025

Submission Date: 16/ 04 /2025

Course Code: SE 231 Course Name: System Analysis and Design Capstone Project

Course Teacher Name: Tahmina Meem

Designation: Lecturer, Department of Software Engineering

Group Member Details:

Student ID	Student Name
0242310005341424	Kamrujjaman Khan
0242310005341377	Zahed Ibna Morshed
0242310005341380	Md Hasibur Rahman
Section: A1 (40)	

Table of Contents

Introduction **6**

Objective *6*

Key Features *6*

Scenario writing: **8**

Scenario 1: User Registration and Authentication *8*

Scenario 2: User Role Management *8*

Scenario 3: Security & Data Privacy *8*

Scenario 4: Lost Item Reporting *8*

Scenario 5: Found Item Reporting *8*

Scenario 6: Lost and Found Item Matching *9*

Scenario 7: Claiming a Found Item *9*

Scenario 8: Proof of Ownership Submission *9*

Scenario 9: AI-Based Image Recognition for Matching *9*

Scenario 10: AI-Powered Fraud Detection *9*

Scenario 11: Smart Location-Based Alerts *10*

Scenario 12: Emergency Lost Item Alert *10*

Scenario 13: Chat or Contact System *10*

Scenario 14: Reward System for Finders *10*

Scenario 15: Feedback & Review System *11*

Scenario 16: Reporting Fake or Incorrect Listings *11*

Scenario 17: Admin Panel for Moderation *11*

Scenario 18: QR Code for Item Verification *11*

Stakeholder **12**

User Profile **12**

User Profile-01: Students *12*

User Profile-02: Faculty and Staff *13*

User Profile-03: University Administration *14*

Scope **15**

1. User Registration and Login *15*

2. User Role Management *15*

3. Lost Item Reporting *15*

5. Lost and Found Item Matching *16*
6. Claiming a Found Item *16*
7. Proof of Ownership Submission *16*
8. AI-Based Image Recognition for Matching *16*
9. AI-Powered Fraud Detection *17*
10. Smart Location-Based Alerts *17*
11. Emergency Lost Item Alert *17*
12. Security & Data Privacy *17*
13. Chat or Contact System *18*
14. Reward System for Finders *18*
15. Feedback & Review System *18*
16. Reporting Fake or Incorrect Listings *19*
17. Admin Panel for Moderation *19*
18. QR Code for Item Verification *19*

Feasibility Study **20**
1. Technical Feasibility *20*
2. Operational Feasibility *20*
3. Economic Feasibility *21*
4. Scheduling Feasibility *21*

Project Block Diagram **22**
Block Diagram - 1: *22*
Block Diagram - 2: *23*
Block Diagram - 3: *24*

Software Requirement Specification (SRS) **25**

Use Case Diagram **29**

User Case Description **30**
Case Description-01: Register Account *30*
Case Description-02: Verify Email or Phone *31*
Case Description-03: Login *32*
Case Description-04: Verify Credentials *33*
Case Description-05: Reset Password *34*
Case Description-06: Report Lost Item *35*

Case Description-07: Upload Item Details *36*
Case Description-08: Flag as Important *37*
Case Description-09: Edit Reported Item *38*
Case Description-10: Claim Lost Item *39*
Case Description-11: Verify Ownership *40*
Case Description-12: Reject Claim *41*
Case Description-13: Request Contact with Finder *42*
Case Description-14: Send Message *43*
Case Description-15: Receive Reward Points *44*
Case Description-16: Earn Badges *45*
Case Description-17: Submit Feedback *46*
Case Description-18: Rate Experience *47*
Case Description-19: View Feedback History *48*
Case Description-20: Report Fake Listing *49*
Case Description-21: Submit Complaint *50*
Case Description-22: Review Complaint *51*
Case Description-23: Take Action *52*
Case Description-24: Manage Users *53*
Case Description-25: Monitor Reports & Feedback *54*
Case Description-26: Take Disciplinary Action *55*
Case Description-27: Suspend User *56*
Case Description-28: Generate QR Code for Found Item *57*
Case Description-29: Scan QR Code *58*
Case Description-30: Review Item Details *59*
Case Description-31: Logout *60*

System Design **61**

Data Flow Diagram **61**

Data Flow Diagram Level 0: *61*
Data Flow Diagram Level 1: *62*
Data Flow Diagram Level 2: *63*

Activity Diagram **64**

FR01 - User Registration and Authentication *64*
FR02 - User Role Management *65*

FR03 - Security & Data Privacy 66
FR04 - Lost Item Reporting 67
FR05 - Found Item Reporting 68
FR06 - Lost and Found Item Matching 69
FR07 - Claiming a Found Item 70
FR08 - Proof of Ownership Submission 71
FR09 - AI-Based Image Recognition for Matching 72
FR10 - AI-Powered Fraud Detection 73
FR11 - Smart Location-Based Alerts 74
FR12 - Emergency Lost Item Alert 75
FR 13 - Chat or Contact System 76
FR 14 – Reward System for Finders 78
FR 15 – Feedback and Review System 79
FR 16 – Reporting Fake or Incorrect Listings 80
FR 17 – Admin Panel for Moderation 81
FR 18 – QR Code for Item Verification 82

Sequence Diagram **83**
1. User Registration and Login 83
2. User Role Management 84
..... 84
3. Lost Item Reporting 85
5. Lost and Found Item Matching 87
6. Claiming a Found Item 88
7. Proof of Ownership Submission 89
8. AI-Based Image Recognition for Matching 90
9. AI-Powered Fraud Detection 91
10. Smart Location-Based Alerts 92
11. Emergency Lost Item Alert 93
12. Security & Data Privacy 94
13. Chat or Contact System 95
14. Reward System for Finders 96
15. Feedback & Review System 97

Class Diagram **98**

Entity Relationship Diagram **99**

Prototype **100**

Introduction

Students within academic facilities frequently lose their personal items such as ID cards and wallets and books as well as electronic devices, even sometimes they might lose something very important like passport or legal documents or bank cards. Students as well as faculty members and staff members all can benefit if there is an efficient security-centered easy-to-use Lost and Found System which lets them submit reports and locate missing things. The system delivers a smooth operation through its integration of AI-powered item matching together with security measures and role-based management features to satisfy users. The system combines modern technologies to improve campus safety and provide clear visibility regarding lost item management and responsible monitoring.

Objective

The Lost and Found System at academic institutions has its main purpose in optimizing reporting and recovery procedures for lost items. The system aims to offer students and faculty an encrypted method to submit reports about lost and found items. The system will use AI item matching to enhance recovery efficiency. The system ensures safe protection of all stored user data. This system enables users to maintain contact with the person who retrieved the item and the person claiming it. The platform aids in preventing fraudulent claims while it enhances accountability regarding item receipt procedures. The system will enhance student and faculty trust on campus security by improving entire-campus participation with the lost and found operations.

Key Features

1. User Registration and Authentication: Secure registration using university credentials. Email verification with OTP or activation link.

2. User Role Management: User roles: Students, Faculty, Staff, Admin. Admins oversee moderation and dispute resolution.

3. Security & Data Privacy: User data encryption. Secure session management. Limited access to personal details.

4. Lost Item Reporting: Users report lost items with details and images. Items can be edited or deleted before a claim is made.

5. Found Item Reporting: Users report found items with descriptions and images. Items can be edited or deleted before a claim is made.

6. Lost and Found Item Matching: AI-powered matching based on keywords and images. Automatic notifications for potential matches.

7. Claiming a Found Item: Users request to claim a found item. Finders can accept or reject claims. Admin intervention in case of disputes.

8. Proof of Ownership Submission: Users upload receipts or images to verify ownership. Finder or Admin verifies claims before approval.

9. AI-Based Image Recognition for Matching: AI scans uploaded images to find visually similar items.

10. AI-Powered Fraud Detection: flags suspicious listings to prevent false claims.

11. Smart Location-Based Alerts: Notifications sent when users are near a lost item's last known location.

12. Emergency Lost Item Alert: Urgent items receive priority visibility and notifications.

13. Chat or Contact System: Secure messaging between finders and claimants.

14. Reward System for Finders: Return of lost items earns users both points and badges as part of their rewards system.

15. Feedback & Review System: Users need to evaluate their satisfaction level after retrieving the misplaced item.

16. Reporting Fake or Incorrect Listings: Users of the platform can report any incorrect or misleading content throughout the platform. The administrators check the information and perform suitable actions.

17. Admin Panel for Moderation: The system provides admins with full control to oversee user management together with listing administration. - Access reports and analytics.

18. QR Code for Item Verification: The system creates QR codes for verifying found items through claims processing.

Scenario writing:

Scenario 1: User Registration and Authentication

Description: A student wants to register in the Lost and Found System using their university credentials.

1. The student visits the login page and clicks on "Register."
2. The system prompts the students to enter their university email and password.
3. After verification, the student completes the registration by providing additional details.
4. The system sends a confirmation email.
5. Upon confirmation, the student can log in to the system.

Scenario 2: User Role Management

Description: The admin assigns different roles to users for access control.

1. The admin logs into the system.
2. They navigate to the "User Management" section.
3. The admin selects a user and assigns them a role (Student, Faculty, Staff).
4. The system updates the user's permissions accordingly.

Scenario 3: Security & Data Privacy

Description: The system ensures secure data storage and restricted access.

1. A faculty member logs in and tries to access another user's lost item report.
2. The system restricts access to only their own reports or admin-approved views.
3. Sensitive data is encrypted and stored securely.

Scenario 4: Lost Item Reporting

Description: A student reports a lost item in the system.

1. The student logs into the system and selects "Report Lost Item."
2. They enter details (category, description, last seen location, and an image).
3. The system saves the lost item report and searches for possible matches.
4. The student receives a confirmation notification.

Scenario 5: Found Item Reporting

Description: A faculty member finds a lost item and reports it in the system.

1. The faculty member logs in and selects "Report Found Item."
2. They upload an image, enter a description, and specify where they found the item.

3. The system records the found item and checks for matching lost item reports.
4. A notification is sent to potential owners.

Scenario 6: Lost and Found Item Matching

Description: The system automatically matches lost items with reported found items.

1. A lost item report is added to the system.
2. The AI-powered system scans all found item reports.
3. A matching item is detected based on description and images.
4. The system notifies the user about a potential match.

Scenario 7: Claiming a Found Item

Description: A student claims a found item that matches their lost report.

1. The student receives a notification about a matching found item.
2. They navigate to the "Claim Item" section and submit a claim request.
3. The system asks for verification details.
4. The admin reviews the request and either approves or denies it.

Scenario 8: Proof of Ownership Submission

Description: A user submits proof of ownership before claiming an item.

1. A student finds their lost item listed as found and submits a claim.
2. The system asks for supporting proof (images, receipts, or descriptions).
3. The admin reviews the proof.
4. If approved, the student is given instructions for collecting the item.

Scenario 9: AI-Based Image Recognition for Matching

Description: The system uses AI to detect similar lost and found items.

1. A staff member uploads a found item image.
2. The AI scans the database for visually similar lost item images.
3. The system suggests potential matches.
4. The admin or the original owner is notified.

Scenario 10: AI-Powered Fraud Detection

Description: The system flags suspicious claims for manual verification.

1. A user tries to claim an item that does not match their past reports.
2. The AI detects inconsistencies and flags the request.
3. The admin reviews the flagged claim.
4. If fraudulent, the claim is rejected, and a warning is issued.

Scenario 11: Smart Location-Based Alerts

Description: The system notifies users when they are near the last known location of their lost item.

1. A student reports losing their bag in a lecture hall.
2. The system tracks users who are near that location.
3. Nearby users receive a notification requesting them to report if they find the bag.

Scenario 12: Emergency Lost Item Alert

Description: The system prioritizes critical lost items such as wallets, keys, or IDs.

1. A student reports losing their university ID card.
2. The system recognizes the item as high priority.
3. An immediate alert is sent to admins and nearby users.
4. The admin prioritizes the case for faster resolution.

Scenario 13: Chat or Contact System

Description: Secure messaging enables both finders and claimants to talk to each other without needing to reveal personal information.

1. The user chooses an item for claiming status.
2. Users can engage in secure communication through the system between both the finder and the claimant.
3. The discussion about item ownership continues between finder and claimant while both parties avoid exchanging their personal contact information.
4. The system enables security supervision through admin access to monitor interactions.

Scenario 14: Reward System for Finders

Description: Those who bring lost possessions back to owners receive points and badges in the system.

1. User reports a found item.
2. The system enables the owner to claim their lost item through verification steps.
3. The system provides points or badges to any person who successfully finds a lost item.
4. Users have access to view all their obtained perks through the platform.

Scenario 15: Feedback & Review System

Description: The system receives feedback together with ratings from users after they recover their lost items to make improvements. The user achieves success in retrieving lost items through the system.

1. The system requests users to share their feedback with the system.
2. Users evaluate their interaction experience then the system requests their feedback remarks.
3. The system's performance receives feedback assessment from admin personnel for ultimate system improvement.

Scenario 16: Reporting Fake or Incorrect Listings

Description: Users can submit questionable lost item listings to admin reviewers who will take necessary action.

1. The user encounters deceptive reports that turn out to be inaccurate.
2. A user can present a complaint that includes specific information about the issue.
3. The report gets reviewed by Admin who verifies its authenticity.
4. The offender will face warnings and banning alongside possible suspension imposed by the admin.

Scenario 17: Admin Panel for Moderation

Description: The administrative control center allows managers to handle user accounts while keeping tabs on posted items and resolve feedback reports from one location.

1. Admin logs into the system.
2. The administrator checks reported listings together with user activities through the system.
3. The administrator reviews reported items before deciding the appropriate course of action.
4. The system allows administrators to generate reports while executing analytics monitoring.

Scenario 18: QR Code for Item Verification

Description: Users need to scan the QR codes printed on found items then confirm the ownership before making a claim.

1. Users can recover reported lost items by scanning the attached QR code.
2. The user activates the scanning of the QR code through the system interface.
3. The system obtains item information from its database and confirms whether the piece is original.
4. The user can continue forward with item claim if the system verifies correct details.

Stakeholder

1. **Students:** Students are responsible for both submitting reports about lost items and retrieving their claimed possessions.
2. **Faculty & Staff:** Members of both faculty and staff must report and claim lost items while helping track lost items.
3. **University Administration (Admin):** University Administration holds responsibility for system supervision and management tasks.
4. **System Administrator:** System Administrators sustain the database while performing tasks to guarantee operational efficiency.

User Profile

User Profile-01: Students

User Class	Notes on Characteristic	Requirement Implied
Type of User	Student	Verification
Age Range	18-25	Verification
Frequency of Use	Occasionally, when they lose or find items	Performance, Operation, Acceptance
Mandatory	No	-
Computer Experience	Basic to Moderate	Documentation
Education	Undergraduate, Graduate	-
Goals	Report lost and found items, claim lost items	Resource, Performance, Security, Acceptance, Operation
Language Skills	Bangla, English	-
Number of Users	10000+	Performance, Operation, Acceptance, Portability
Training	Not required	-
Other System Used	Social media, University Portals	-
Ways of Working	Expect a simple, user-friendly system	Acceptance, Safety, Security, Operation, Maintenance, Portability

User Profile-02: Faculty and Staff

User Class	Notes on Characteristic	Requirement Implied
Type of User	Faculty, Staff	Verification
Age Range	25-55	Verification
Frequency of Use	Occasionally, mainly for reporting or resolving issues	Performance, Operation, Acceptance
Mandatory	No	-
Computer Experience	Moderate to Advanced	Documentation
Education	Master's, PhD (Faculty), Graduate (Staff)	-
Goals	Report lost and found items, assist students if needed	Resource, Performance, Security, Acceptance, Operation
Language Skills	Bangla, English	-
Number of Users	200-500	Performance, Operation, Acceptance, Portability
Training	May require basic system guidance	Documentation
Other System Used	University Portals, Email	-
Ways of Working	Expect a structured, efficient reporting system	Acceptance, Safety, Security, Operation, Maintenance, Portability

User Profile-03: University Administration

User Class	Notes on Characteristic	Requirement Implied
Type of User	University Admin	Verification
Age Range	30-55	Verification
Frequency of Use	Daily, for moderation and management	Performance, Operation, Acceptance
Mandatory	Yes	-
Computer Experience	Advanced	Documentation
Education	Master's, PhD	-
Goals	Manage user accounts, resolve disputes, maintain system security	Resource, Performance, Security, Acceptance, Operation
Language Skills	Bangla, English	-
Number of Users	10-50	Performance, Operation, Acceptance, Portability
Training	May accept training, but prefer minimal learning curve	Documentation
Other System Used	University Portals, Admin Dashboards	-
Ways of Working	Require high control and security features	Acceptance, Safety, Security, Operation, Maintenance, Portability

Scope

1. User Registration and Login

a. Objective:

To allow users to register and log in to the system securely.

b. Functionality:

- Users will provide necessary information for registration.
- Secure authentication mechanisms will be implemented for user logins.
- Password recovery options will be available.

2. User Role Management

a. Objective:

To assign different roles to users based on their access permissions.

b. Functionality:

- Admins can assign and manage roles for students, faculty, staff, and other users.
- Different access levels will be enforced to ensure system security.

3. Lost Item Reporting

a. Objective:

Enable users to report lost items efficiently.

b. Functionality:

- Users can submit details such as item category, description, last known location, and an image.
- The system will store lost item reports for matching and retrieval.

4. Found Item Reporting

a. Objective:

Allow users to report items they have found.

b. Functionality:

- Users can submit details of the found item, including location, description, and images.
- The system will store found item reports and attempt to match them with lost items.

5. Lost and Found Item Matching

a. Objective:

To automate the process of matching lost and found items using AI.

b. Functionality:

- AI algorithms will scan reported lost and found items to find potential matches.
- The system will notify users when a possible match is found.

6. Claiming a Found Item

a. Objective:

Enable users to claim found items through a structured process.

b. Functionality:

- Users can submit a claim request for an item.
- The system will require proof of ownership before approval.

7. Proof of Ownership Submission

a. Objective:

Ensure that only the rightful owner can claim a found item.

b. Functionality:

- Users must submit proof such as receipts, images, or item descriptions.
- Admins will review and approve or reject claims based on submitted proof.

8. AI-Based Image Recognition for Matching

a. Objective:

Enhance lost and found item matching through AI-powered image recognition.

b. Functionality:

- The system will scan uploaded images and compare them with found item reports.
- Matches will be suggested to potential owners.

9. AI-Powered Fraud Detection

a. Objective:

Prevent fraudulent claims and duplicate reports.

b. Functionality:

- AI will flag suspicious or duplicate claims.
- Admins will manually review flagged reports before approval.

10. Smart Location-Based Alerts

a. Objective:

Notify users when they are near the last known location of a lost item.

b. Functionality:

- Users who pass near a reported lost item location will receive notifications.
- Alerts will be triggered based on proximity to the reported location.

11. Emergency Lost Item Alert

a. Objective:

Prioritize urgent lost items such as wallets, keys, and university ID cards.

b. Functionality:

- The system will categorize critical lost items as high priority.
- Urgent alerts will be sent to admins and nearby users.

12. Security & Data Privacy

a. Objective:

Ensure the security and privacy of user data and reports.

b. Functionality:

- Secure authentication and encrypted data storage will be implemented.
- Access to reports and personal details will be restricted based on user roles.

13. Chat or Contact System

a. Objective:

The system should allow loss finders to securely contact claimants for lost item recovery purposes.

b. Functionality:

- End-to-end encrypted messaging system.
- Every user of the platform can both send and receive private messages.
- The platform provides administrators with the ability to watch for unsuitable text or images.
- Option to block/report users in case of suspicious behavior.

14. Reward System for Finders

a. Objective:

The system will motivate users to bring back missing items through a points and badges reward system.

b. Functionality:

- The process of recovering lost items awards users with points that they can earn as finders.
- Users receive awards through badges when they reach specific item return milestones starting from 5 items up to 20 items.
- Users have access to see all their previously received rewards.
- Through the reward system administrators maintain full authority to modify reward parameters as well as to reset them altogether.

15. Feedback & Review System

a. Objective:

The system includes a feedback option whereby users can describe their experience after finding lost items to enhance system reliability.

b. Functionality:

- The application enables users to give ratings through a system of five stars.
- Additional feedback occurs through optional text reviews.
- The system allows administrative personnel to analyze feedback for making system improvements.
- Reviews visible on the claim history page.

16. Reporting Fake or Incorrect Listings

a. Objective:

The system protects its integrity by enabling users to submit reports about deceitful and inaccurate lost and found reports.

b. Functionality:

- Users enjoy the ability to report all listings containing fake information.
- Users can submit their reports into specific divisions such as spam or wrongful claims and duplicate content.
- User Administered reports lead to needed actions where admins either accept or decline the content alongside full deletion possibilities.
- All users will receive information about actions taken in the system.

17. Admin Panel for Moderation

a. Objective:

The system will equip administrators with capabilities to oversee users, listings, reports together with system analytics.

b. Functionality:

- The system enables administrator users to review listing records for lost items as well as their management functions.
- The system provides administrators with tools that allow them to block or prevent access for users carrying out deceitful actions.
- The system offers administrative dashboards which give reports about system utilization patterns alongside notification of flagged content.
- Log history of administrative actions.

18. QR Code for Item Verification

a. Objective:

The system will enable rightful claimants to retrieve lost property by verifying each item through exclusive QR codes.

b. Functionality:

- The system generates individual QR codes for every found object.
- Users need to scan the QR codes to prove they are rightful owners of the items.
- The system identifies the scanned QR code reference against stored database records.
- Administrators can reset QR codes and assign new ones to different items.

Feasibility Study

The Lost and Found System for Students presents a secure platform which enhances the efficiency of lost item tracking and reporting and item retrieval processes in academic organizations. The study examines the technical capability as well as operational efficiency and economic value and scheduling requirements to ensure productive system deployment.

1. Technical Feasibility

a. Hardware Compatibility:

The Lost and Found System provides support for operation through desktop platforms as well as mobile platforms. The analysis evaluates whether the system matches existing university technological infrastructure that includes portals and administrative systems to achieve flawless integration.

b. Software Compatibility:

The system development team will build compatibility for mainstream operating systems together with common web browsers. The system will integrate with university authentication services including student ID login systems for safe data management and authentication evaluation.

c. Technical Expertise:

The system demands developers who possess talents in web technology and its related databases. A technical evaluation will confirm that both internal university IT employees and outside developers possess the qualifications required to support system maintenance and remedy technical issues. institutional administration members as well as students must participate in organized training programs to use the system effectively.

2. Operational Feasibility

a. User Acceptance:

The system development will target users from both student and faculty groups along with university administration. Ease of use and accessibility testing through surveys and usability tests will assess the system. The system development team will adapt secure messaging features alongside QR code verification and reporting tools by incorporating client feedback.

b. Impact on Current Operations:

The system operates to eliminate the conventional paper-based approaches of lost and found management. The system will use a step-by-step rollout plan to reduce interruptions while establishing support services for user assistance during implementation.

3. Economic Feasibility

a. Cost-Benefit Analysis:

Assessment of system development expenses and support costs alongside advantages including better return rates for lost items and less paperwork and simplified student use will be conducted for decision-making. Research will determine both university budget availability and possibilities for securing sponsorships as possible funding options.

b. Return on Investment (ROI):

The projected return on investment will take into consideration the increased efficiency along with lesser valuable item loss and possible workforce reduction. The assessment will evaluate supplementary advantages stemming from increased student satisfaction together with a stronger institutional reputation.

4. Scheduling Feasibility

a. Project Timeline:

A system development schedule will incorporate phases spanning requirement analysis followed by system development then testing afterward the deployment stage and user training phase. The project will incorporate safeguard provisions to address possible delays that emerge from system integration problems.

