VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM 590014

Software Engineering

Report on

LOST AND FOUND

By

Navneeth K.S(1BM22CS174)
Nishanth K.S(1BM22CS183)
Nithin Koushik P.V(1BM22CS185)
Pranav Srinivas(1BM22CS203)

Under the Guidance of
Sneha S Bagalkot
Assistant Professor, Department of CSE
B.M.S. College of Engineering
Software Engineering carried out at

Department of Computer Science and Engineering
B.M.S. College of Engineering
(Autonomous college under VTU)
P.O. Box No.: 1908, Bull Temple Road, Bangalore-560 019
2024-2025

B.M.S. COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND

ENGINEERING

CERTIFICATE

This is to certify that the Software Engineering (23CS4PCSED) titled "**LOST AND FOUND**" has been carried out by **Navneeth K.S(1BM22CS174),Nishanth K.S(1BM22CS183), Nithin Koushik P.V(1BM22CS185), Pranav Srinivas(1BM22CS203)** during the academic year 2024-2025.

Signature of the guide
Guide Name: Sneha S Bagalkot

Designation: Assistant Professor

Department of Computer Science and Engineering

B.M.S. College of Engineering, Bangalore

B.M.S. COLLEGE OF ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We,**Navneeth K.S(1BM22CS174)**, **Nishanth K.S(1BM22CS183),Nithin Koushik P.V(1BM22CS185), Pranav Srinivas(1BM22CS203) ,** students of 4thSemester, B.E, Department of Computer Science and Engineering, B.M.S. College of Engineering, Bangalore, hereby declare that, this Software Engineering Alternative Assessment entitled **"LOST AND FOUND"** has been carried out by us under the guidance of Sneha S Bagalkot, Assistant Professor, Department of CSE, B.M.S. College of Engineering, Bangalore during the academic semester April-August 2024.

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

Navneeth K.S(1BM22CS174)
Nishanth K.S(1BM22CS183)
Nithin Koushik P.V(1BM22CS185)
Pranav Srinivas(1BM22CS203)

Table of Contents

LOST AND FOUND

2.INTRODUCTION

Introducing our Lost and Found App, a user-friendly platform designed to reconnect people with their misplaced belongings efficiently. Whether you've lost your wallet, keys, or other valuable items, our app provides a comprehensive solution for reporting and locating lost possessions. With detailed item descriptions, photos, and location information, users can easily browse or report lost and found items. The app features advanced search functionality, real-time notifications, and robust security measures to protect user data. By fostering a supportive community, our Lost and Found App aims to bring peace of mind to those seeking to recover their lost items.

Primary Features:

- **User-Friendly Interface:** Our app is designed with a clean and intuitive interface, ensuring a hassle-free experience for users of all ages.
- **Detailed Item Listings:** Each lost and found item entry includes comprehensive details, such as descriptions, photos, and the location where the item was last seen or found.
- **Efficient Search Functionality:** Quickly search for lost items using keywords, categories, or locations to narrow down potential matches.
- **Real-Time Notifications:** Receive instant updates and notifications when a found item matches your reported lost item.
- **Secure and Private:** User data is protected with robust security measures to ensure privacy and confidentiality.
- **Community Engagement:** Foster a supportive community by enabling users to report found items and help others in their search for lost belongings.

How It Works:

1. **Report a Lost Item:** Enter detailed information about your lost item, including a description, photos, and the location where it was last seen.
2. **Browse Found Items:** Explore the list of found items posted by other users, complete with descriptions and photos to help you identify your lost possession.
3. **Claim or Report Items:** If you find a match, contact the person who found your item directly through the app. Similarly, if you find an item, you can report it to help the rightful owner locate it.

3.SOFTWARE REQUIREMENT SPECIFICATION(SRS)

3.1 Project Description

3.1.1 Purpose

The purpose of our Lost and Found App is to provide a reliable and efficient platform for individuals to report, search for, and reclaim lost items. By facilitating the connection between those who have lost belongings and those who have found them, the app aims to reduce the stress and inconvenience associated with misplaced possessions. Our goal is to foster a supportive community where users can assist each other, ultimately enhancing the chances of recovering lost items quickly and securely.

3.1.2 Scope

The scope of our Lost and Found App is comprehensive, encompassing a range of functionalities designed to make the process of reporting and recovering lost items as seamless as possible. Users can create accounts, manage their profiles, and securely store their contact information. The app allows users to report lost or found items with detailed descriptions, photos, and the locations where items were last seen or found

3.1.3 Objectives

Streamline the Recovery Process: Provide a user-friendly platform that simplifies the process of reporting, searching for, and recovering lost items.