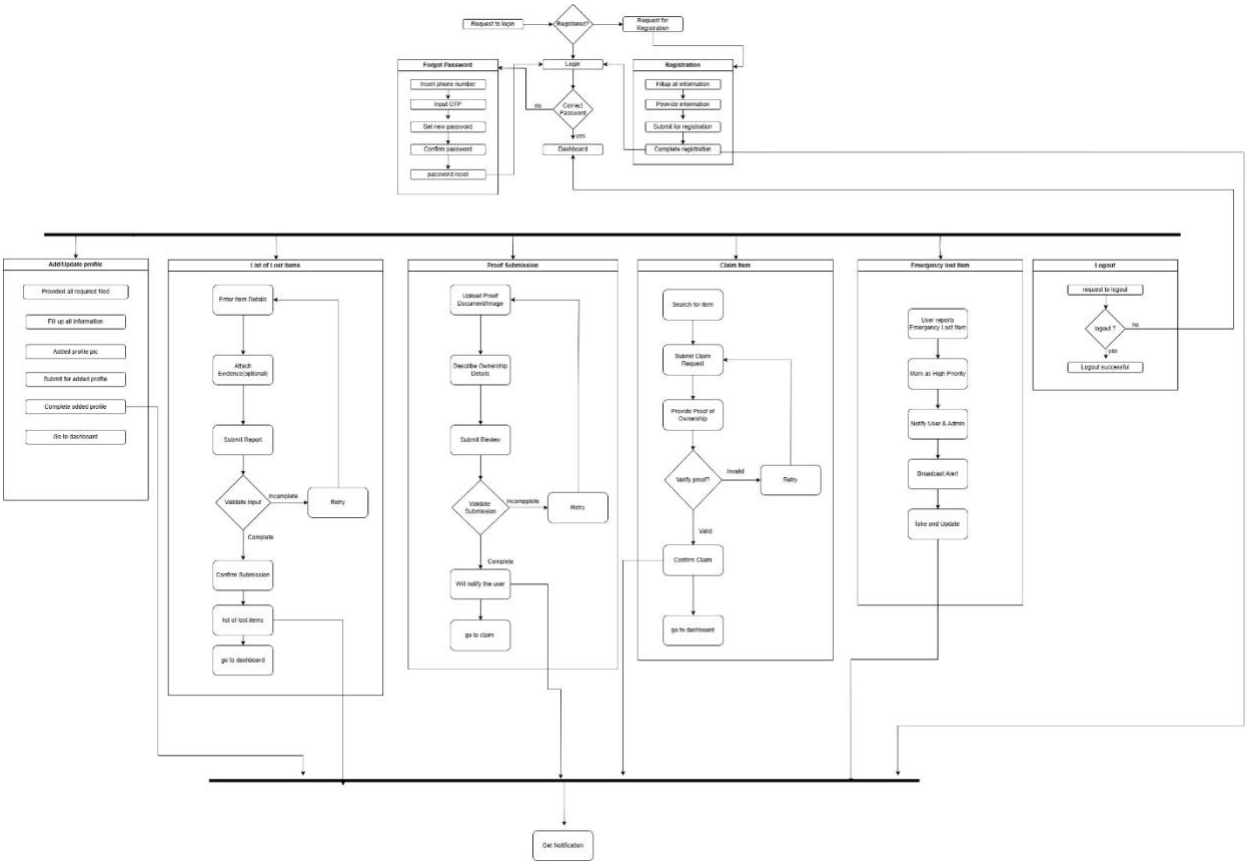
b. Dependencies:

Successful deployment depends on multiple outside factors including university IT readiness qualifications and approval oversight for student data systems and requirements to connect with external party systems such as QR generator programs and communication platforms. All dependent elements of the project will find their place in the project plan.

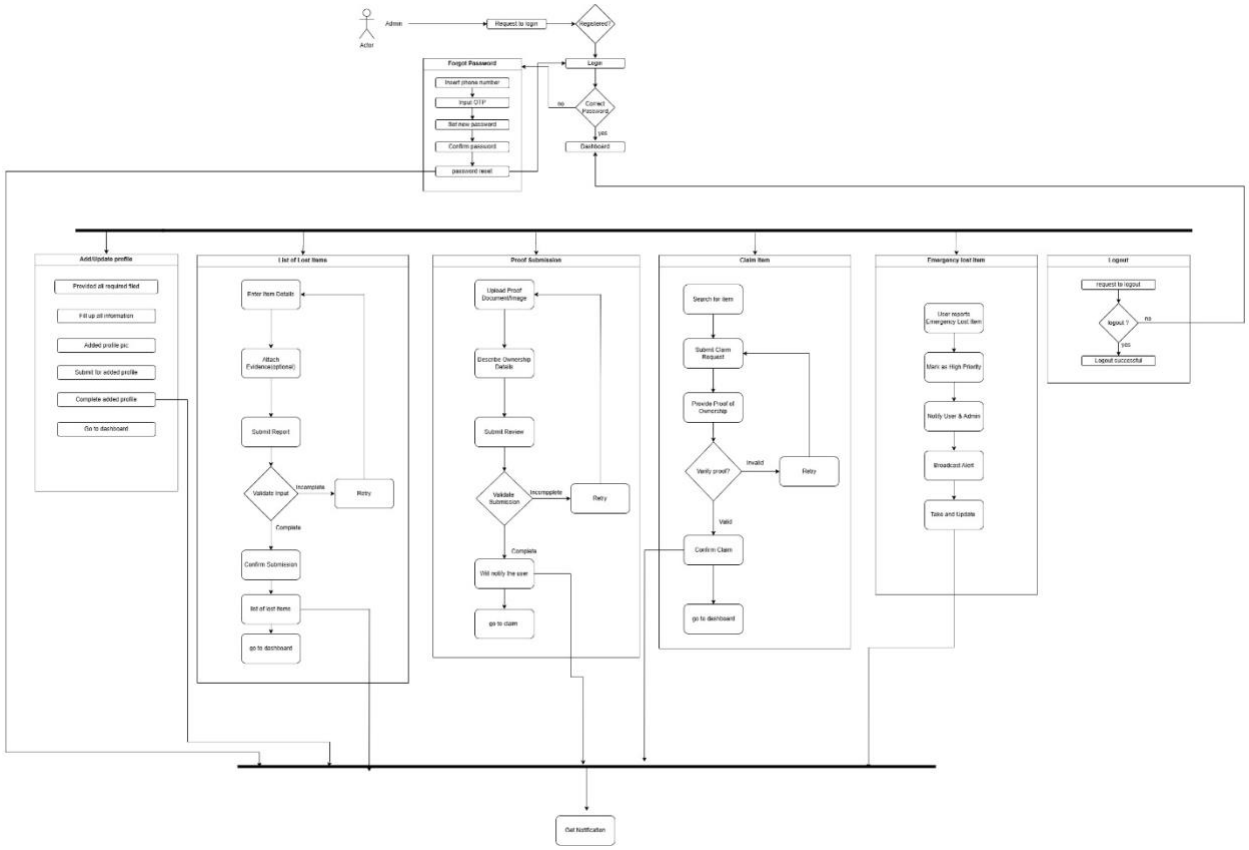
The evaluation determines that the Student Lost and Found System represents an operationally viable solution that is both technically and economically valid as well as time-planned feasible. The system has several advantages which consist of better lost item tracking processes along with improved student involvement and greater security measures. The development expenses along with implementation costs yield efficient operations and active system use into the long term. The project implementation plan will integrate risk reduction procedures including staged deployments together with training sessions and ongoing feedback gathering to achieve project success.

Project Block Diagram

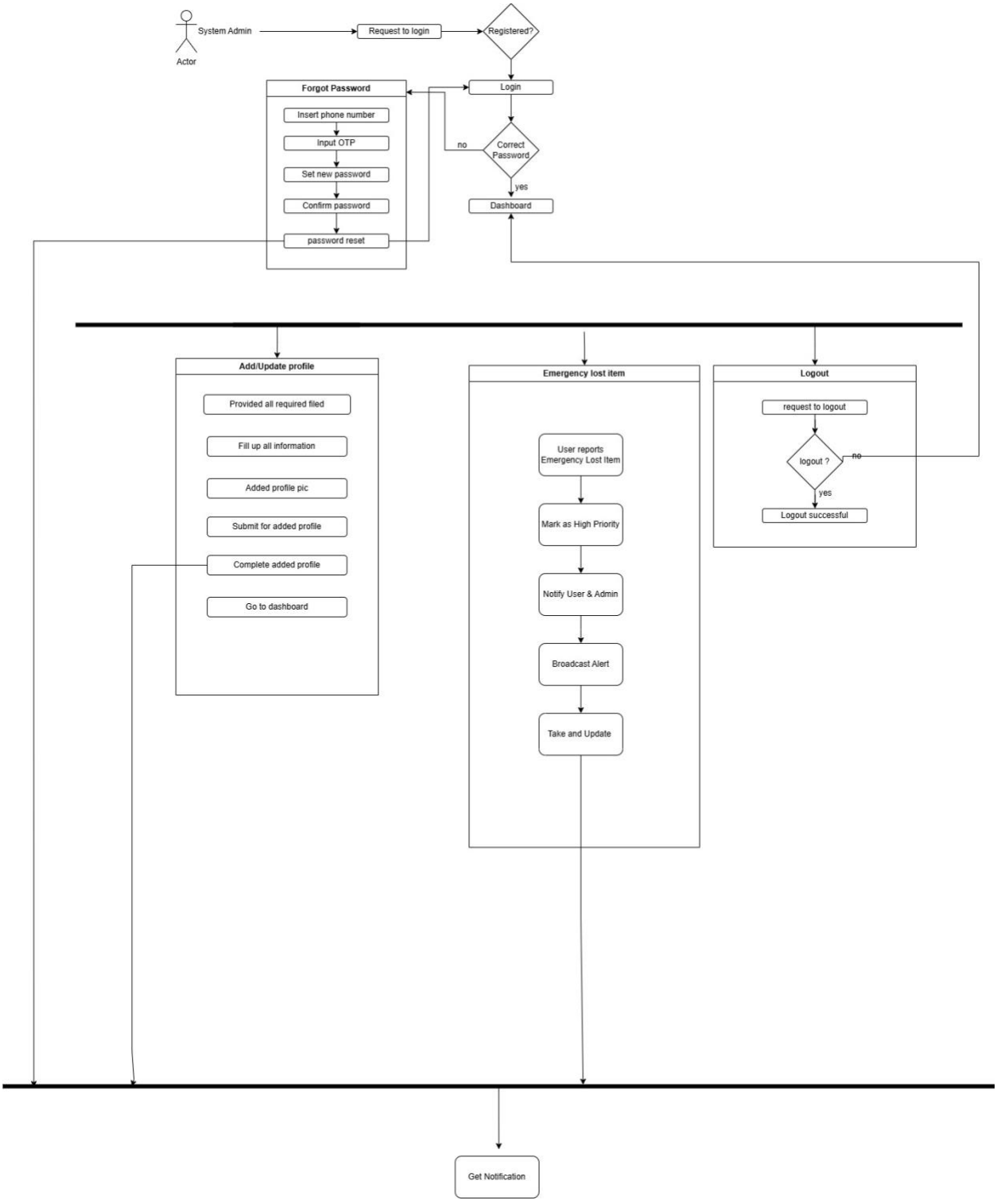
Block Diagram - 1:



Block Diagram - 2:



Block Diagram - 3:



Software Requirement Specification (SRS)

FR01	**User Registration and Authentication**
Description	Users must register using institutional credentials before accessing the system.
Stakeholder	Students, Faculty, Staff, Admin

FR02	**User Role Management**
Description	Admins assign different roles (Student, Faculty, Staff) with varying permissions.
Stakeholder	Admin

FR03	**Security & Data Privacy**
Description	The system ensures data encryption and restricted access to user information.
Stakeholder	Students, Faculty, Staff, Admin

FR04	**Lost Item Reporting**
Description	Users can report lost items by providing details such as category, location, and images.
Stakeholder	Students, Faculty, Staff

FR05	**Found Item Reporting**
Description	Users can report found items by submitting relevant details and images.
Stakeholder	Students, Faculty, Staff

FR06	**Lost and Found Item Matching**
Description	AI automatically matches lost and found items based on descriptions and images.
Stakeholder	Students, Faculty, Staff

FR07	**Claiming a Found Item**
Description	Users can submit a claim request to retrieve a found item.
Stakeholder	Students, Faculty, Staff, Admin

FR08	**Proof of Ownership Submission**
Description	Users must submit proof of ownership (images, receipts, or detailed descriptions) before claiming a found item.
Stakeholder	Students, Faculty, Staff, Admin

FR09	**AI-Based Image Recognition for Matching**
Description	AI scans uploaded images to detect similarities between lost and found items.
Stakeholder	System Administrator

FR10	**AI-Powered Fraud Detection**
Description	AI flags suspicious or duplicate listings to prevent fraudulent claims.
Stakeholder	University Admin, System Administrator

FR11	**Smart Location-Based Alerts**
Description	Users receive notifications when they are near the last known location of a lost item.
Stakeholder	Students, Faculty, Staff

FR12	Emergency Lost Item Alert
Description	High-priority lost items (wallets, keys, IDs) receive urgent alerts for faster recovery.
Stakeholder	Students, Faculty, Staff, Admin

FR13	Chat or Contact System
Description	Secure messaging between finders and claimants for safe communication.
Stakeholder	Students, Faculty, Staff, Admin

FR14	Reward System for Finders
Description	Users earn points and badges for successfully returning lost items to their rightful owners.
Stakeholder	Students, Faculty, Staff, Admin

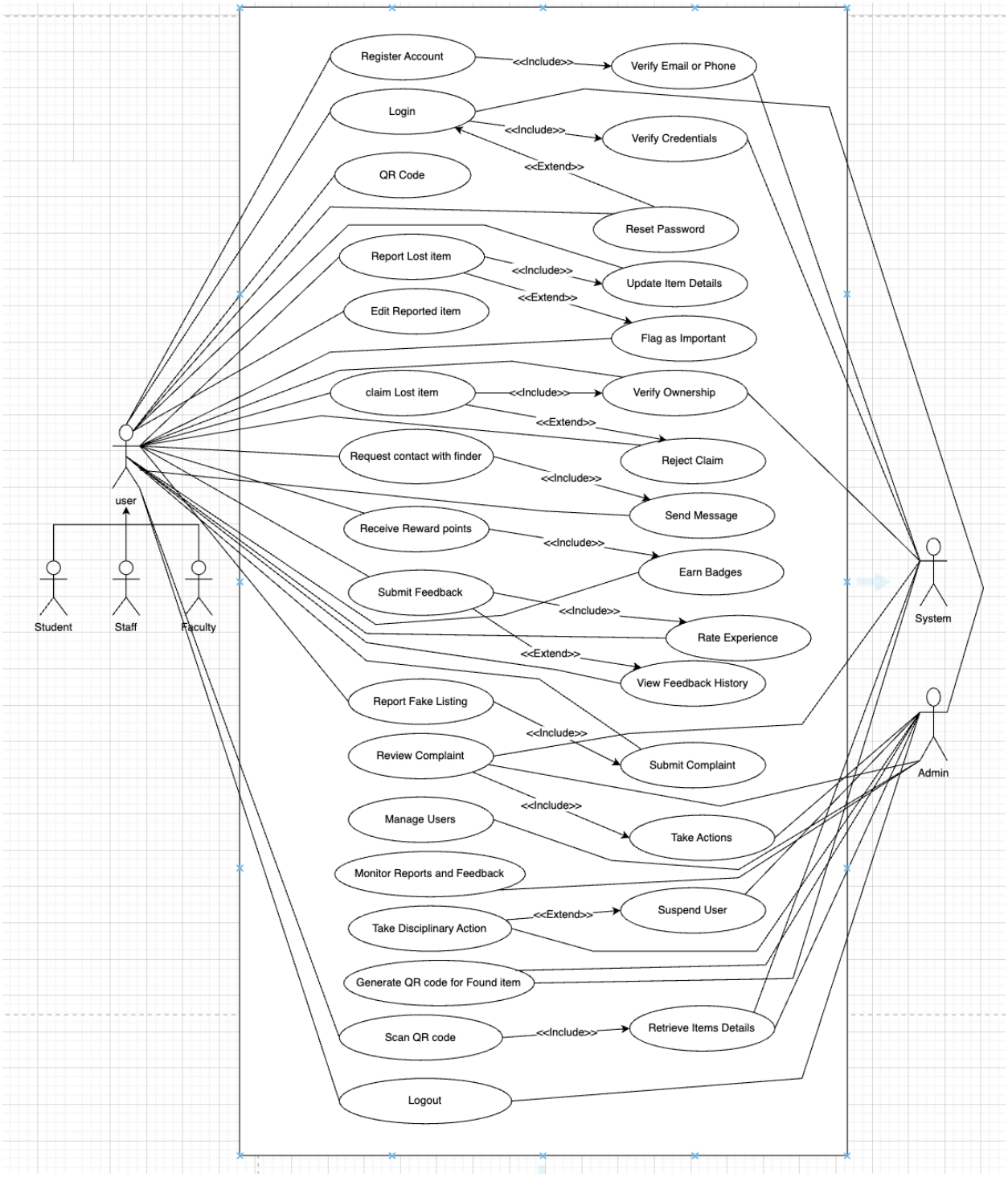
FR15	Feedback & Review System
Description	Users can rate and provide feedback on their experience after retrieving a lost item.
Stakeholder	Students, Faculty, Staff

FR16	Reporting Fake or Incorrect Listings
Description	Users can report false or misleading lost item listings, which will be reviewed by admins.
Stakeholder	Students, Faculty, Staff, Admin

FR17	Admin Panel for Moderation
Description	Admins can manage users, monitor listings, review reports, and access system analytics.
Stakeholder	Admin

FR18	**QR Code for Item Verification**
Description	A unique QR code is generated for found items to verify ownership claims efficiently.
Stakeholder	Students, Faculty, Staff, Admin

Use Case Diagram



User Case Description

Case Description-01: Register Account

<table>
<tr><td>Use Case</td><td>Register Account</td></tr>
<tr><td>Goal</td><td>Users can create a new account.</td></tr>
<tr><td>Precondition</td><td>Users must not have an existing account.</td></tr>
<tr><td>Success End Condition</td><td>Notification: "Registration Successful"</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Registration Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>The user initiates the account registration.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Press "Register" button</td></tr>
<tr><td>2.</td><td>Provide registration details</td></tr>
<tr><td>3.</td><td>Verify Email or Phone</td></tr>
<tr><td>4.</td><td>Account Created</td></tr>
<tr><td>5.</td><td>Notification: "Registration Successful"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again</td></tr>
<tr><td>2.1</td><td>Email/Phone Verification Failed</td></tr>
<tr><td></td><td>2.1.a Retry verification.</td></tr>
<tr><td>3.1</td><td>User enters incorrect code</td></tr>
<tr><td></td><td>3.1.a System prompts retry.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Users must complete registration within 15 minutes.</td></tr>
</table>

Case Description-02: Verify Email or Phone

<table>
<tr><td>Use Case</td><td>Verify Email or Phone</td></tr>
<tr><td>Goal</td><td>Ensure user-provided email or phone number is valid for account registration or recovery.</td></tr>
<tr><td>Precondition</td><td>User has initiated the registration or password reset process.</td></tr>
<tr><td>Success End Condition</td><td>User's email or phone number is verified, allowing the process to continue.</td></tr>
<tr><td>Failed End Condition</td><td>Verification fails due to incorrect or unresponsive contact details.</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

System</td></tr>
<tr><td>Trigger</td><td>User enters an email or phone number and requests verification.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User inputs an email or phone number.</td></tr>
<tr><td>2.</td><td>System sends a verification code to the provided contact.</td></tr>
<tr><td>3.</td><td>User enters the received code.</td></tr>
<tr><td>4.</td><td>System verifies the code.</td></tr>
<tr><td>5.</td><td>User proceeds with registration or password reset.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>User enters incorrect details</td></tr>
<tr><td></td><td>1.1.a System prompts re-entry.</td></tr>
<tr><td>2.1</td><td>User doesn’t receive code</td></tr>
<tr><td></td><td>2.1.a System allows resending the code.</td></tr>
<tr><td>3.1</td><td>User enters incorrect code</td></tr>
<tr><td></td><td>3.1.a System prompts retry.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Fast and reliable message delivery, proper validation, secure handling of user data.</td></tr>
</table>

Case Description-03: Login

<table>
<tr><td>Use Case</td><td>Login</td></tr>
<tr><td>Goal</td><td>Users can enter the system by logging in.</td></tr>
<tr><td>Precondition</td><td>Users must have a registered account.</td></tr>
<tr><td>Success End Condition</td><td>Notification: "Login Successful"</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Login Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>The user initiates the login process.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Press "Login" button</td></tr>
<tr><td>2.</td><td>Provide login details</td></tr>
<tr><td>3.</td><td>Enter user ID and password</td></tr>
<tr><td>4.</td><td>Verify Credentials</td></tr>
<tr><td>5.</td><td>Notification: "Login Successful"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again</td></tr>
<tr><td>2.1</td><td>Server Not Found</td></tr>
<tr><td></td><td>2.1.a Try Again Later!</td></tr>
<tr><td>3.1</td><td>Incorrect Credentials</td></tr>
<tr><td></td><td>3.1.a Notify user</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Users must fill in login details within 10 minutes.</td></tr>
</table>

Case Description-04: Verify Credentials

<table>
<tr><td>Use Case</td><td>Verify Credentials</td></tr>
<tr><td>Goal</td><td>Authenticate user credentials for system access.</td></tr>
<tr><td>Precondition</td><td>User has an existing account.</td></tr>
<tr><td>Success End Condition</td><td>User is successfully authenticated and granted access.</td></tr>
<tr><td>Failed End Condition</td><td>Authentication fails due to incorrect credentials.</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

System</td></tr>
<tr><td>Trigger</td><td>User attempts to log in.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User enters a username and password.</td></tr>
<tr><td>2.</td><td>System validates credentials.</td></tr>
<tr><td>3.</td><td>If correct, user is logged in and redirected to the dashboard.</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>Incorrect credentials</td></tr>
<tr><td></td><td>1.1.a System shows an error message and allows retry.</td></tr>
<tr><td>2.1</td><td>User forgets password</td></tr>
<tr><td></td><td>2.1.a Redirects to password reset.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Secure authentication mechanism, fast response time, protection against brute force attacks.</td></tr>
</table>

Case Description-05: Reset Password

<table>
<tr><td>Use Case</td><td>Reset Password</td></tr>
<tr><td>Goal</td><td>Allow users to reset their password if forgotten.</td></tr>
<tr><td>Precondition</td><td>User has an account and requests a password reset.</td></tr>
<tr><td>Success End Condition</td><td>User successfully resets the password and logs in.</td></tr>
<tr><td>Failed End Condition</td><td>Reset fails due to incorrect verification or expired reset link.</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

System</td></tr>
<tr><td>Trigger</td><td>User clicks "Forgot Password" and requests a reset.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User enters a registered email or phone number.</td></tr>
<tr><td>2.</td><td>System sends a reset link or code.</td></tr>
<tr><td>3.</td><td>User enters new password</td></tr>
<tr><td>4.</td><td>System updates password.</td></tr>
<tr><td>5.</td><td>User logs in with the new password.</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>Email/phone not found</td></tr>
<tr><td></td><td>1.1.a System notifies the user.</td></tr>
<tr><td>2.1</td><td>Reset link expires</td></tr>
<tr><td></td><td>2.1.a User requests a new link.</td></tr>
<tr><td>3.1</td><td>Weak password</td></tr>
<tr><td></td><td>System prompts for a stronger one.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Secure password encryption, time-sensitive reset links, user-friendly experience.</td></tr>
</table>

Case Description-06: Report Lost Item

<table>
<tr><td>Use Case</td><td>Report Lost Item</td></tr>
<tr><td>Goal</td><td>Users can report a lost item.</td></tr>
<tr><td>Precondition</td><td>User must be logged in.</td></tr>
<tr><td>Success End Condition</td><td>Notification: "Lost Item Reported Successfully"</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Lost Item Reporting Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>The user initiates the lost item report.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Press "Report Lost Item" button</td></tr>
<tr><td>2.</td><td>Provide item details</td></tr>
<tr><td>3.</td><td>Upload item images</td></tr>
<tr><td>4.</td><td>Submit report</td></tr>
<tr><td>5.</td><td>Notification: "Lost Item Reported Successfully"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again</td></tr>
<tr><td>2.1</td><td>Invalid Item Details</td></tr>
<tr><td></td><td>2.1.a Ask user to enter correct details</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Users should provide accurate item details.</td></tr>
</table>

Case Description-07: Upload Item Details

<table>
<tr><td>Use Case</td><td>Upload Item Details</td></tr>
<tr><td>Goal</td><td>Users can upload item details for a lost item.</td></tr>
<tr><td>Precondition</td><td>User must be reporting a lost item.</td></tr>
<tr><td>Success End Condition</td><td>Item details are saved successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Upload Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>The user provides item details.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User enters item details</td></tr>
<tr><td>2.</td><td>System validates information</td></tr>
<tr><td>3.</td><td>User uploads images</td></tr>
<tr><td>4.</td><td>System saves details</td></tr>
<tr><td>5.</td><td>Notification: "Item details uploaded successfully"</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Image File Too Large</td></tr>
<tr><td></td><td>2.1.a System asks user to upload a smaller file.</td></tr>
<tr><td>3.1</td><td>Missing Required Fields</td></tr>
<tr><td></td><td>3.1.a System prompts user to complete all fields.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Users must upload clear and relevant item details.</td></tr>
</table>

Case Description-08: Flag as Important

<table>
<tr><td>Use Case</td><td>Flag as Important</td></tr>
<tr><td>Goal</td><td>Users can mark a lost item as important.</td></tr>
<tr><td>Precondition</td><td>Item must be reported as lost.</td></tr>
<tr><td>Success End Condition</td><td>Item is flagged as important.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Failed to Flag Item!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>The user marks an item as important.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User selects lost item</td></tr>
<tr><td>2.</td><td>Clicks "Flag as Important"</td></tr>
<tr><td>3.</td><td>System updates item status</td></tr>
<tr><td>4.</td><td>Notification: "Item flagged as important"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Sys Try again later.</td></tr>
<tr><td>2.1</td><td>Item Already Flagged</td></tr>
<tr><td></td><td>2.1.a System notifies the user.</td></tr>
<tr><td>3.1</td><td>Unauthorized User Tries to Flag</td></tr>
<tr><td></td><td>3.1.a System denies access and alerts admin.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Only the item owner should be able to flag it.</td></tr>
</table>

Case Description-09: Edit Reported Item

<table>
<tr><td>Use Case</td><td>Edit Reported Item</td></tr>
<tr><td>Goal</td><td>Users can modify details of their reported lost item.</td></tr>
<tr><td>Precondition</td><td>Item must be reported as lost.</td></tr>
<tr><td>Success End Condition</td><td>Changes are saved successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Failed to Save Changes!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>The user updates the reported item details.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User selects lost item</td></tr>
<tr><td>2.</td><td>Clicks "Edit"</td></tr>
<tr><td>3.</td><td>Makes necessary changes</td></tr>
<tr><td>4.</td><td>Saves changes</td></tr>
<tr><td>5.</td><td>Notification: "Item details updated"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Changes Not Saved</td></tr>
<tr><td></td><td>2.1.a System prompts user to try again.</td></tr>
<tr><td>3.1</td><td>Unauthorized Edit Attempt</td></tr>
<tr><td></td><td>3.1.a System denies access and notifies the user.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Users should only edit their own reported items.</td></tr>
</table>

Case Description-10: Claim Lost Item

<table>
<tr><td>Use Case</td><td>Claim Lost Item</td></tr>
<tr><td>Goal</td><td>Users can claim an item they lost.</td></tr>
<tr><td>Precondition</td><td>User must provide proof of ownership.</td></tr>
<tr><td>Success End Condition</td><td>Item claim is approved.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Claim Rejected!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>The user initiates a claim request.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User selects lost item</td></tr>
<tr><td>2.</td><td>Provides ownership proof</td></tr>
<tr><td>3.</td><td>Submits claim</td></tr>
<tr><td>4.</td><td>Admin reviews claim</td></tr>
<tr><td>5.</td><td>Notification: "Claim Successful"</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Insufficient Proof Provided</td></tr>
<tr><td></td><td>2.1.a System requests additional documents.</td></tr>
<tr><td>3.1</td><td>Duplicate Claim Submission</td></tr>
<tr><td></td><td>3.1.a System notifies user of an existing claim.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Users must provide valid proof of ownership.</td></tr>
</table>

Case Description-11: Verify Ownership

<table>
<tr><td>Use Case</td><td>Verify Ownership</td></tr>
<tr><td>Goal</td><td>Admin verifies ownership proof for a claimed item.</td></tr>
<tr><td>Precondition</td><td>A user must have submitted a claim with proof.</td></tr>
<tr><td>Success End Condition</td><td>Ownership verified and claim approved.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Ownership Verification Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin

User</td></tr>
<tr><td>Trigger</td><td>Admin initiates verification process.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>Admin reviews submitted claim.</td></tr>
<tr><td>2.</td><td>Checks ownership proof.</td></tr>
<tr><td>3.</td><td>Confirms validity.</td></tr>
<tr><td>4.</td><td>Approves claim.</td></tr>
<tr><td>5.</td><td>Notification: "Claim Approved."</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Insufficient Proof</td></tr>
<tr><td></td><td>2.1.a Request additional documents.</td></tr>
<tr><td>3.1</td><td>Fraudulent Claim Detected</td></tr>
<tr><td></td><td>3.1.a Reject claim and notify user.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Admin must verify documents thoroughly to avoid fraud.</td></tr>
</table>

Case Description-12: Reject Claim

<table>
<tr><td>Use Case</td><td>Reject Claim</td></tr>
<tr><td>Goal</td><td>Admin rejects a claim due to invalid proof.</td></tr>
<tr><td>Precondition</td><td>A user must have submitted a claim.</td></tr>
<tr><td>Success End Condition</td><td>Notification: "Claim Rejected."</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Claim Rejection Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>Admin

User</td></tr>
<tr><td>Trigger</td><td>Admin decides to reject a claim.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>Admin reviews claim.</td></tr>
<tr><td>2.</td><td>Finds insufficient or false proof.</td></tr>
<tr><td>3.</td><td>Rejects the claim.</td></tr>
<tr><td>4.</td><td>Notifies the user with rejection reason.</td></tr>
<tr><td>5.</td><td>Claim status updated to "Rejected."</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>User Requests Re-evaluation</td></tr>
<tr><td></td><td>2.1.a Admin reviews claim again.</td></tr>
<tr><td>3.1</td><td>False Rejection</td></tr>
<tr><td></td><td>3.1.a User is allowed to appeal.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Admin should provide clear reasons for rejection.</td></tr>
</table>

Case Description-13: Request Contact with Finder

<table>
<tr><td>Use Case</td><td>Request Contact with Finder</td></tr>
<tr><td>Goal</td><td>Users request admin to contact the person who found their item.</td></tr>
<tr><td>Precondition</td><td>The item must be marked as found.</td></tr>
<tr><td>Success End Condition</td><td>Finder and owner are successfully connected.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Contact Request Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>User submits a contact request.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User selects found item.</td></tr>
<tr><td>2.</td><td>Requests contact with the finder.</td></tr>
<tr><td>3.</td><td>Admin reviews request.</td></tr>
<tr><td>4.</td><td>Contact details shared securely.</td></tr>
<tr><td>5.</td><td>Notification: "Contact Request Approved."</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Finder Declines Contact</td></tr>
<tr><td></td><td>2.1.a User is notified.</td></tr>
<tr><td>3.1</td><td>Unauthorized Contact Request</td></tr>
<tr><td></td><td>3.1.a System denies request and alerts admin.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Contact details should be shared securely and with consent.</td></tr>
</table>

Case Description-14: Send Message

<table>
<tr><td>Use Case</td><td>Send Message</td></tr>
<tr><td>Goal</td><td>Users can send messages related to lost items.</td></tr>
<tr><td>Precondition</td><td>User must have a verified account.</td></tr>
<tr><td>Success End Condition</td><td>Message sent successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Message Sending Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>User initiates message sending.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User opens message interface.</td></tr>
<tr><td>2.</td><td>Composes a message.</td></tr>
<tr><td>3.</td><td>Sends it to admin/finder.</td></tr>
<tr><td>4.</td><td>System delivers message.</td></tr>
<tr><td>5.</td><td>Notification: "Message Sent."</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Message Contains Restricted Content</td></tr>
<tr><td></td><td>2.1.a System blocks message and notifies user.</td></tr>
<tr><td>3.1</td><td>Recipient Unavailable</td></tr>
<tr><td></td><td>3.1.a User is informed of delivery failure.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Messages should be monitored for inappropriate content.</td></tr>
</table>

Case Description-15: Receive Reward Points

<table>
<tr><td>Use Case</td><td>Receive Reward Points</td></tr>
<tr><td>Goal</td><td>Users receive points for reporting or returning lost items.</td></tr>
<tr><td>Precondition</td><td>The user must have successfully completed an action that earns points.</td></tr>
<tr><td>Success End Condition</td><td>Points are added to the user's account.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Failed to Receive Points!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

System</td></tr>
<tr><td>Trigger</td><td>The system automatically assigns reward points.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User reports or returns a lost item.</td></tr>
<tr><td>2.</td><td>System verifies action.</td></tr>
<tr><td>3.</td><td>Points are calculated based on predefined rules.</td></tr>
<tr><td>4.</td><td>System updates user’s reward balance.</td></tr>
<tr><td>5.</td><td>Notification: "You have received reward points!"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Action Does Not Qualify for Points</td></tr>
<tr><td></td><td>2.1.a System notifies user.</td></tr>
<tr><td>3.1</td><td>Points Not Updated</td></tr>
<tr><td></td><td>3.1.a System recalculates and retries.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Reward points should be credited immediately and accurately.</td></tr>
</table>

Case Description-16: Earn Badges

<table>
<tr><td>Use Case</td><td>Earn Badges</td></tr>
<tr><td>Goal</td><td>Users can earn badges based on their contributions.</td></tr>
<tr><td>Precondition</td><td>Users must complete specific actions that qualify for a badge.</td></tr>
<tr><td>Success End Condition</td><td>Badge is awarded and displayed on the user's profile.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Failed to Earn Badge!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

System</td></tr>
<tr><td>Trigger</td><td>User completes an action that qualifies for a badge.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User performs an action that qualifies for a badge.</td></tr>
<tr><td>2.</td><td>System verifies the completion of the action.</td></tr>
<tr><td>3.</td><td>System awards a badge.</td></tr>
<tr><td>4.</td><td>Badge is displayed on the user's profile.</td></tr>
<tr><td>5.</td><td>Notification: "Congratulations! You've earned a badge!"</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Action Does Not Meet Badge Criteria</td></tr>
<tr><td></td><td>2.1.a System notifies the user.</td></tr>
<tr><td>3.1</td><td>Badge Not Displayed</td></tr>
<tr><td></td><td>3.1.a System refreshes and updates user profile.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Users should be notified instantly when earning a badge.</td></tr>
</table>

Case Description-17: Submit Feedback

<table>
<tr><td>Use Case</td><td>Submit Feedback</td></tr>
<tr><td>Goal</td><td>Users can submit feedback on the system or their experience.</td></tr>
<tr><td>Precondition</td><td>User must be logged in to submit feedback.</td></tr>
<tr><td>Success End Condition</td><td>Feedback is successfully recorded in the system.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Feedback Submission Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>User submits feedback through the system.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User navigates to the feedback section.</td></tr>
<tr><td>2.</td><td>Fills out the feedback form.</td></tr>
<tr><td>3.</td><td>Submits the feedback.</td></tr>
<tr><td>4.</td><td>System records the feedback.</td></tr>
<tr><td>5.</td><td>Notification: "Feedback Submitted Successfully!"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Feedback Contains Restricted Content</td></tr>
<tr><td></td><td>2.1.a System blocks and notifies user.</td></tr>
<tr><td>3.1</td><td>Form Not Completed</td></tr>
<tr><td></td><td>3.1.a System prompts user to fill in required fields.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Feedback should be stored securely and used for system improvements.</td></tr>
</table>

Case Description-18: Rate Experience

<table>
<tr><td>Use Case</td><td>Rate Experience</td></tr>
<tr><td>Goal</td><td>Users can rate their experience on the platform.</td></tr>
<tr><td>Precondition</td><td>User must have interacted with a lost item report or claim.</td></tr>
<tr><td>Success End Condition</td><td>Rating is submitted and stored successfully.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Rating Submission Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

System</td></tr>
<tr><td>Trigger</td><td>User submits a rating after an interaction.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User selects a star rating.</td></tr>
<tr><td>2.</td><td>(Optional) Provides additional comments.</td></tr>
<tr><td>3.</td><td>Clicks submit.</td></tr>
<tr><td>4.</td><td>System stores the rating.</td></tr>
<tr><td>5.</td><td>Notification: "Thank you for your rating!"</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Invalid Rating Selection</td></tr>
<tr><td></td><td>2.1.a System prompts user to select a valid rating.</td></tr>
<tr><td>3.1</td><td>Rating Not Saved</td></tr>
<tr><td></td><td>3.1.a System attempts to resubmit.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Ratings should be collected and analyzed for system improvements.</td></tr>
</table>

Case Description-19: View Feedback History

<table>
<tr><td>Use Case</td><td>View Feedback History</td></tr>
<tr><td>Goal</td><td>Users can view their previously submitted feedback.</td></tr>
<tr><td>Precondition</td><td>User must have submitted feedback before.</td></tr>
<tr><td>Success End Condition</td><td>Feedback history is displayed.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Unable to Load Feedback History!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

System</td></tr>
<tr><td>Trigger</td><td>User navigates to the feedback history section.</td></tr>
<tr><td>Description / Main Success Scenario</td><td>
<table>
<tr><td>1.</td><td>User opens feedback history.</td></tr>
<tr><td>2.</td><td>System retrieves all submitted feedback.</td></tr>
<tr><td>3.</td><td>User views past feedback details.</td></tr>
<tr><td>4.</td><td>Optionally, user can edit or delete feedback.</td></tr>
<tr><td>5.</td><td>Notification: "Feedback history loaded successfully!"</td></tr>
</table>
</td></tr>
<tr><td>Alternative Flows</td><td>
<table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>No Feedback Found</td></tr>
<tr><td></td><td>2.1.a System informs user.</td></tr>
<tr><td>3.1</td><td>Data Retrieval Failed</td></tr>
<tr><td></td><td>3.1.a System refreshes and retries.</td></tr>
</table>
</td></tr>
<tr><td>Quality Requirements</td><td>Feedback history should be updated in real-time.</td></tr>
</table>

Case Description-20: Report Fake Listing

<table>
<tr><td>Use Case</td><td>Report Fake Listing</td></tr>
<tr><td>Goal</td><td>Users can report fake or fraudulent lost item listings.</td></tr>
<tr><td>Precondition</td><td>User must have seen a suspicious listing.</td></tr>
<tr><td>Success End Condition</td><td>Fake listing is flagged for admin review.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Failed to Report Fake Listing!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>User reports a listing suspected to be fake.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User selects the suspicious listing.</td></tr>
<tr><td>2.</td><td>Clicks "Report Fake Listing."</td></tr>
<tr><td>3.</td><td>Provides a reason.</td></tr>
<tr><td>4.</td><td>System flags the listing for admin review.</td></tr>
<tr><td>5.</td><td>Notification: "Report Submitted Successfully."</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Report Lacks Sufficient Information</td></tr>
<tr><td></td><td>2.1.a System requests more details.</td></tr>
<tr><td>3.1</td><td>Listing Already Reported</td></tr>
<tr><td></td><td>3.1.a System informs user of ongoing review.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Fake listing reports should be reviewed promptly.</td></tr>
</table>

Case Description-21: Submit Complaint

<table>
<tr><td>Use Case</td><td>Submit Complaint</td></tr>
<tr><td>Goal</td><td>Users can submit complaints regarding platform issues or disputes.</td></tr>
<tr><td>Precondition</td><td>User must be logged in.</td></tr>
<tr><td>Success End Condition</td><td>Complaint is recorded and sent for review.</td></tr>
<tr><td>Failed End Condition</td><td>Notification: "Complaint Submission Failed!"</td></tr>
<tr><td>Primary Actors:

Secondary Actors:</td><td>User

Admin</td></tr>
<tr><td>Trigger</td><td>User initiates a complaint submission.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table>
<tr><td>1.</td><td>User fills out the complaint form.</td></tr>
<tr><td>2.</td><td>Provides necessary details.</td></tr>
<tr><td>3.</td><td>Submits the complaint.</td></tr>
<tr><td>4.</td><td>System records and forwards it to admin.</td></tr>
<tr><td>5.</td><td>Notification: "Complaint Submitted Successfully."</td></tr>
</table></td></tr>
<tr><td>Alternative Flows</td><td><table>
<tr><td>1.1</td><td>System Error</td></tr>
<tr><td></td><td>1.1.a Try again later.</td></tr>
<tr><td>2.1</td><td>Missing Complaint Details</td></tr>
<tr><td></td><td>2.1.a System prompts user to complete required fields.</td></tr>
<tr><td>3.1</td><td>Complaint Not Sent</td></tr>
<tr><td></td><td>3.1.a System retries submission.</td></tr>
</table></td></tr>
<tr><td>Quality Requirements</td><td>Complaints should be addressed in a timely manner.</td></tr>
</table>

Case Description-22: Review Complaint

Use Case	Review Complaint
Goal	Admin reviews user-submitted complaints about fake or incorrect listings.
Precondition	A complaint must be submitted by a user.
Success End Condition	The complaint is reviewed, and a decision is made.
Failed End Condition	Complaint remains unresolved.
Primary Actors	Admin, System
Secondary Actors	
Trigger	A user submits a complaint.
Description / Main Success Scenario	1. Admin logs into the system. 2. Admin navigates to the complaint section. 3. Admin reviews complaint details and evidence. 4. Admin verifies the claim. 5. Admin marks the complaint as reviewed. 6. System records the review.
Alternative Flows	1.1 System Error 1.1.a Try Again! 2.1 No valid evidence is found 2.1.a Complaint is rejected. 3.1 Complaint is already resolved 3.1.a Notify user.
Quality Requirements	Complaints must be reviewed within 48 hours.

Case Description-23: Take Action

<table>
<tr><td>Use Case</td><td>Take Action</td></tr>
<tr><td>Goal</td><td>Admin takes appropriate action based on complaint review.</td></tr>
<tr><td>Precondition</td><td>Complaint must be reviewed.</td></tr>
<tr><td>Success End Condition</td><td>The action (warning, suspension, ban) is recorded and enforced.</td></tr>
<tr><td>Failed End Condition</td><td>No action is taken.</td></tr>
<tr><td>Primary Actors</td><td>Admin</td></tr>
<tr><td>Secondary Actors</td><td></td></tr>
<tr><td>Trigger</td><td>Admin reviews a complaint.</td></tr>
<tr><td>Description / Main Success Scenario</td><td><table><tr><td>1.</td><td>Admin selects the reviewed complaint.</td></tr><tr><td>2.</td><td>Admin chooses an action (warn, suspend, ban).</td></tr><tr><td>3.</td><td>Admin confirms the action.</td></tr><tr><td>4.</td><td>System records and enforces the action.</td></tr><tr><td>5.</td><td>System notifies the involved user(s).</td></tr></table></td></tr>
<tr><td>Alternative Flows</td><td><table><tr><td>1.1</td><td>System Error</td></tr><tr><td></td><td>1.1.a Try Again!</td></tr><tr><td>2.1</td><td>User appeals action</td></tr><tr><td></td><td>2.1.a Admin re-evaluates the case.</td></tr></table></td></tr>
<tr><td>Quality Requirements</td><td>Actions must be enforced within 24 hours.</td></tr>
</table>

Case Description-24: Manage Users

Use Case	Manage Users
Goal	Admin manages user accounts, including activation, suspension, and removal.
Precondition	Admin must be logged in.
Success End Condition	User management changes are successfully applied.
Failed End Condition	No changes are made.
Primary Actors	Admin
Secondary Actors	
Trigger	A new user registered.
Description / Main Success Scenario	1. Admin accesses the user management panel. 2. Admin searches for a user. 3. Admin modifies user status (active, suspended, banned). 4. System applies the changes.
Alternative Flows	1.1 System Error 1.1.a Try Again! 2.1 User does not exist 2.1.a how error message.
Quality Requirements	System updates user records instantly.

Case Description-25: Monitor Reports & Feedback

Use Case	Monitor Reports & Feedback
Goal	Admin monitors reported listings and user feedback.
Precondition	Reports and feedback exist in the system.
Success End Condition	Reports and feedback are reviewed and analyzed.
Failed End Condition	Reports and feedback remain unreviewed.
Primary Actors	Admin
Secondary Actors	
Trigger	A user submits a report or feedback.
Description / Main Success Scenario	1. Admin accesses reports and feedback. 2. Admin filters reports by status (pending, resolved). 3. Admin analyzes trends. 4. Admin marks reports as reviewed.
Alternative Flows	1.1 No new reports available 1.1.a Display "No New Reports."
Quality Requirements	Reports must be reviewed weekly.

Case Description-26: Take Disciplinary Action

Use Case	Take Disciplinary Action
Goal	Admin takes disciplinary action against users violating policies.
Precondition	User violation must be identified.
Success End Condition	Action is recorded and enforced.
Failed End Condition	No action is taken.
Primary Actors	Admin
Secondary Actors	
Trigger	A user violates a rule.
Description / Main Success Scenario	1. Admin selects a user. 2. Admin selects the type of action (warn, suspend, ban). 3. Admin submits the action. 4. System enforces the action.
Alternative Flows	1.1 Action does not meet criteria 1.1.a Display error message.
Quality Requirements	Actions must be reviewed before enforcement.

Case Description-27: Suspend User

Use Case	Suspend User
Goal	Admin suspends a user for rule violations.
Precondition	Violation must be confirmed.
Success End Condition	User is suspended.
Failed End Condition	User remains active.
Primary Actors	Admin
Secondary Actors	
Trigger	Admin clicks Suspend based on violation.
Description / Main Success Scenario	1. Admin selects the violating user. 2. Admin clicks "Suspend." 3. System applies suspension.
Alternative Flows	1.1 User not found 1.1.a Show error message.
Quality Requirements	Suspension should be effective immediately.

Case Description-28: Generate QR Code for Found Item

Use Case	Generate QR Code for Found Item
Goal	Finder generates a QR code for a reported found item.
Precondition	Item must be reported.
Success End Condition	QR code is generated and linked to the item.
Failed End Condition	QR code generation fails.
Primary Actors	Student, Faculty, Staff.
Secondary Actors	System
Trigger	User reports a found item.
Description / Main Success Scenario	1. Finder selects a found item. 2. System generates a QR code. 3. Finder downloads/prints the QR code.
Alternative Flows	1.1 System Error 1.1.a Try Again! 2.1 QR code generation fails 2.1.a Try Again Later.
Quality Requirements	QR codes should be unique and secure.

Case Description-29: Scan QR Code

Use Case	Scan QR Code
Goal	Claimant scans a QR code to retrieve item details.
Precondition	QR code must exist.
Success End Condition	Item details are retrieved.
Failed End Condition	QR scan fails.
Primary Actors	Student, Faculty, Staff.
Secondary Actors	System
Trigger	Claimant clicks scan button.
Description / Main Success Scenario	1. Claimant scans the QR code. 2. System fetches item details. 3. Claimant reviews details.
Alternative Flows	1.1 Invalid QR code 1.1.a Display error message.
Quality Requirements	QR scans should retrieve details within 2 seconds.