Enhance Connectivity: Facilitate communication between users who have lost items and those who have found them, fostering a supportive community.

Increase Recovery Rates: Utilize advanced search capabilities and real-time notifications to increase the likelihood of lost items being found and returned to their owners.

Ensure User Security and Privacy: Implement robust security measures to protect user data and maintain privacy throughout the recovery process.

Promote Community Engagement: Encourage users to report found items and assist others in their search for lost belongings, creating a collaborative environment.

3.1.4 Target Audience

Individuals: Anyone who has lost or found personal belongings, such as wallets, keys, phones, or other valuables, and is seeking a reliable platform to report and recover these items.

Students: Students who frequently lose items such as textbooks, laptops, or personal accessories on school or college campuses.

Commuters: People who travel regularly using public transportation and may misplace items during their commute.

3.2. Problem Statement

The problem addressed by our Lost and Found App is the frequent and often frustrating occurrence of losing personal belongings in various public and private spaces. Individuals commonly misplace items such as wallets, keys, phones, and personal accessories, leading to stress, inconvenience, and sometimes significant financial loss. Existing solutions for recovering lost items are often fragmented, inefficient, or reliant on physical lost and found offices, which may not be accessible or effective.

3.3. System Requirements

3.3.1. Functional Requirements

User Registration and Login:

- Users must be able to register an account using an email address, phone number, or social media accounts.
- Users must be able to log in to their accounts securely.

Item Reporting:

- Users must be able to report lost items by providing details such as item description, photos, and last known location.
- Users must be able to report found items with similar details and the location where the item was found.

Item Listings:

- The app must display a list of reported lost and found items with relevant details and photos.
- Users must be able to view detailed information about each item, including descriptions, photos, and locations.

Communication:

- Users must be able to contact the person who found their lost item through secure messaging within the app.
- Users must be able to respond to inquiries about items they have reported as found.

Security and Privacy:

- The app must implement secure authentication and data encryption to protect user information.
- Users must have control over their privacy settings and data sharing preferences.

3.3.2 Non-Functional Requirements

Performance:

- The app must load the main user interface within 2 seconds.
- Search results must be displayed within 3 seconds after a query is submitted.
- Notifications must be delivered to users within 5 seconds of a matching item being reported.

Scalability:

- The app must be able to handle a large number of concurrent users without performance degradation.
- The backend infrastructure must support scaling up to accommodate increasing numbers of item reports and searches.

Reliability:

- The app must have an uptime of 99.9%, ensuring it is available to users at all times.
- The app must include mechanisms for data redundancy and failover to ensure continuous operation during server failures.

Security:

- All user data must be encrypted in transit and at rest.
- The app must use secure authentication methods, including multi-factor authentication (MFA).
- The app must comply with relevant data protection regulations (e.g., GDPR, CCPA).

Usability:

- The app must have an intuitive and user-friendly interface, accessible to users of all technical skill levels.
- The app must provide clear instructions and feedback to guide users through reporting and searching for items.
- The app must be accessible to users with disabilities, following WCAG 2.1 guidelines.

3.3.3. Domain Requirements

.User Authentication and Authorization:

- The app must allow users to create accounts using valid email addresses, phone numbers, or social media accounts.
- The app must implement role-based access control, ensuring that only authorized users can access specific features (e.g., admin dashboard).

Item Classification:

- The app must support categorizing items into predefined categories (e.g., electronics, clothing, accessories) to facilitate easy searching and reporting.
- Each category must have specific attributes (e.g., brand, color, size) to enhance item descriptions and searches.

Data Privacy and Protection:

- The app must comply with data protection regulations (e.g., GDPR, CCPA) to safeguard user information.
- Users must have control over their data, including options to delete their accounts and associated data.

Reporting and Verification:

- Users must be able to report lost and found items with detailed descriptions and photos.
- The app must implement verification processes to validate the authenticity of reports, reducing fraudulent activity.

3.4 Appendices

3.4.1. Appendix A: Glossary of Terms

1. User: An individual who uses the Lost and Found App to report, search for, or recover lost items.
2. Admin: A user with elevated permissions to manage the app, including user accounts, item reports, and resolving disputes.
3. Lost Item: An item reported by a user as missing.
4. Found Item: An item reported by a user as discovered, which may belong to another person.
5. Report: A submission made by a user detailing a lost or found item.