Case Description-30: Review Item Details

Use Case	Review Item Details
Goal	Claimant reviews item details before claiming.
Precondition	Item must exist in the system.
Success End Condition	Claimant views item details.
Failed End Condition	Details are not retrieved.
Primary Actors	Student, Faculty, Staff.
Secondary Actors	System
Trigger	Claimant checks item details.
Description / Main Success Scenario	1. Claimant accesses item details. 2. System displays item description and images. 3. Claimant verifies details.
Alternative Flows	1.1 Item not found 1.1.a Show error message.
Quality Requirements	Items should display correct details 100% of the time.

Case Description-31: Logout

Use Case	Logout
Goal	User logs out from the system.
Precondition	User must be logged in.
Success End Condition	User is logged out.
Failed End Condition	Logout fails.
Primary Actors	Student, Faculty, Staff
Secondary Actors	
Trigger	User clicks Logout.
Description / Main Success Scenario	1. User clicks "Logout." 2. System ends the session. 3. User is redirected to the login page.
Alternative Flows	1.1 System Error 1.1.a Try Again!
Quality Requirements	Logout should happen within 2 seconds.

System Design

Data Flow Diagram

Data Flow Diagram Level 0:

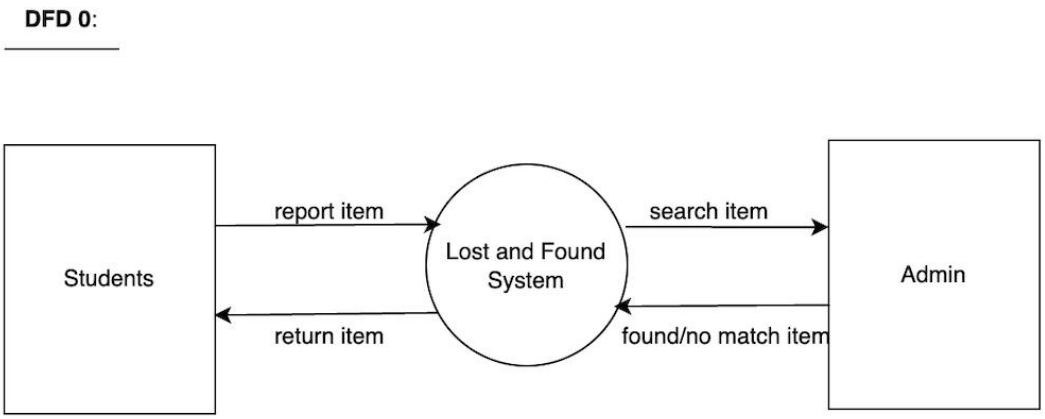


Figure: Data Flow Diagram Level 0

Data Flow Diagram Level 1:

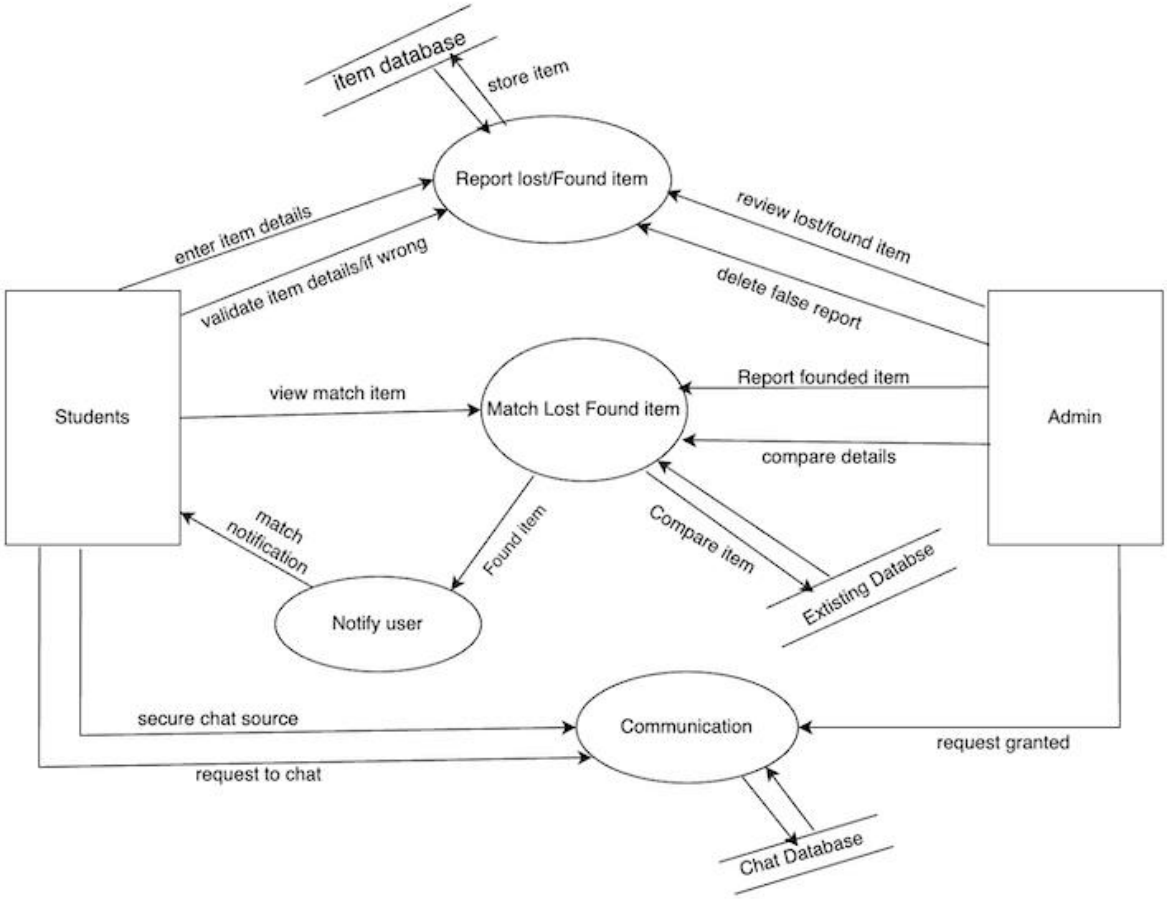


Figure: Data Flow Diagram Level - 1

Data Flow Diagram Level 2:

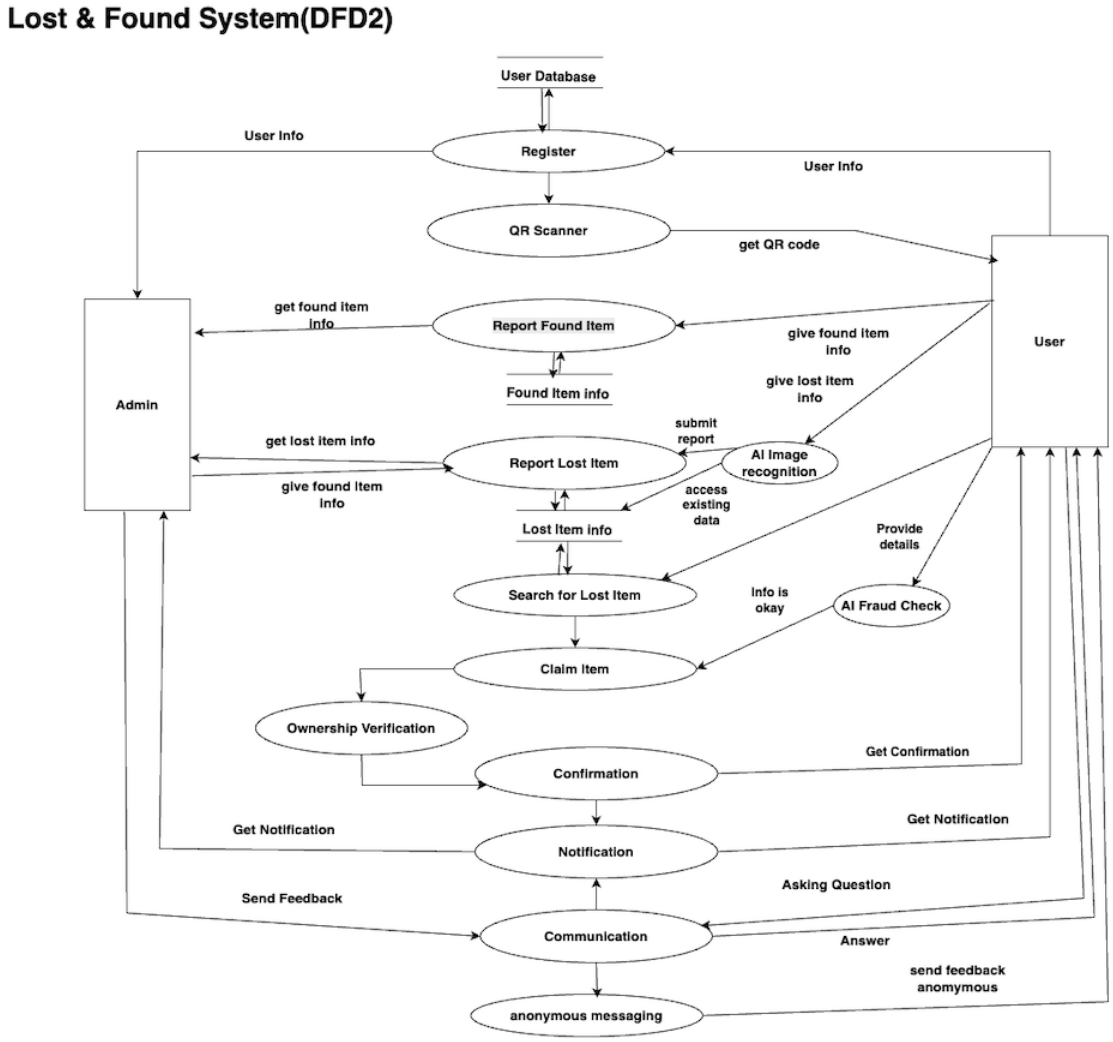
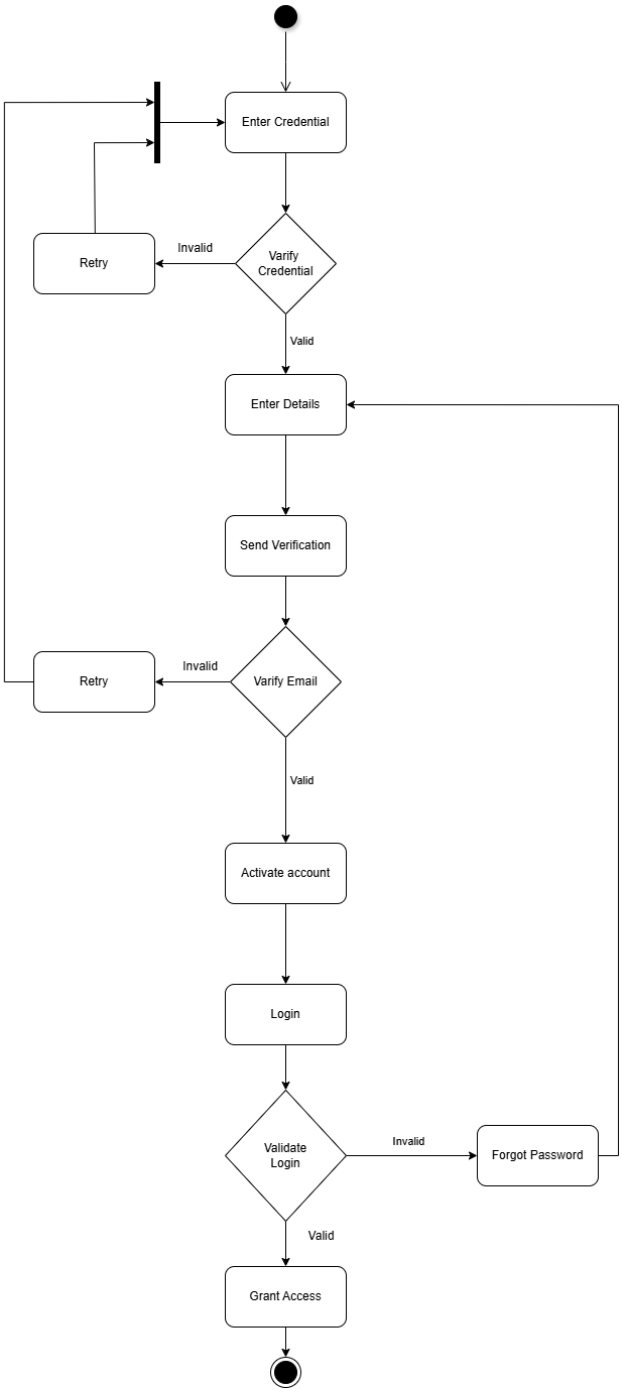


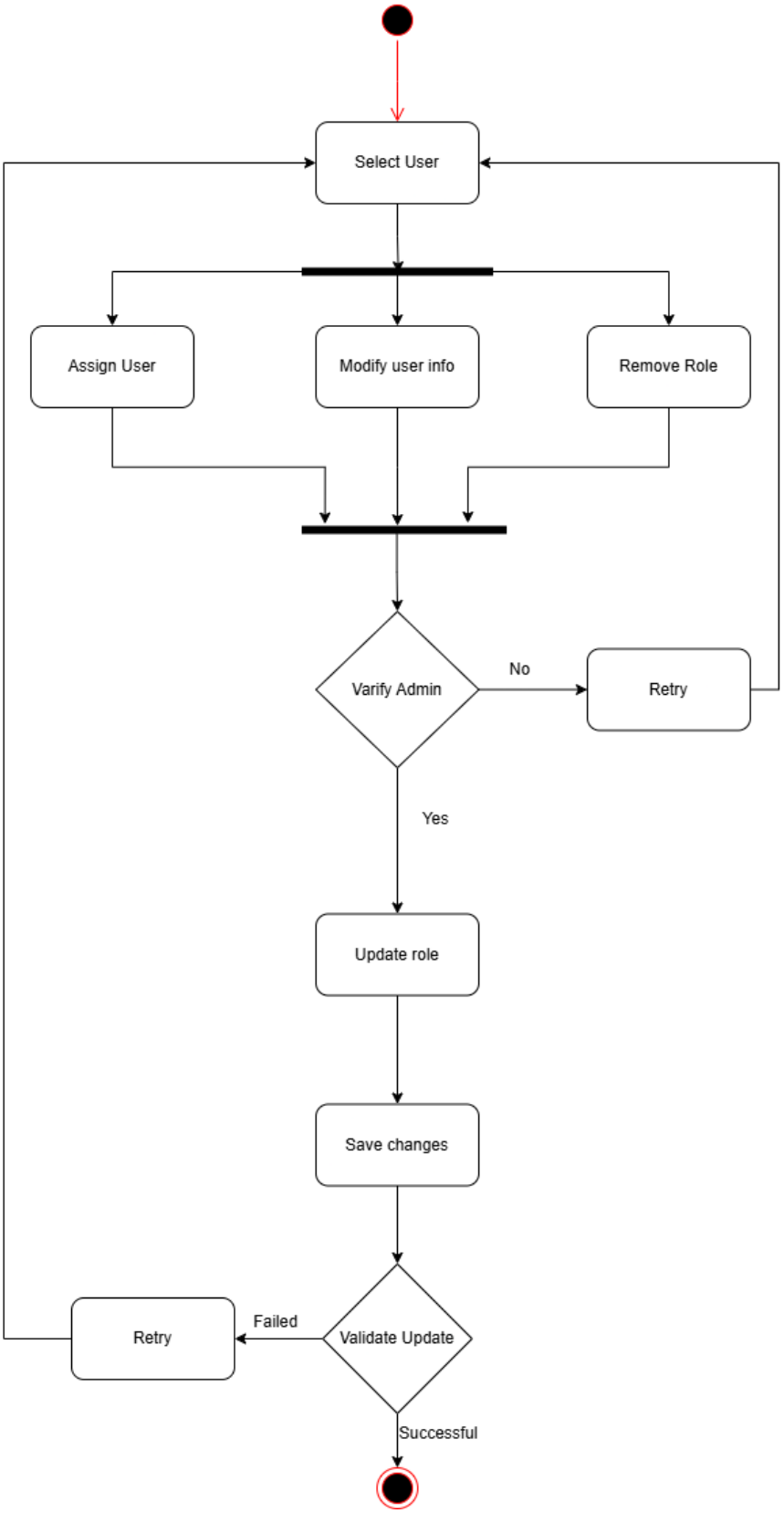
Figure: Data Flow Diagram Level - 2

Activity Diagram

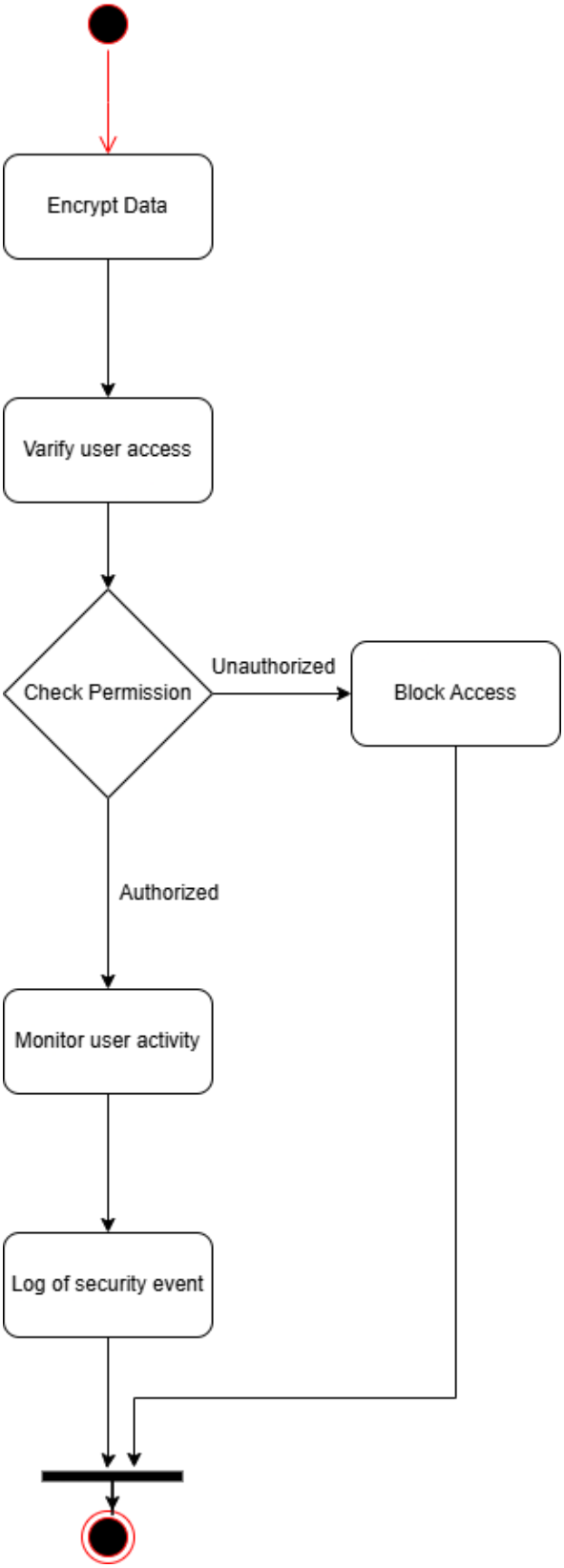
FR01 - User Registration and Authentication



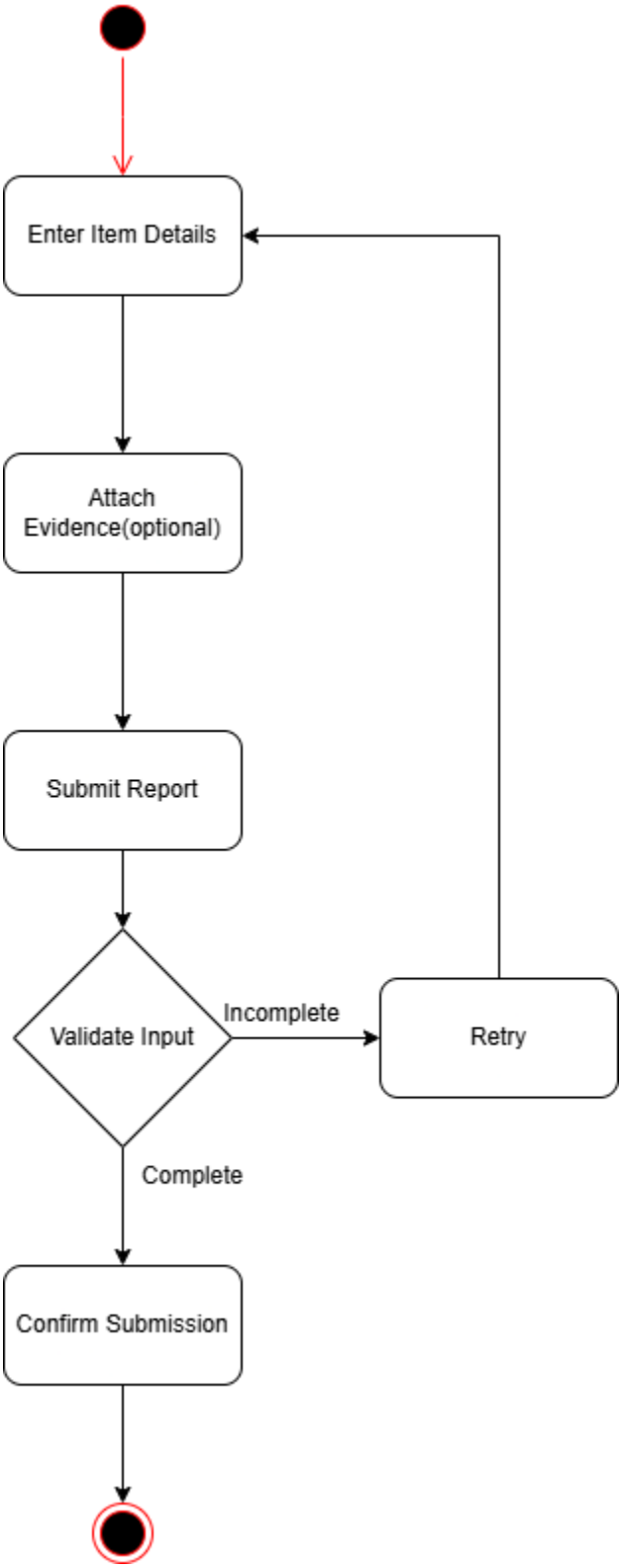
FR02 - User Role Management



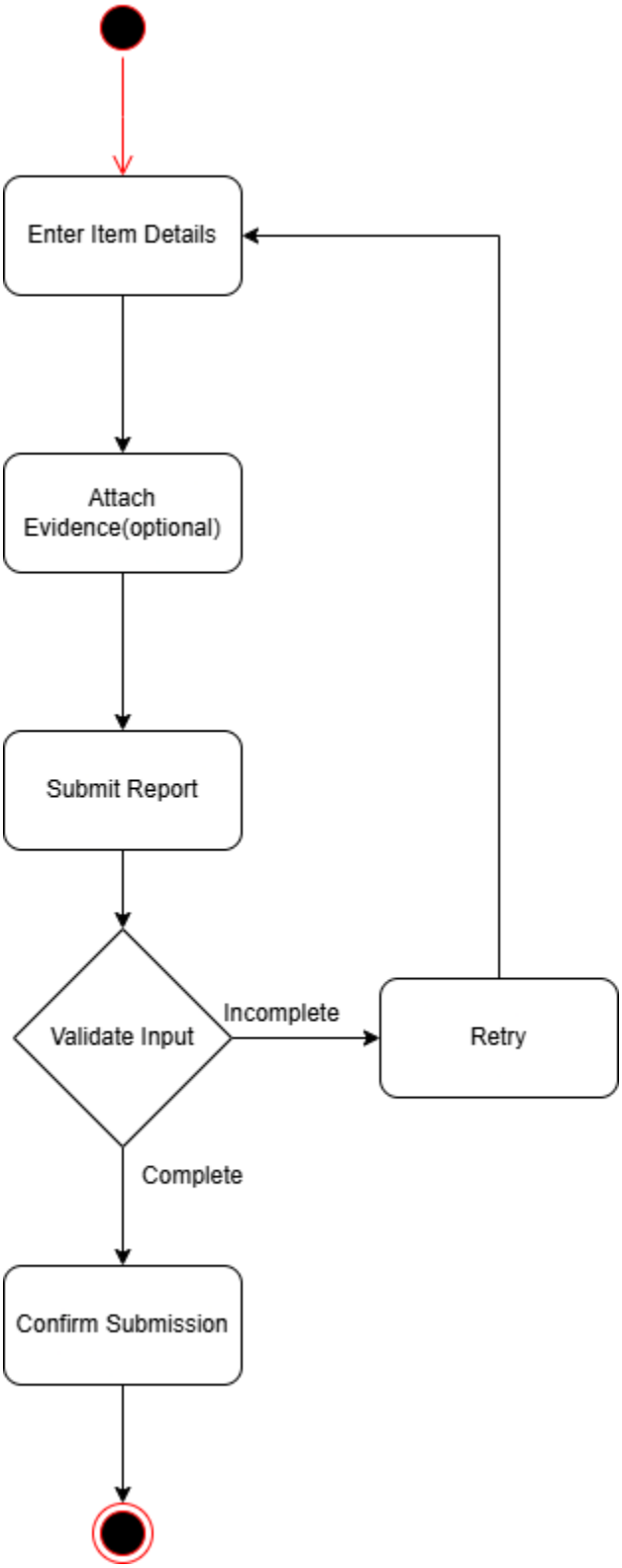
FR03 - Security & Data Privacy



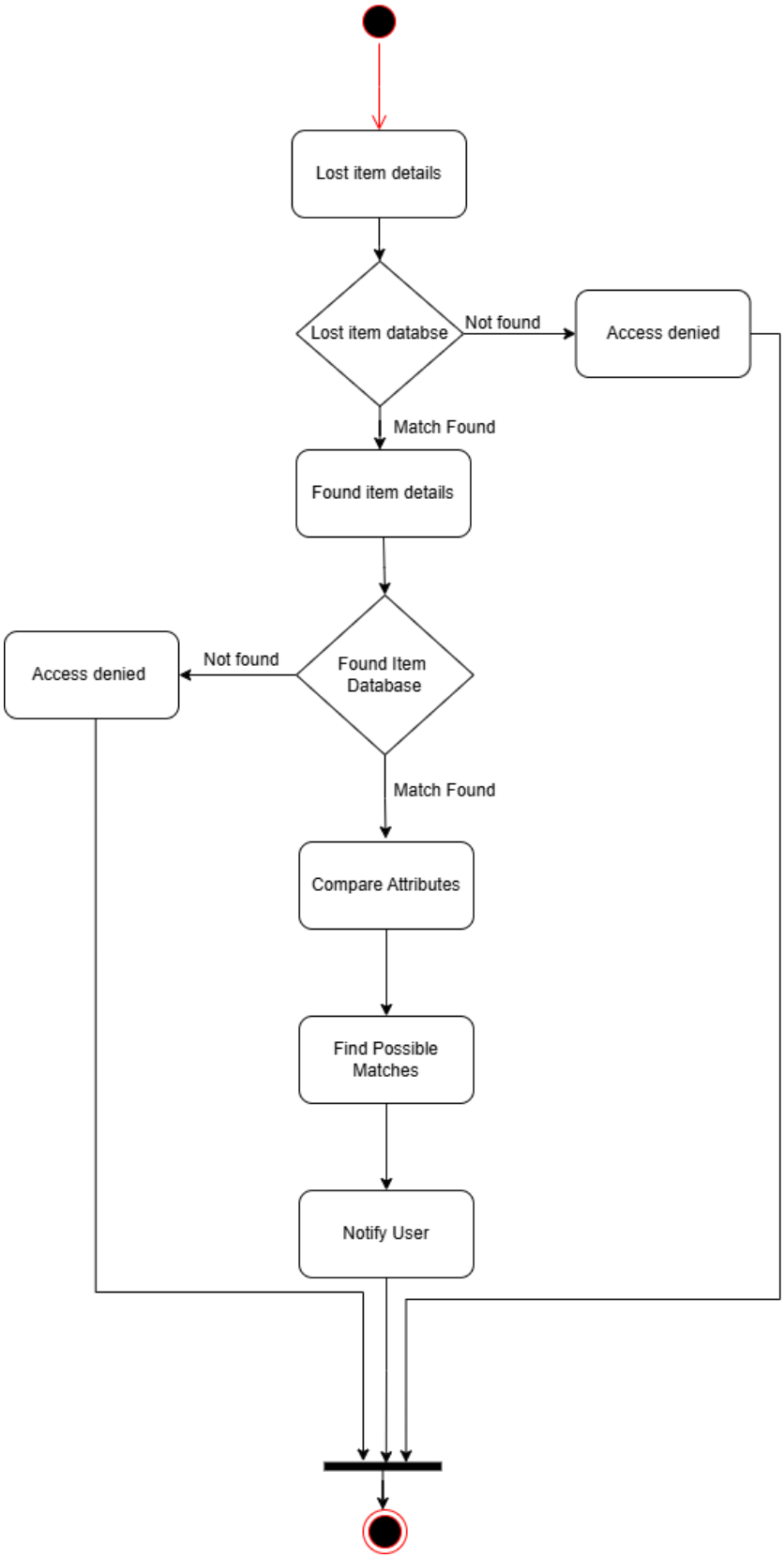
FR04 - Lost Item Reporting



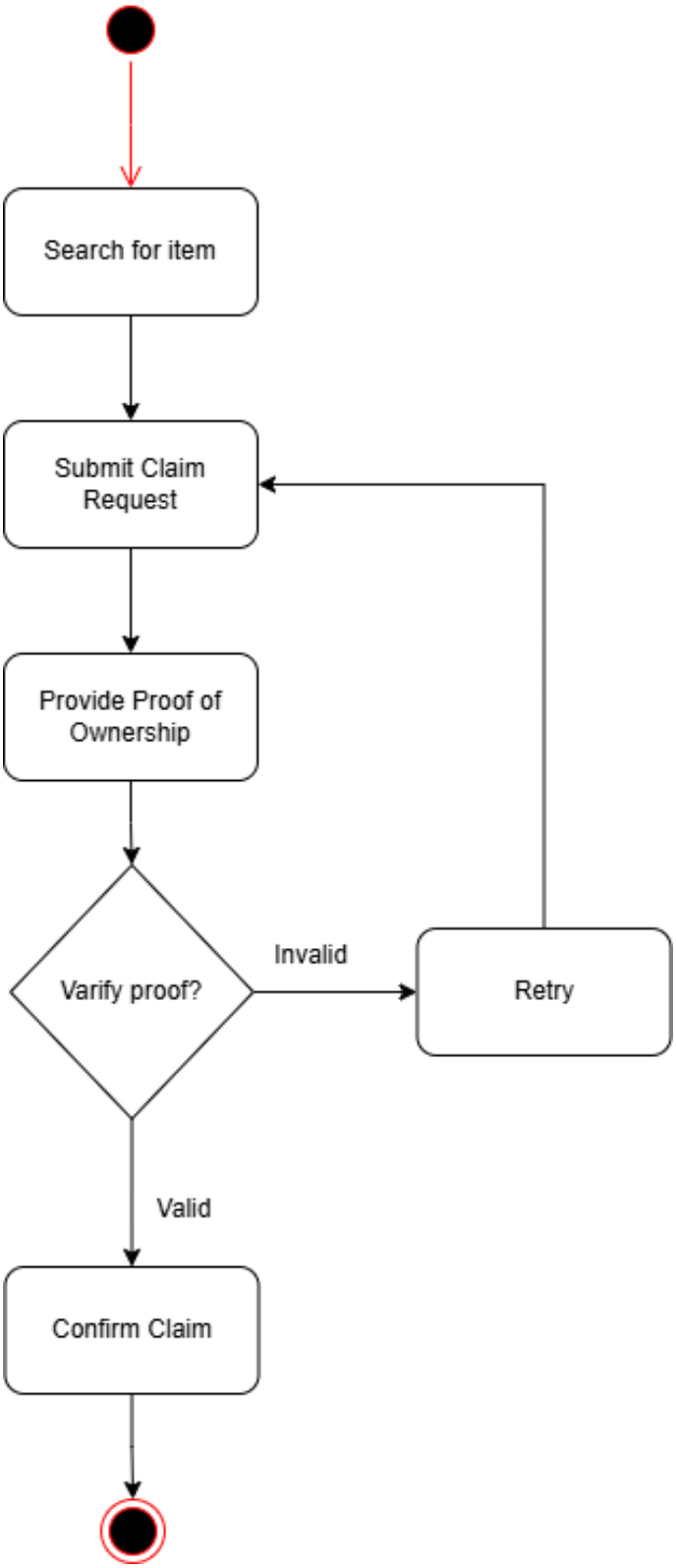
FR05 - Found Item Reporting



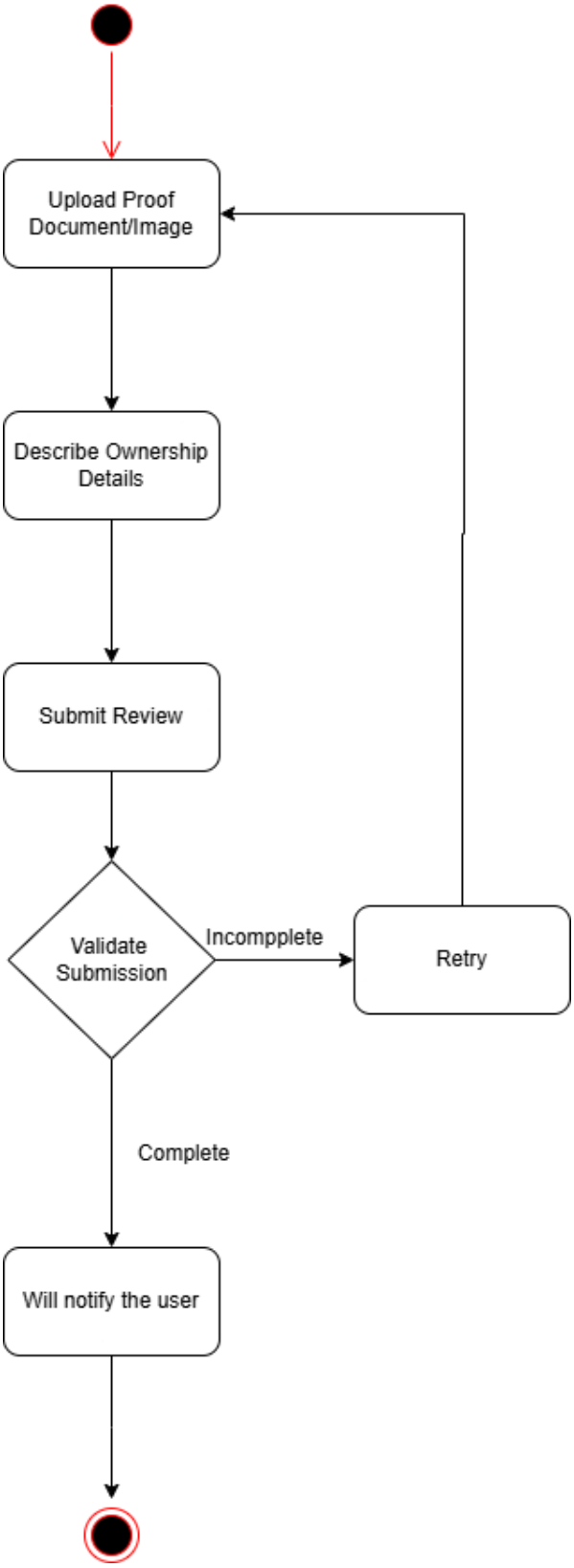
FR06 - Lost and Found Item Matching



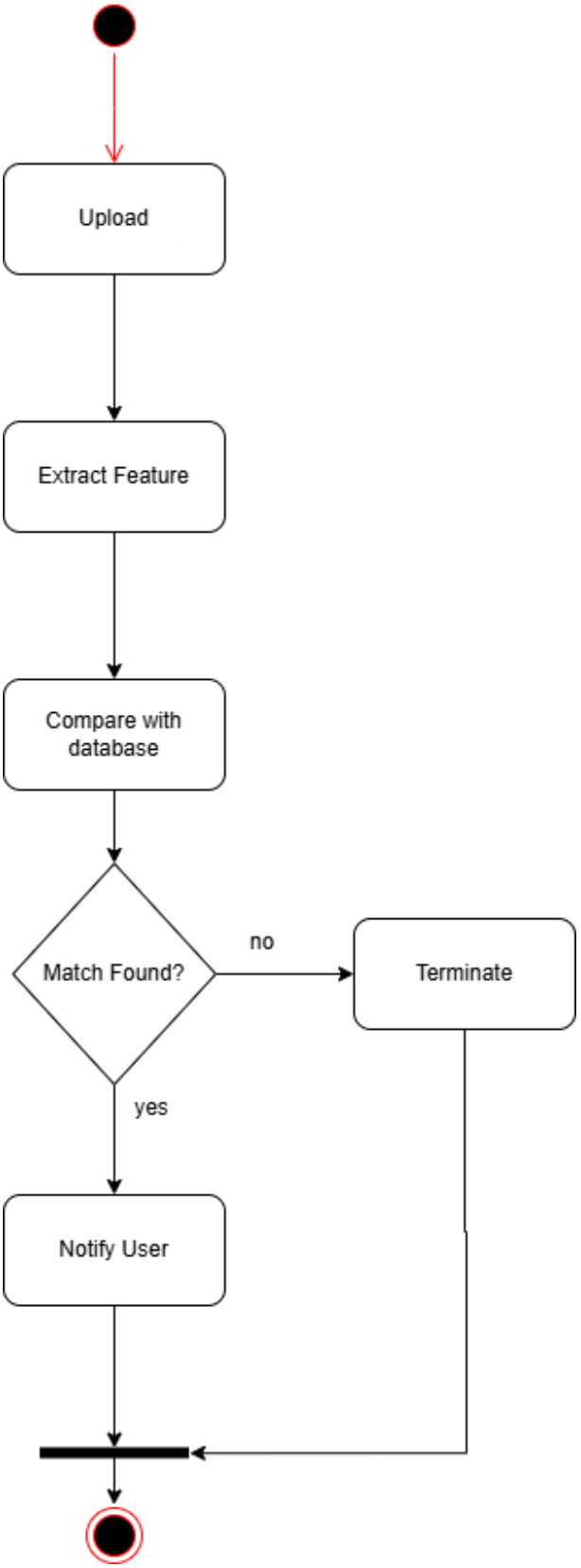
FR07 - Claiming a Found Item



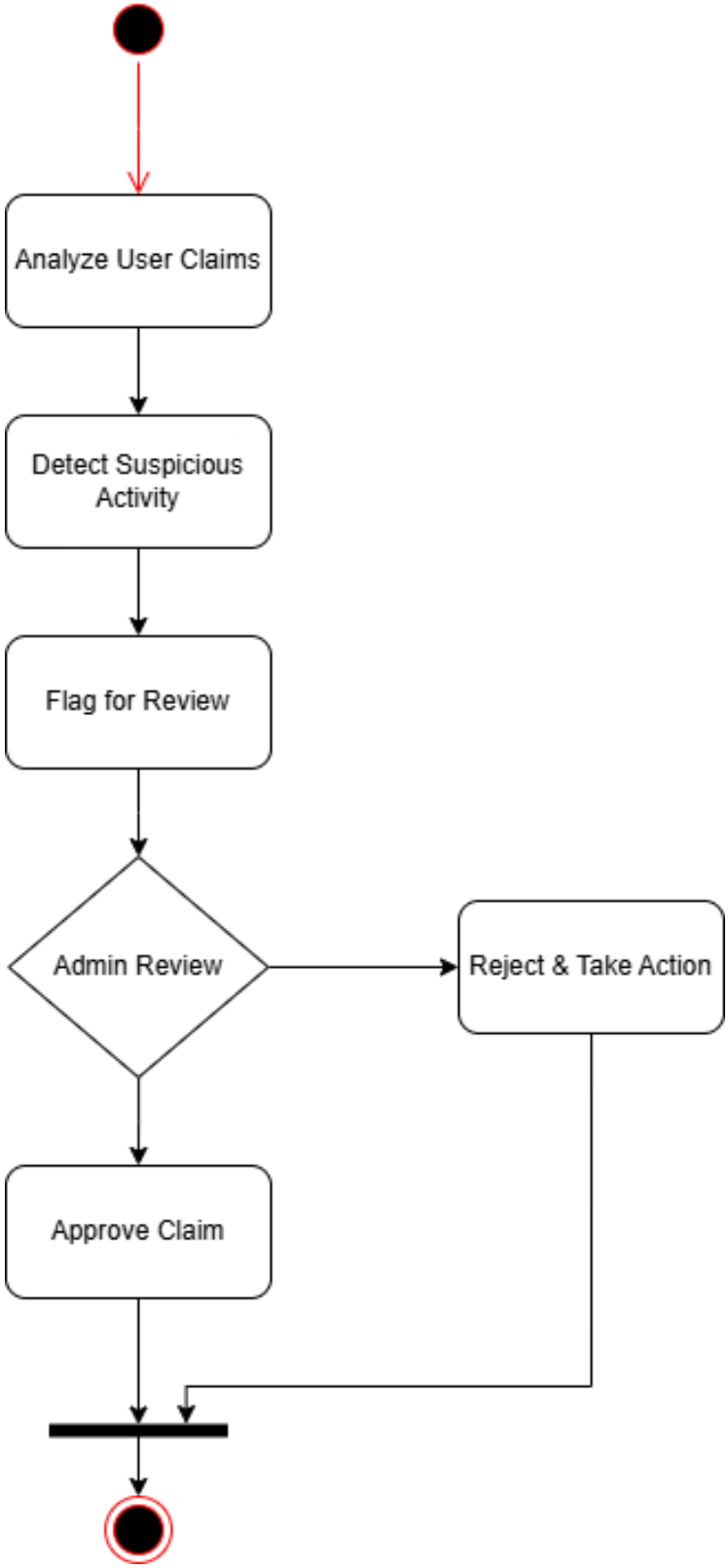
FR08 - Proof of Ownership Submission



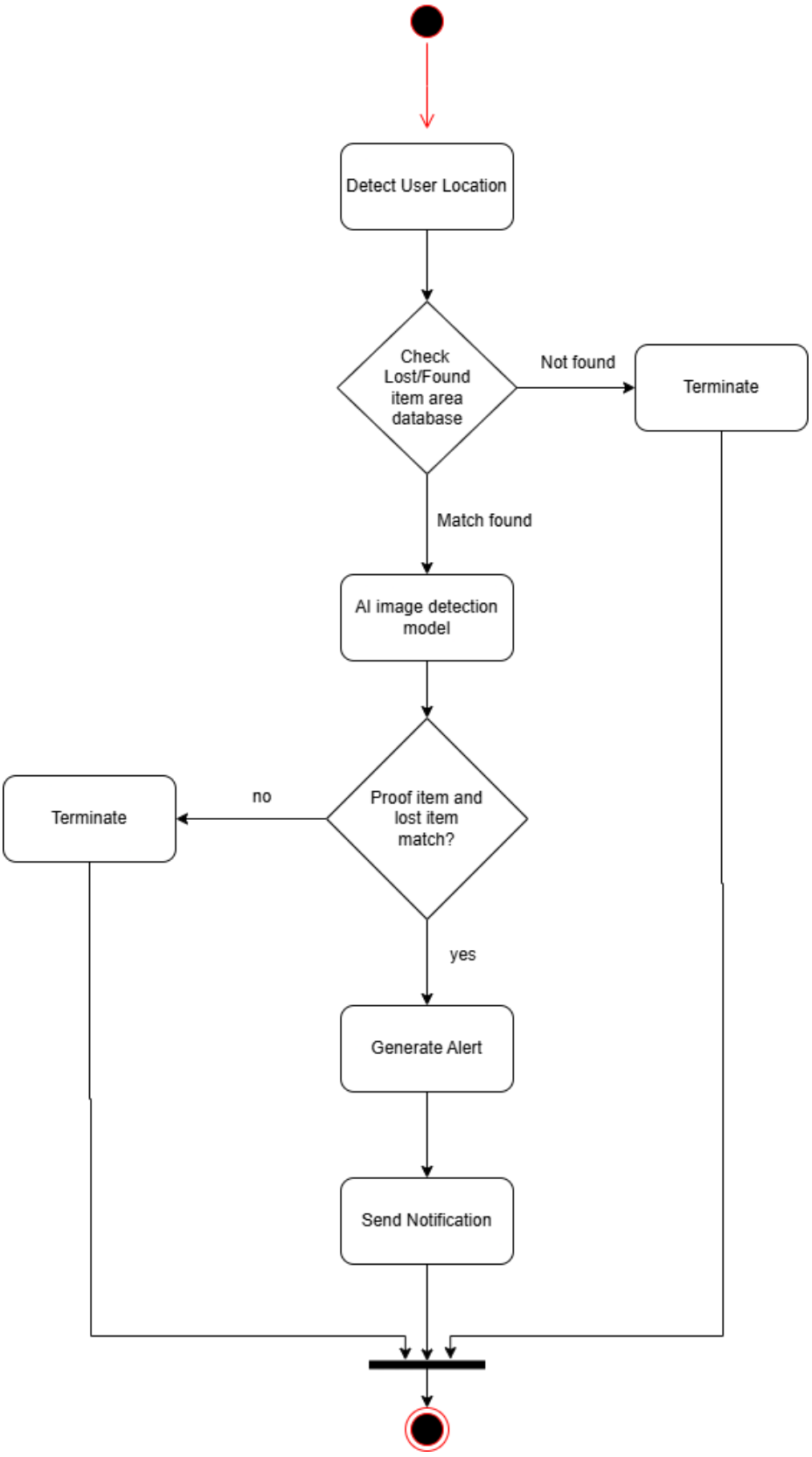
FR09 - AI-Based Image Recognition for Matching