3.4.2 Appendix B: Analysis Models

- Use Case Diagrams
- Sequence Diagrams

4. ARCHITECTURAL DESIGN

This section outlines the architectural design models for the Lost and Found mobile application, ensuring a robust, scalable, and maintainable system. The design includes various models such as

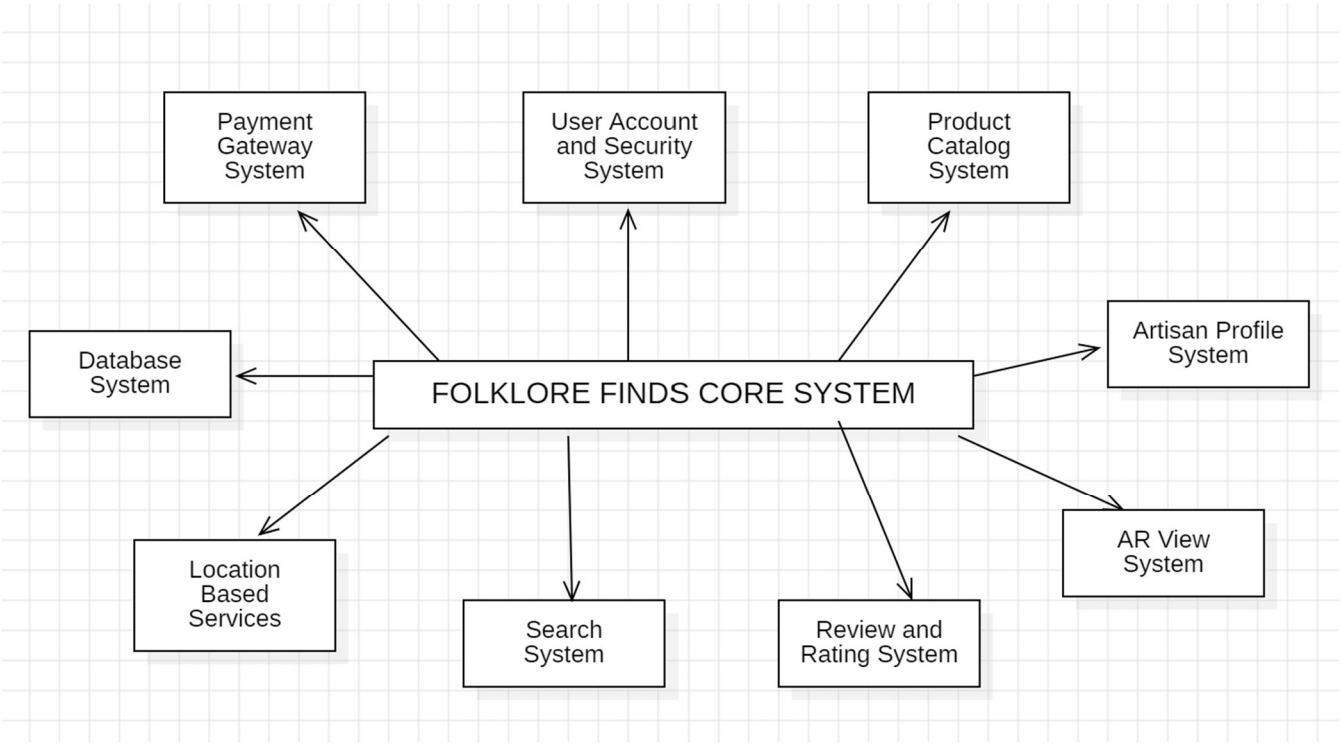
the context model, structural model, and control model. The system architecture for the Lost and Found App is based on a client-server model with a focus on modularity, scalability, and security. The primary components include the mobile app (client), backend server, database, and third-party services. This architecture ensures efficient communication between users, secure data management, and seamless integration with location-based services and notification systems.

4.1 CONTEXT MODEL

4.1.1 Introduction

The context model provides an overview of the interactions between the Lost and Found mobile application and its external entities. This model helps in understanding the system boundaries and the external systems and users that interact with the application.

4.1.2. Context Model



4.1.2 Description

Central System: Lost and Found Core System The central hub managing the app's core functionalities, including user interactions, item reporting and searching, notification delivery, and database management.

Connected Systems:

Item Reporting System

- Manages the inventory of reported lost and found items.
- Updates item listings, statuses, and detailed descriptions.

User Profile System

- Stores and updates profiles of users.
- Integrates user details into item reports and communications.

Notification System

- Provides real-time notifications for matched items and updates.
- Integrates with the core system to alert users about relevant activity.

Search and Filtering System

- Handles the search functionality within the app.
- Provides advanced filtering options to help users find items efficiently.

Location-Based Services System

- Integrates with the location features of the app.
- Provides geolocation data to help users pinpoint lost and found items.

User Account and Security System

- Manages user account information, authentication, and security.
- Ensures secure transactions and data protection.