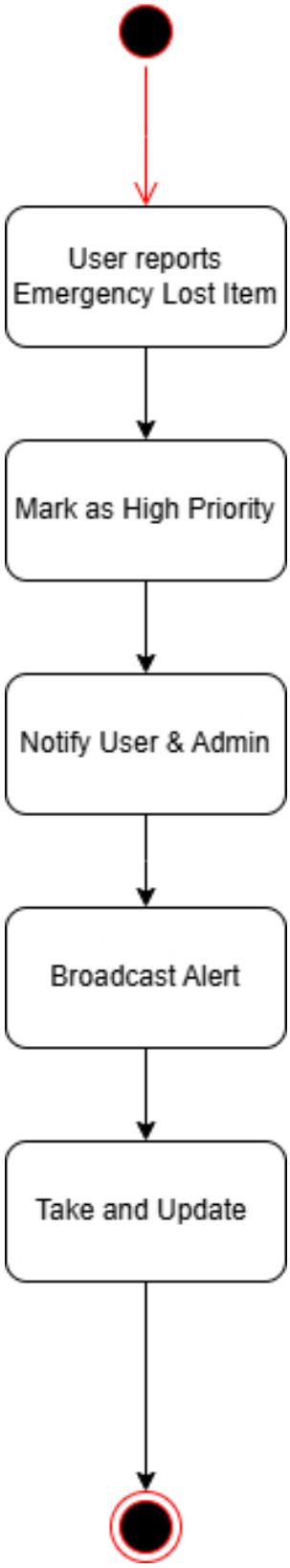
FR10 - AI-Powered Fraud Detection



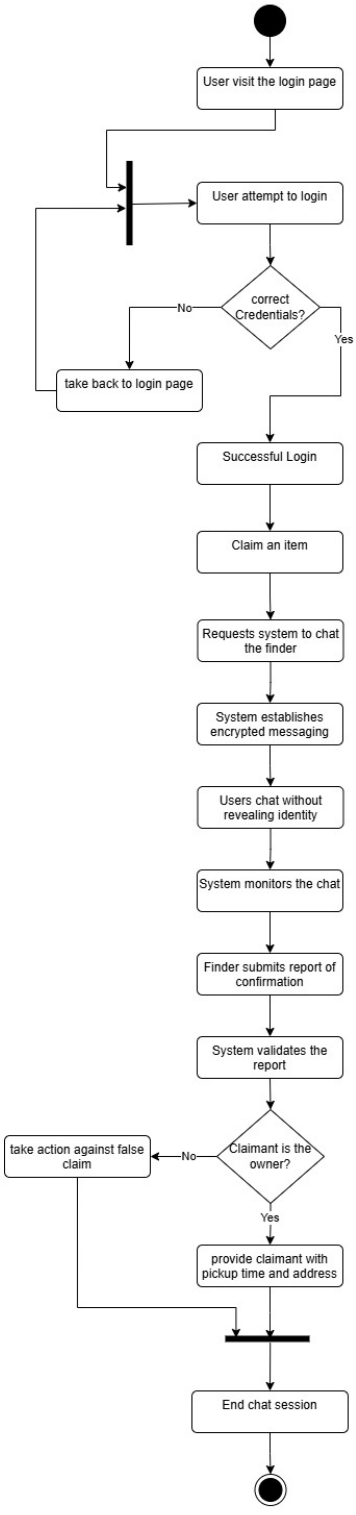
FR11 - Smart Location-Based Alerts



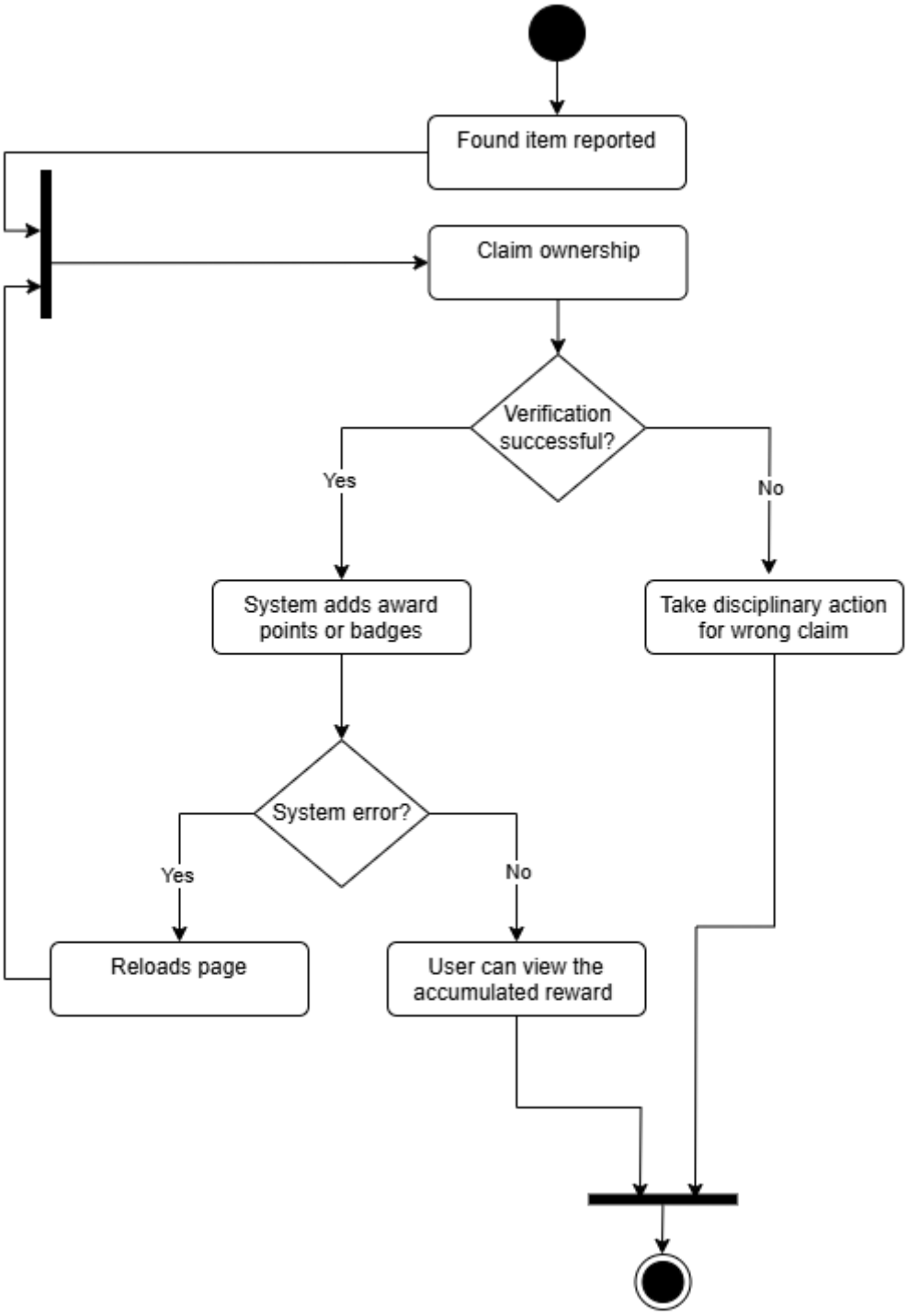
FR12 - Emergency Lost Item Alert



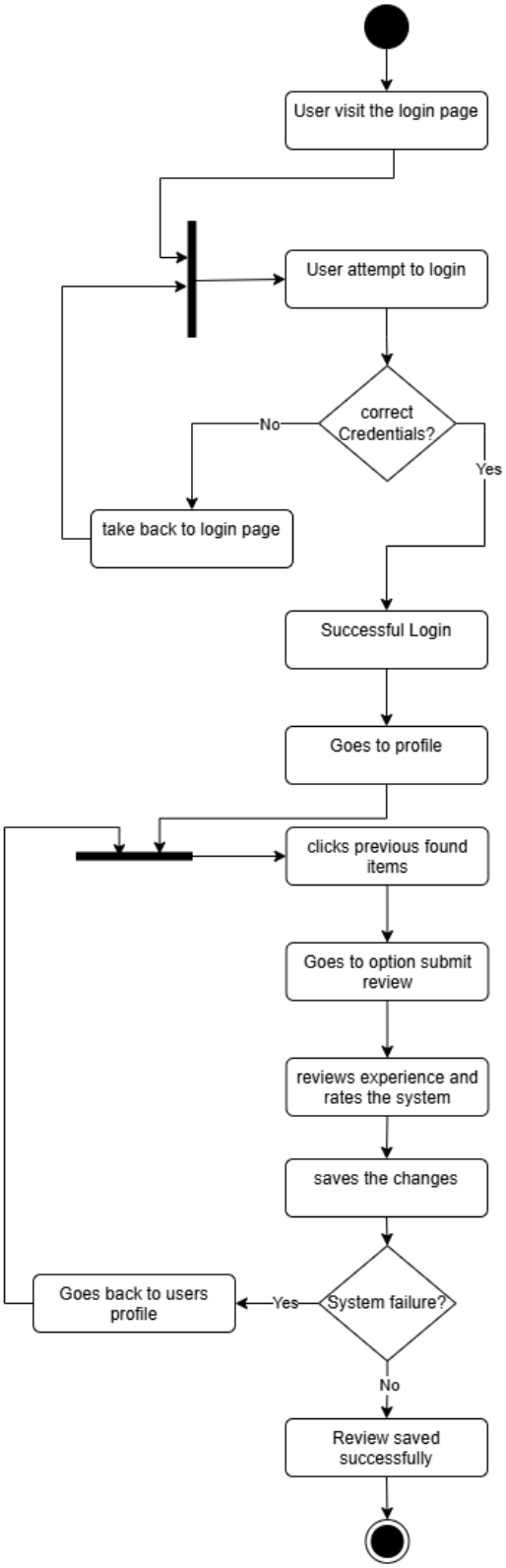
FR 13 - Chat or Contact System



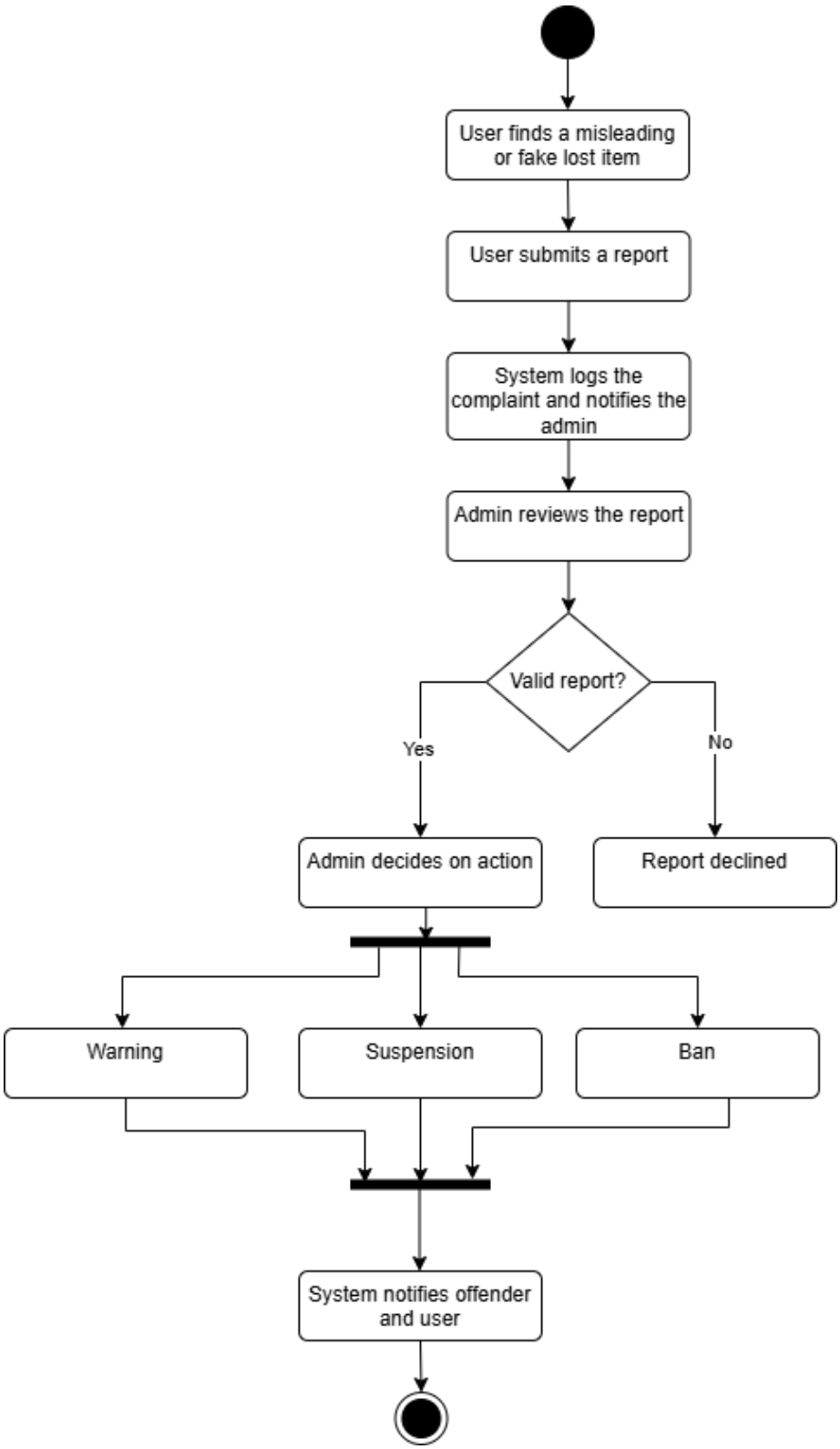
FR 14 – Reward System for Finders



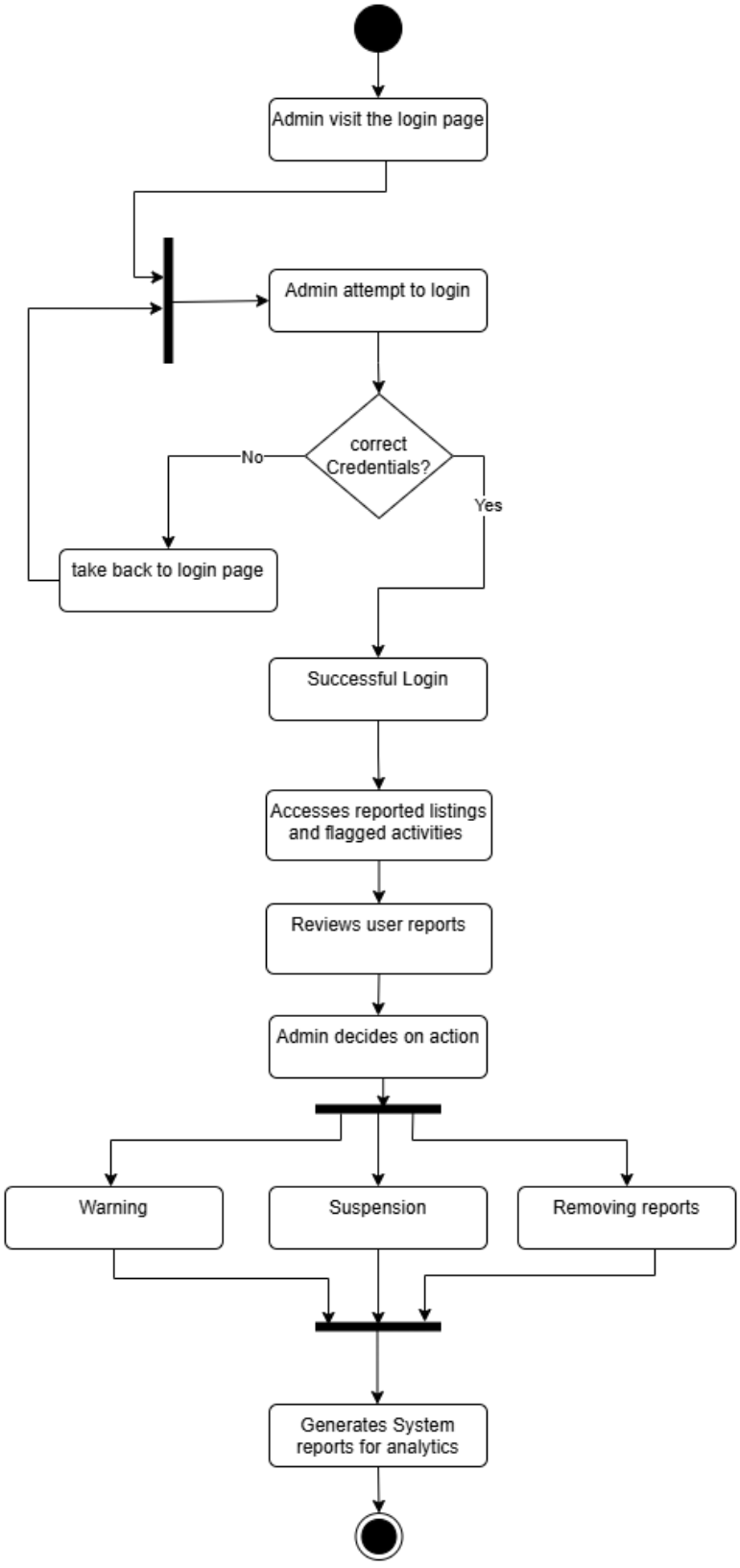
FR 15 – Feedback and Review System



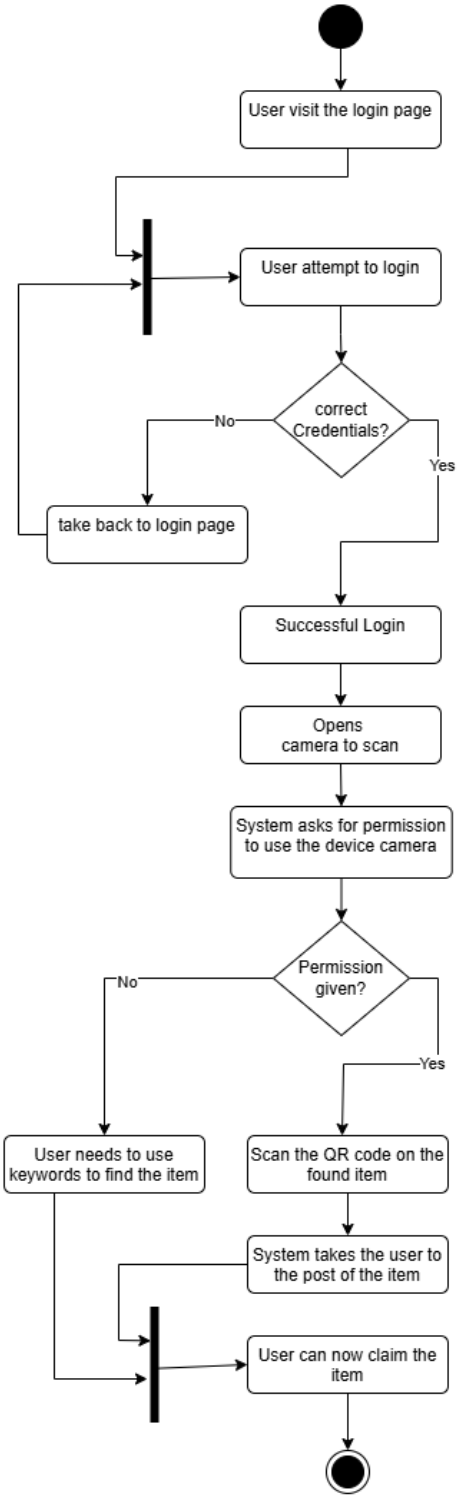
FR 16 – Reporting Fake or Incorrect Listings



FR 17 – Admin Panel for Moderation

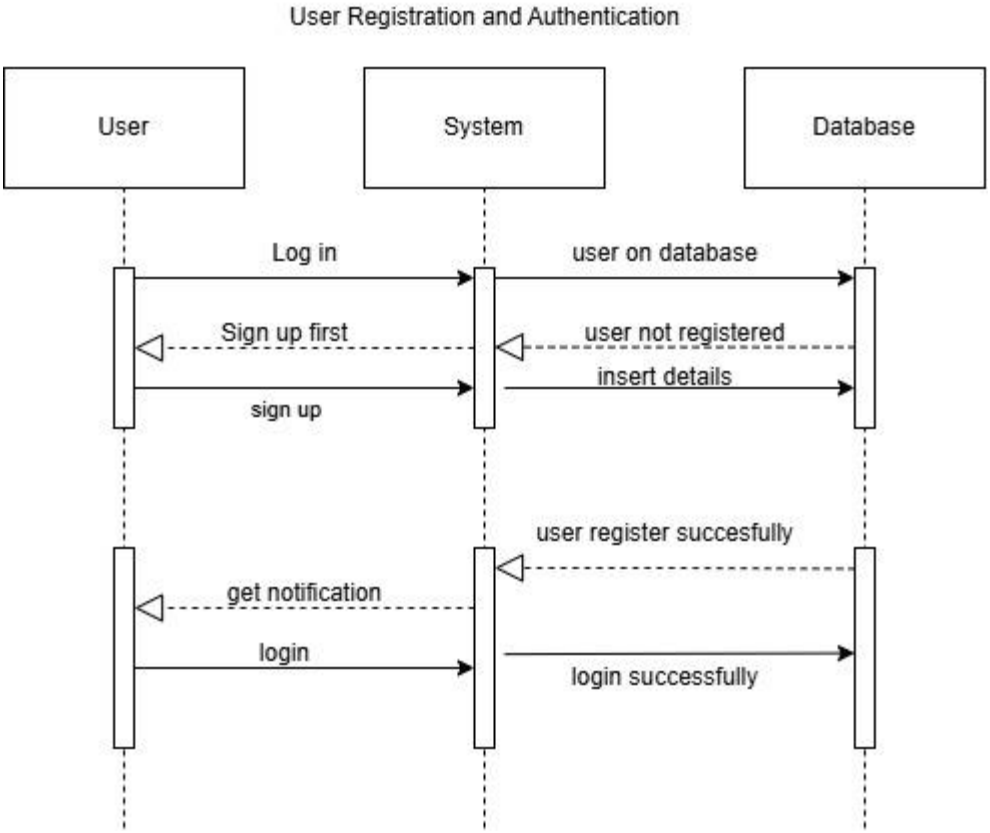


FR 18 – QR Code for Item Verification

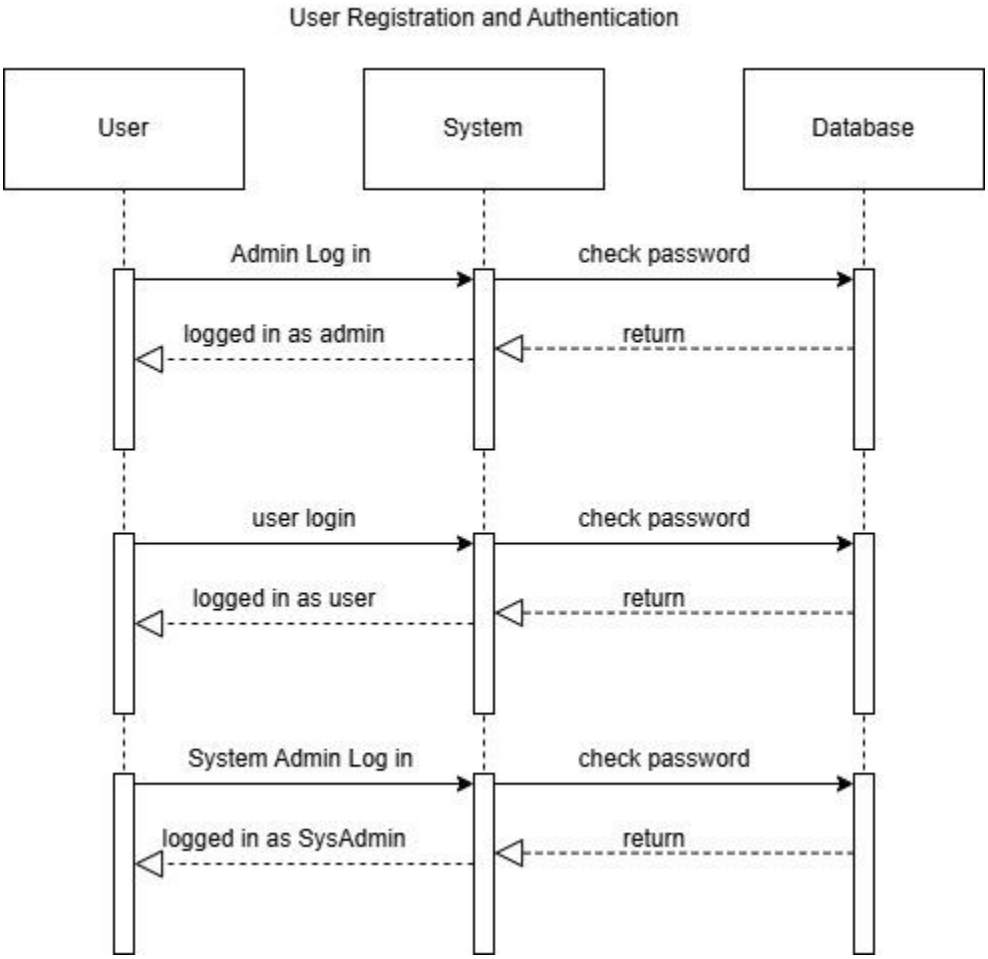


Sequence Diagram

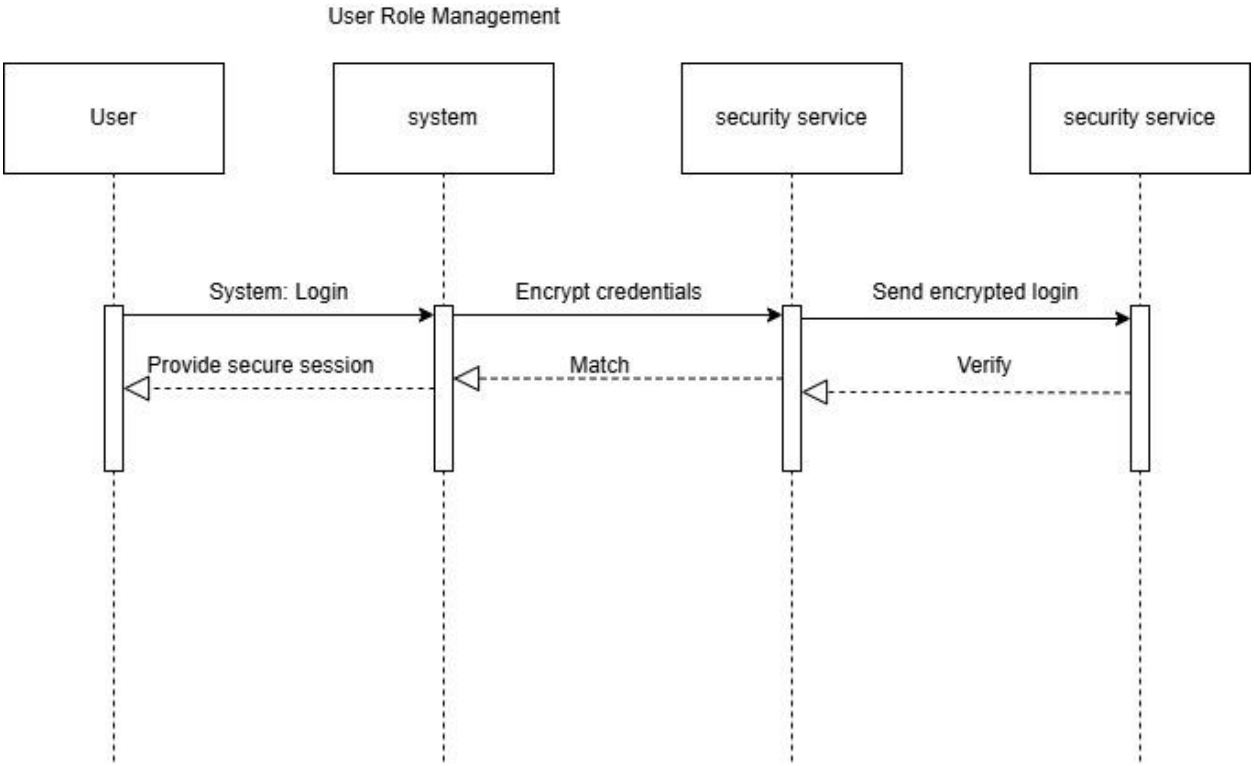
1. User Registration and Login



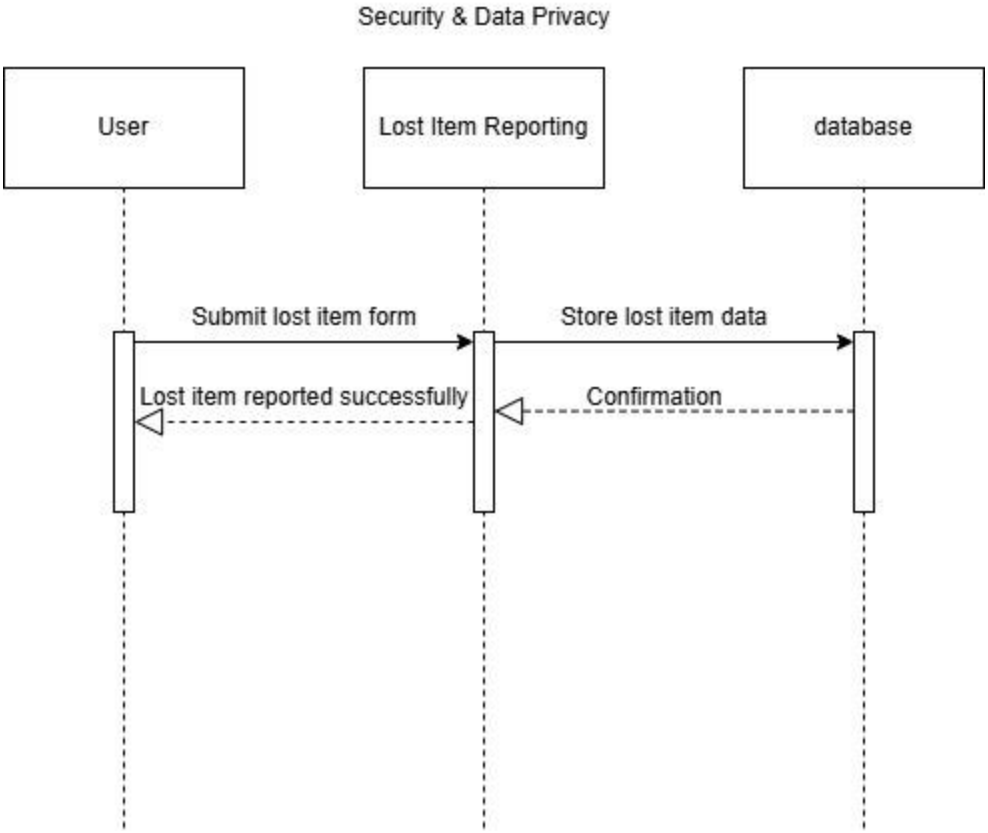
2. User Role Management



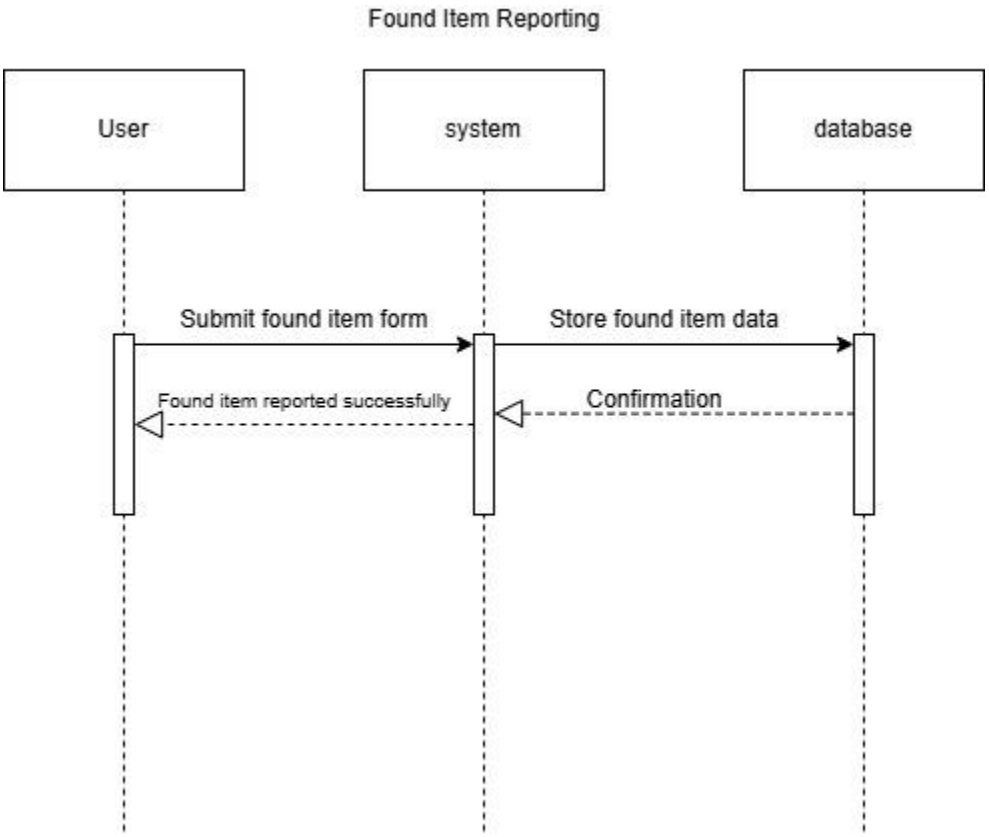
3. Lost Item Reporting



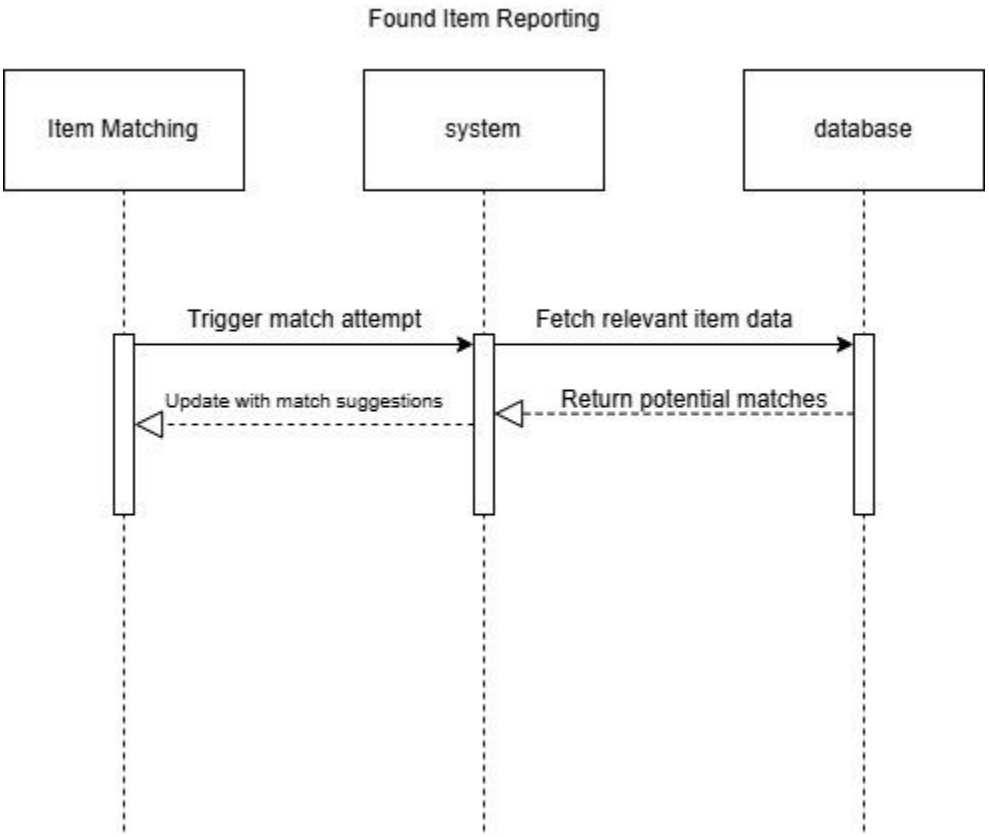
4. Found Item Reporting



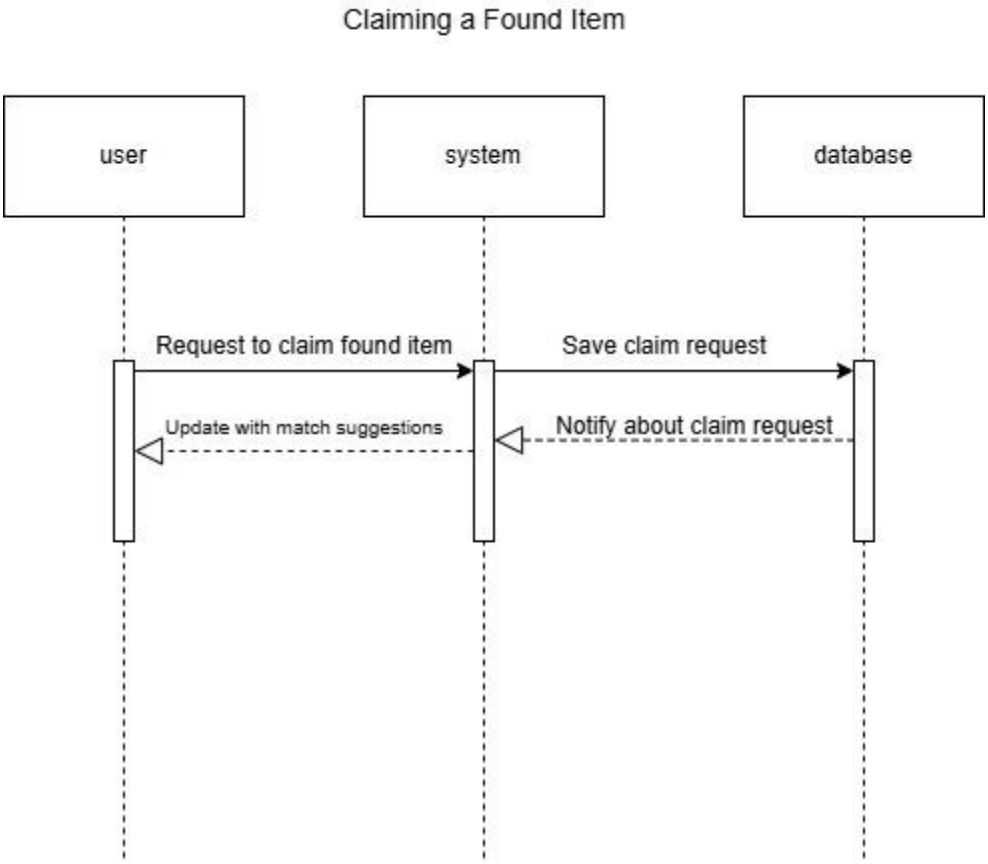
5. Lost and Found Item Matching



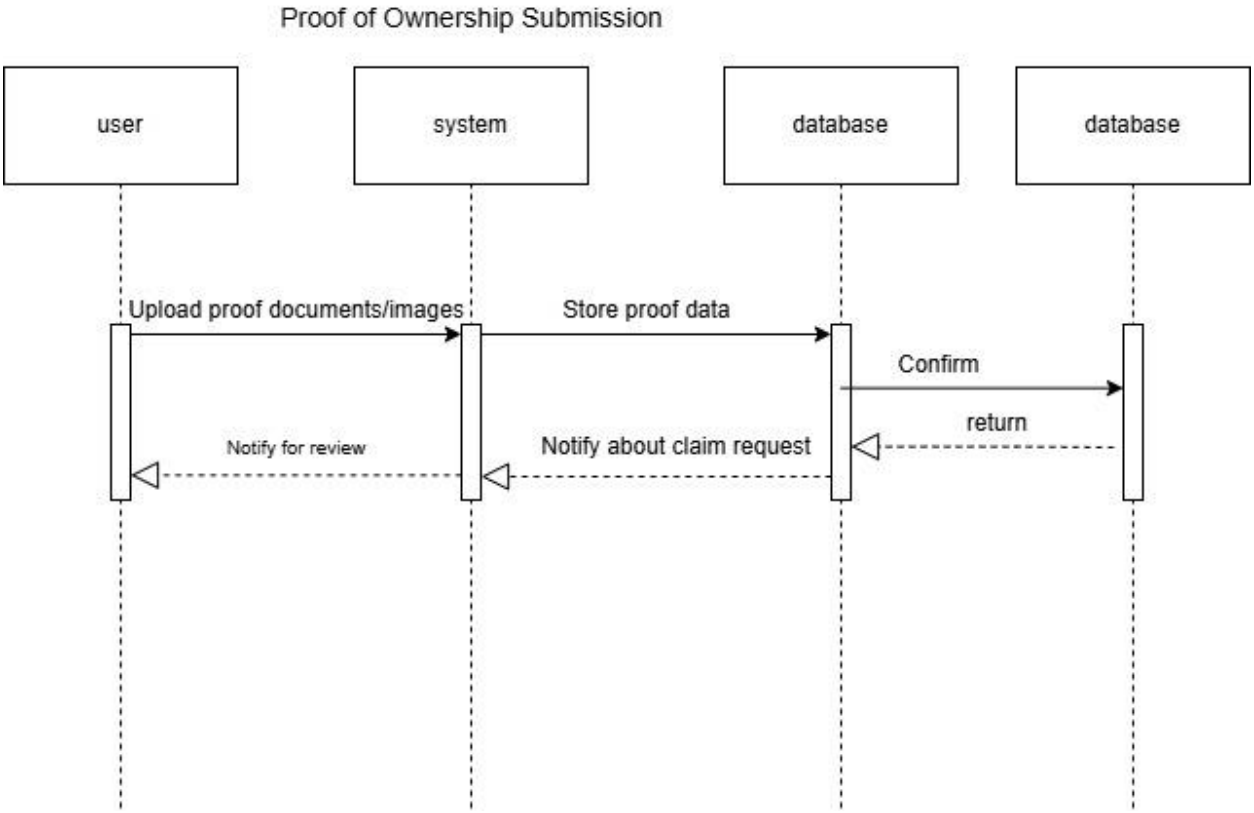
6. Claiming a Found Item



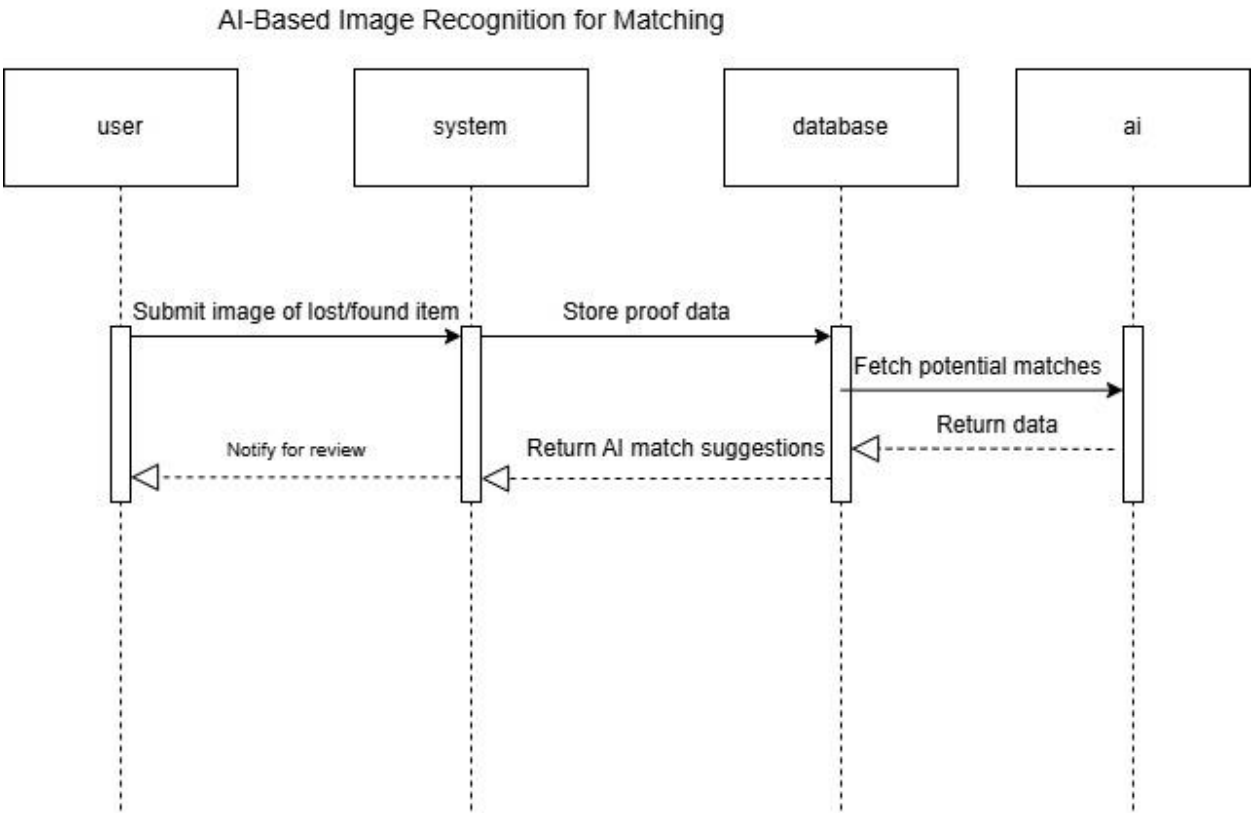
7. Proof of Ownership Submission



8. AI-Based Image Recognition for Matching

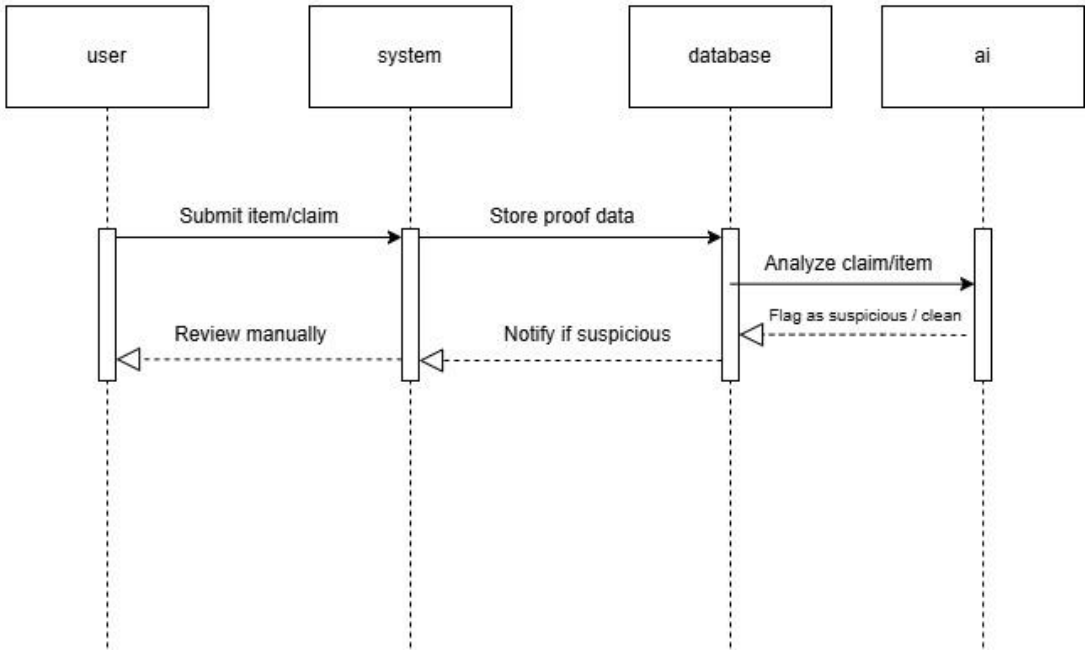


9. AI-Powered Fraud Detection



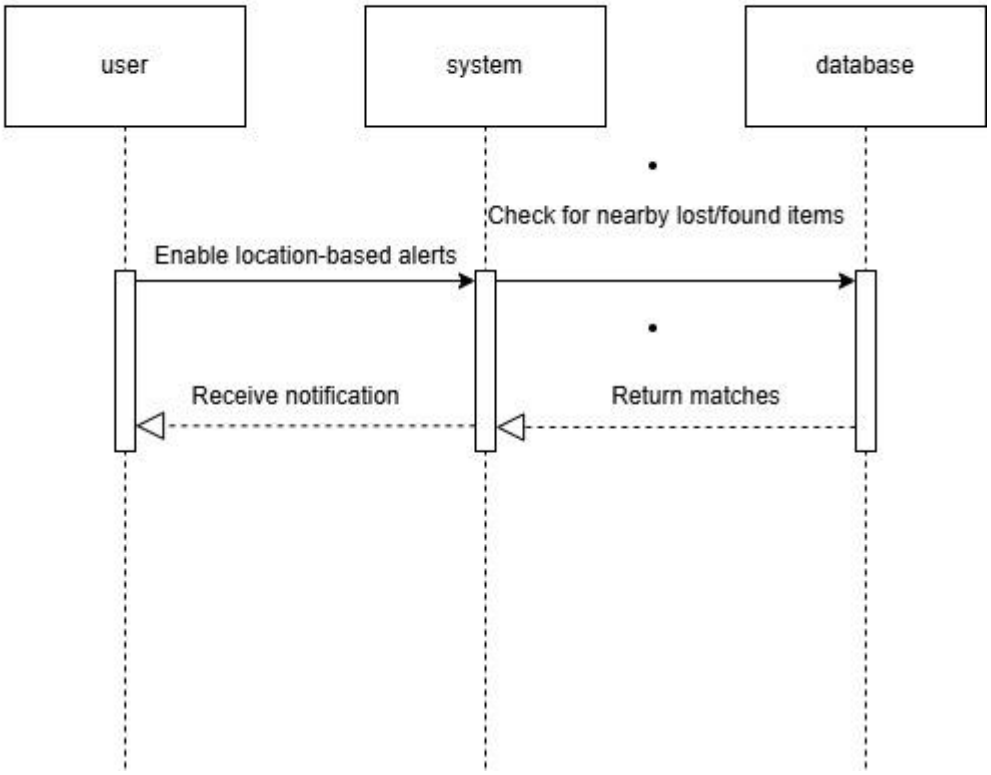
10. Smart Location-Based Alerts

FR10: AI-Powered Fraud Detection



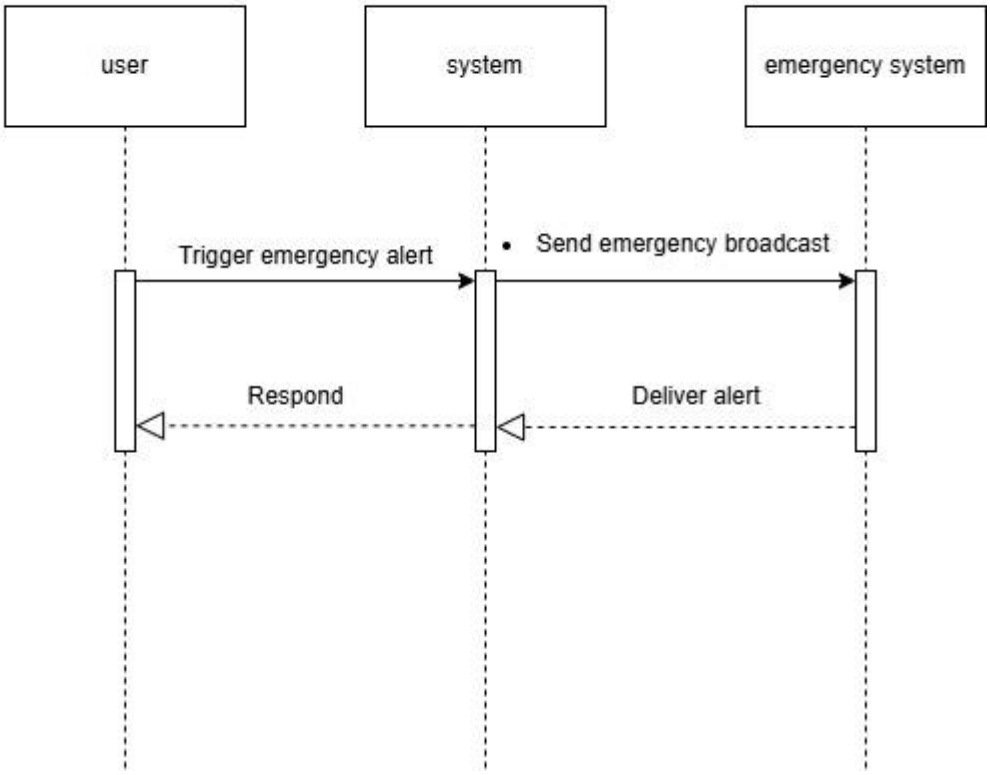
11. Emergency Lost Item Alert

FR11: Smart Location-Based Alerts

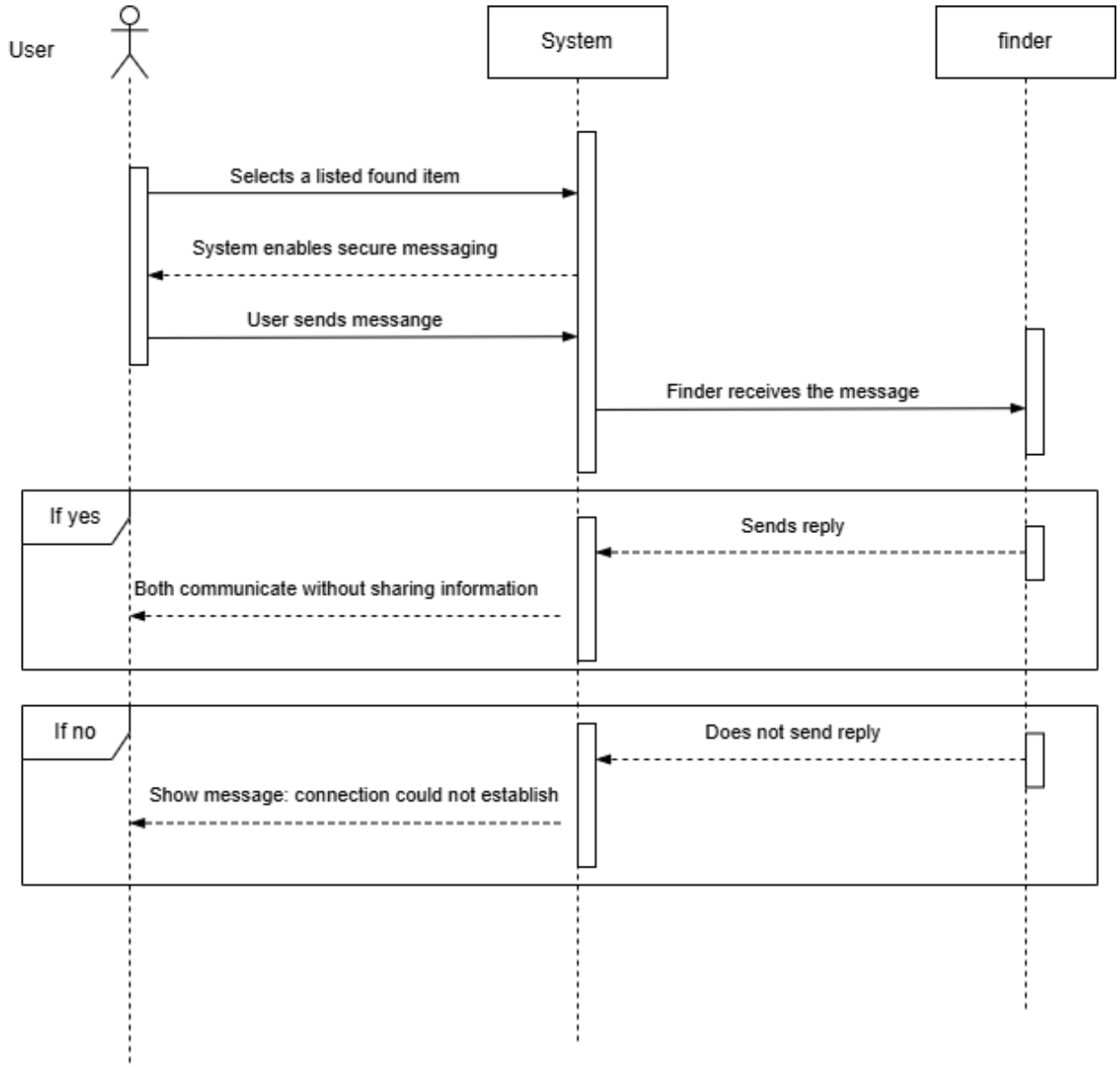


12. Security & Data Privacy

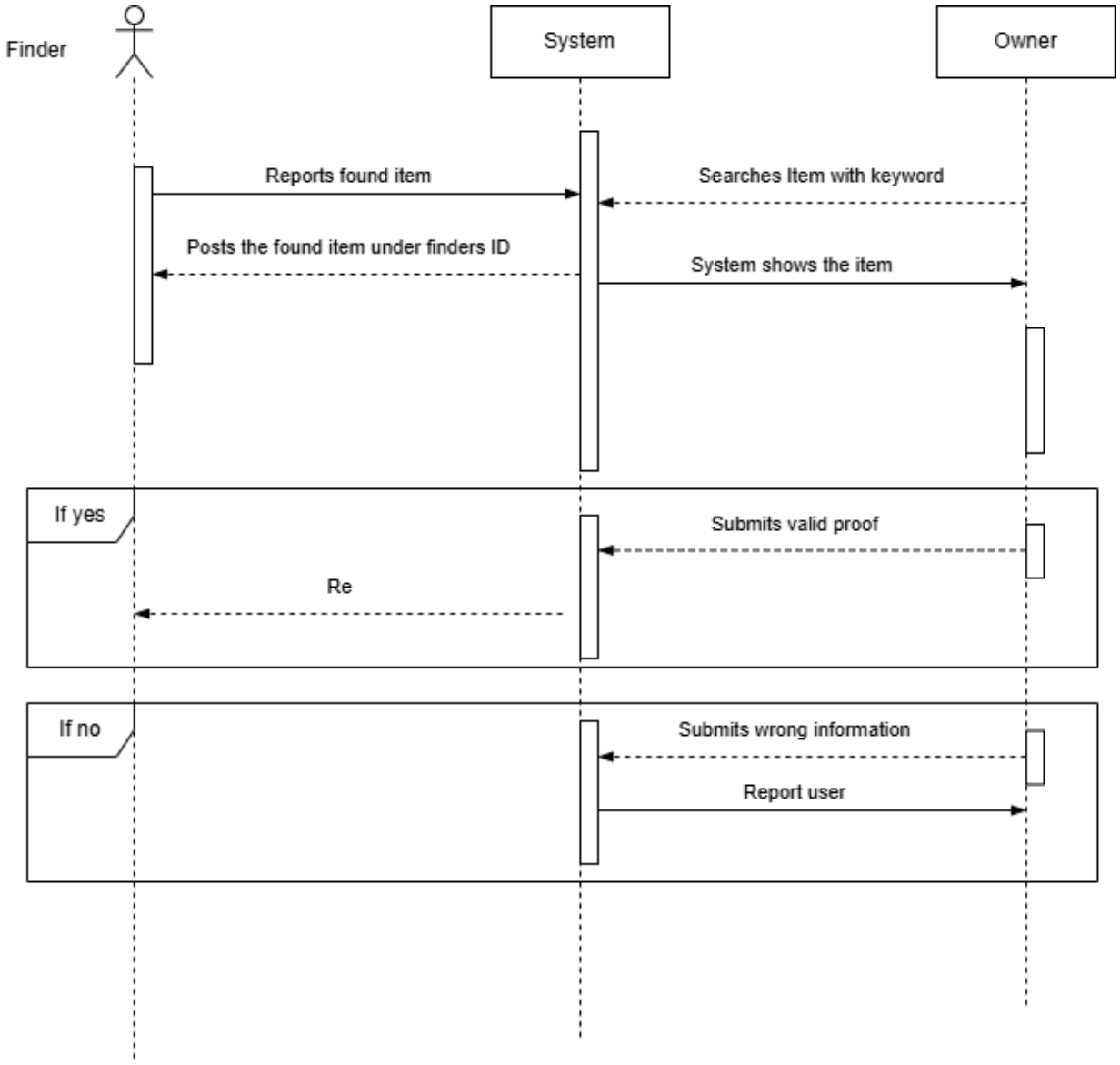
FR12: Emergency Lost Item Alert



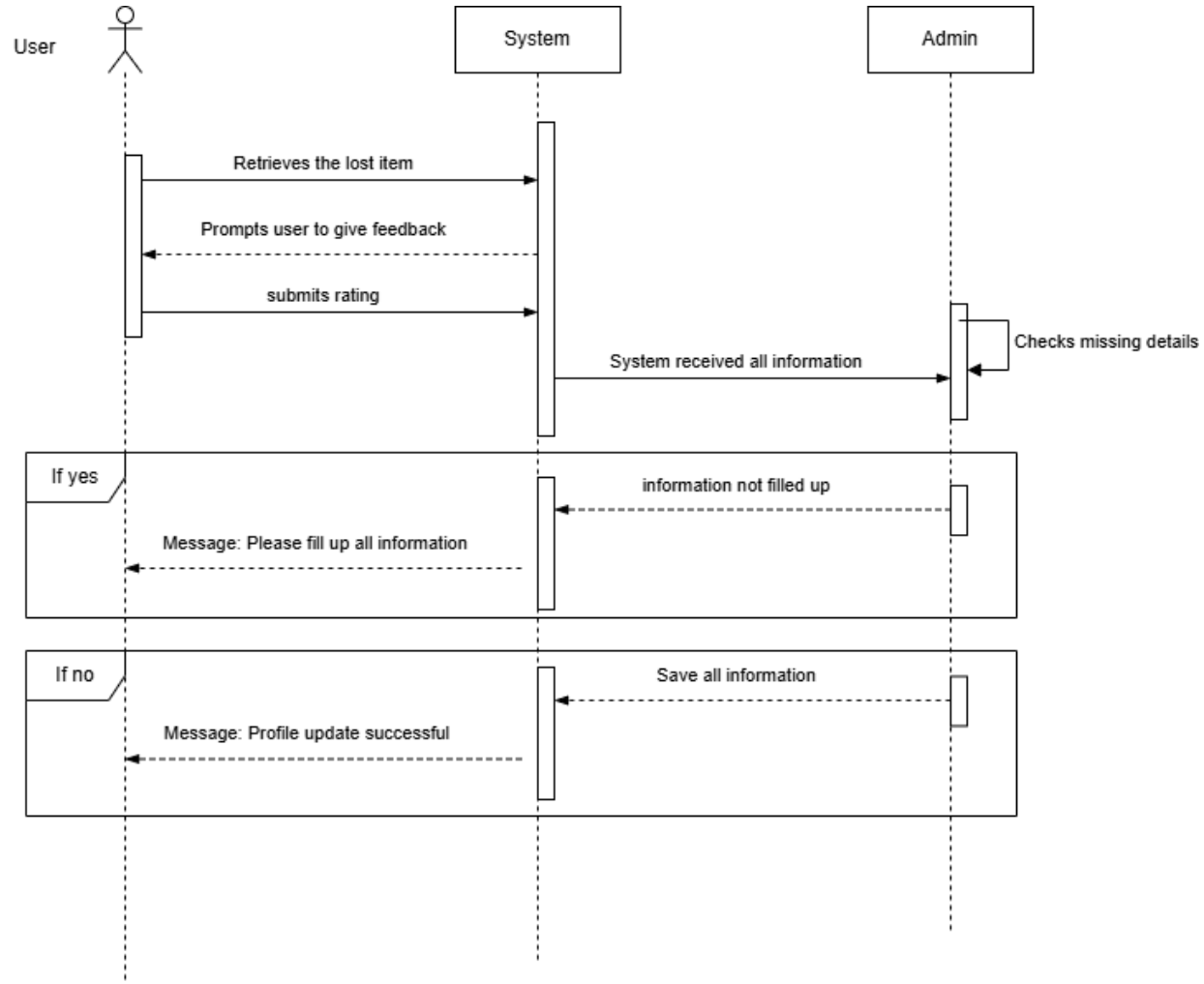
13. Chat or Contact System



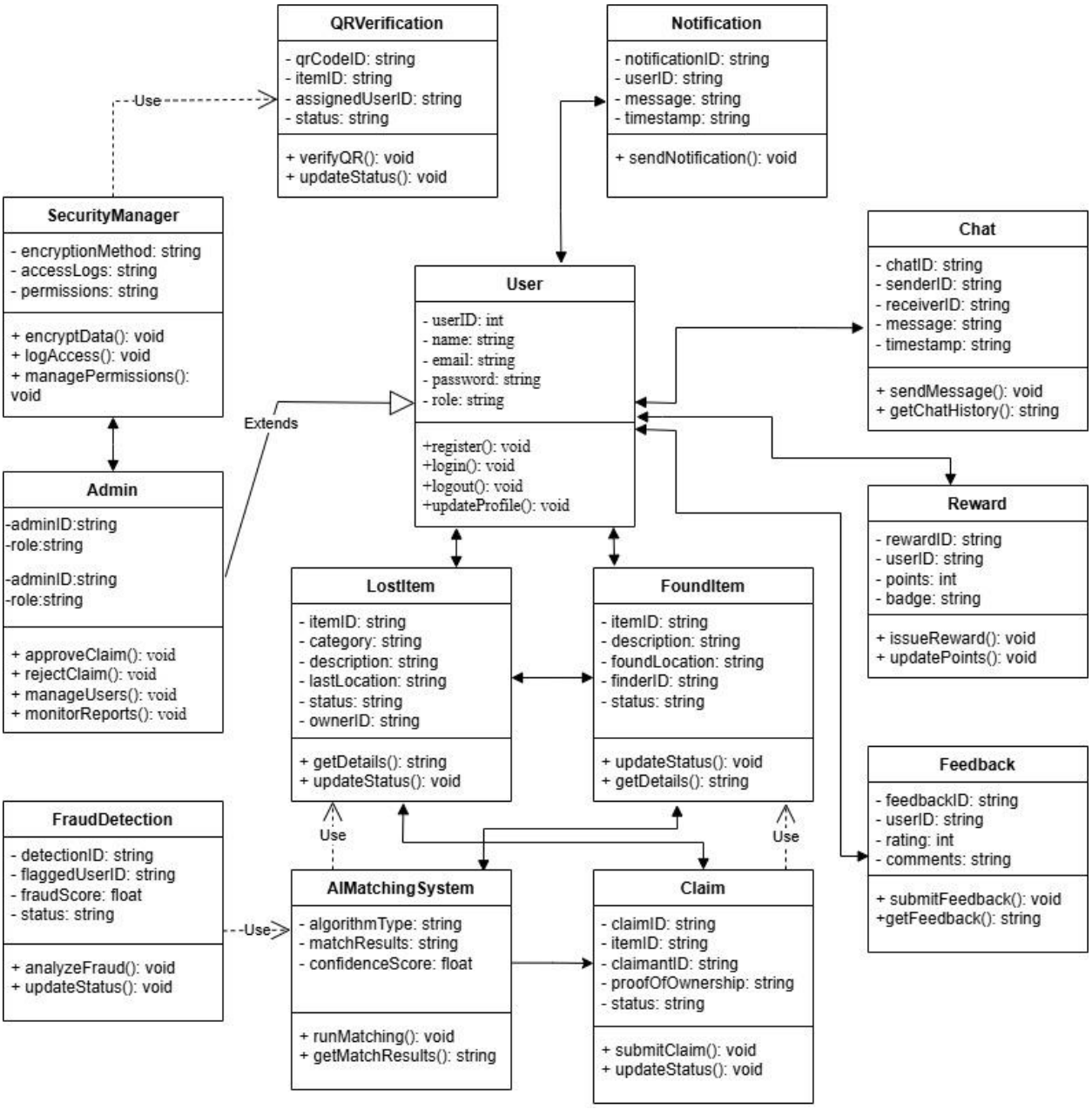
14. Reward System for Finders



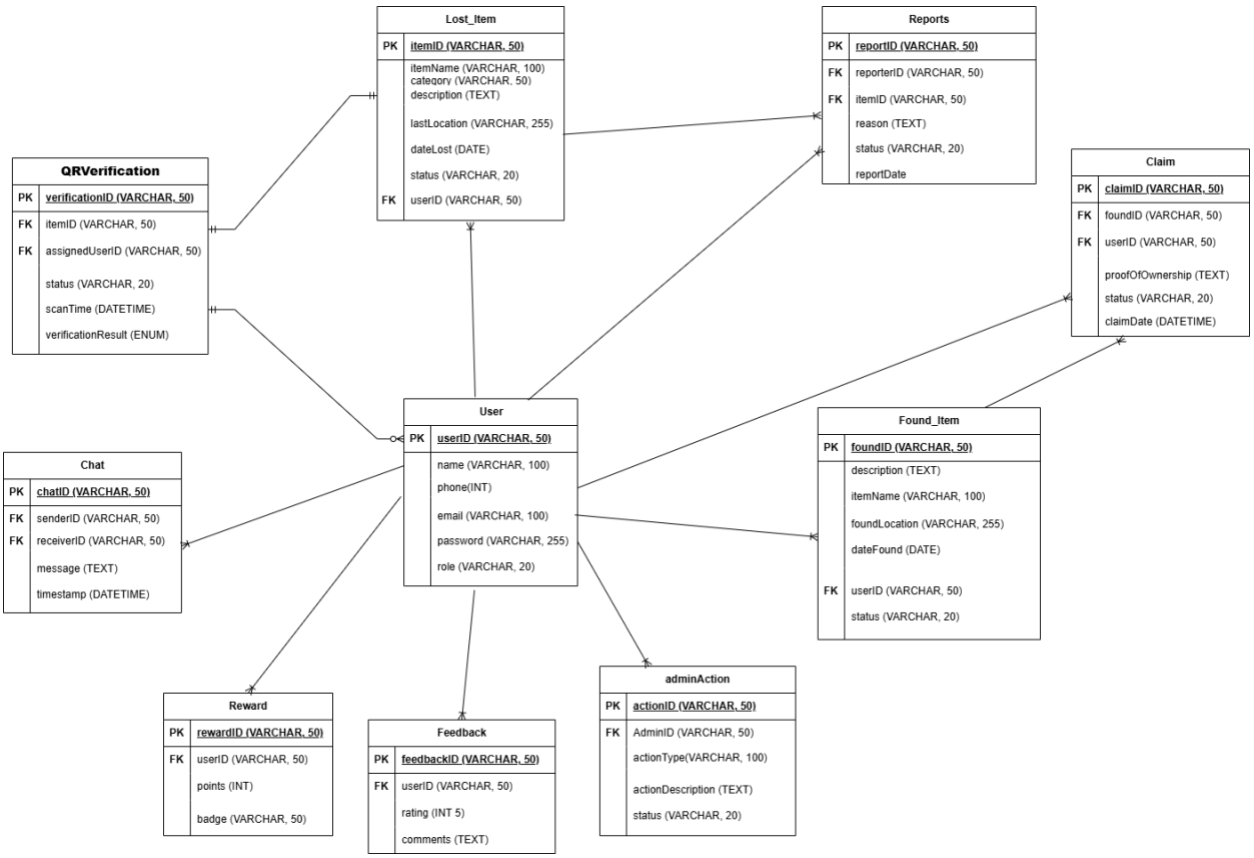
15. Feedback & Review System



Class Diagram



Entity Relationship Diagram



Prototype

Prototype Link

https://www.figma.com/proto/pF4EvZqYkkBlFAxtumdDCK/Lost-and-Found-System?node-id=1-967&p=f&t=ZWc19d4REmSCDSeq-1&scaling=min-zoom&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=1%3A967