Messaging System

- Facilitates secure in-app communication between users.
- Allows users to arrange the return of found items and verify ownership.

Database System

- Stores all app-related data, including item reports, user data, and transaction history.
- Ensures data integrity and accessibility for other systems.

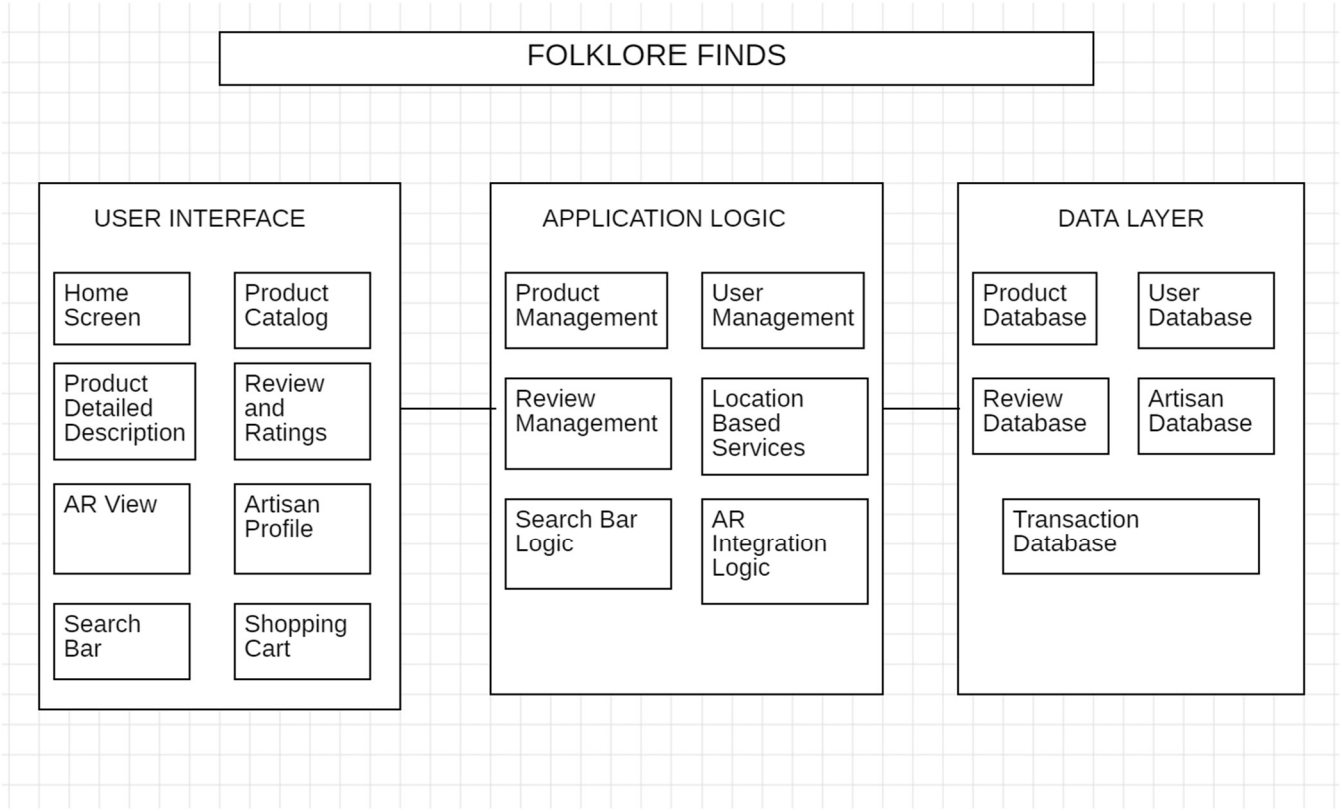
The context model for the Lost and Found App outlines the key interactions between the application and its external entities, ensuring a clear understanding of the system boundaries. This model serves as a foundation for designing and developing the application, ensuring all necessary interactions are accounted for.

4.1 STRUCTURAL MODEL

4.2.1 Introduction

This model outlines the main components, their subcomponents, and the interactions between them, representing the architecture of the application.

4.2.2 Structural Model



4.1.2 Description

User Interface: The user interacts with various UI components such as the Home Screen, Item Reporting, Item Detail Page, Search Bar, Notifications, User Profiles, and Messaging System. User actions such as reporting a lost or found item, searching for items, viewing item details, and communicating with other users trigger specific processes.

Application Logic: The Application Logic layer handles the business logic and processes user actions. Components like Item Management, User Management, Search Logic, Notification Management, Location Services, and Messaging Logic process the respective tasks. This layer interacts with the Data Layer to fetch and store necessary data.

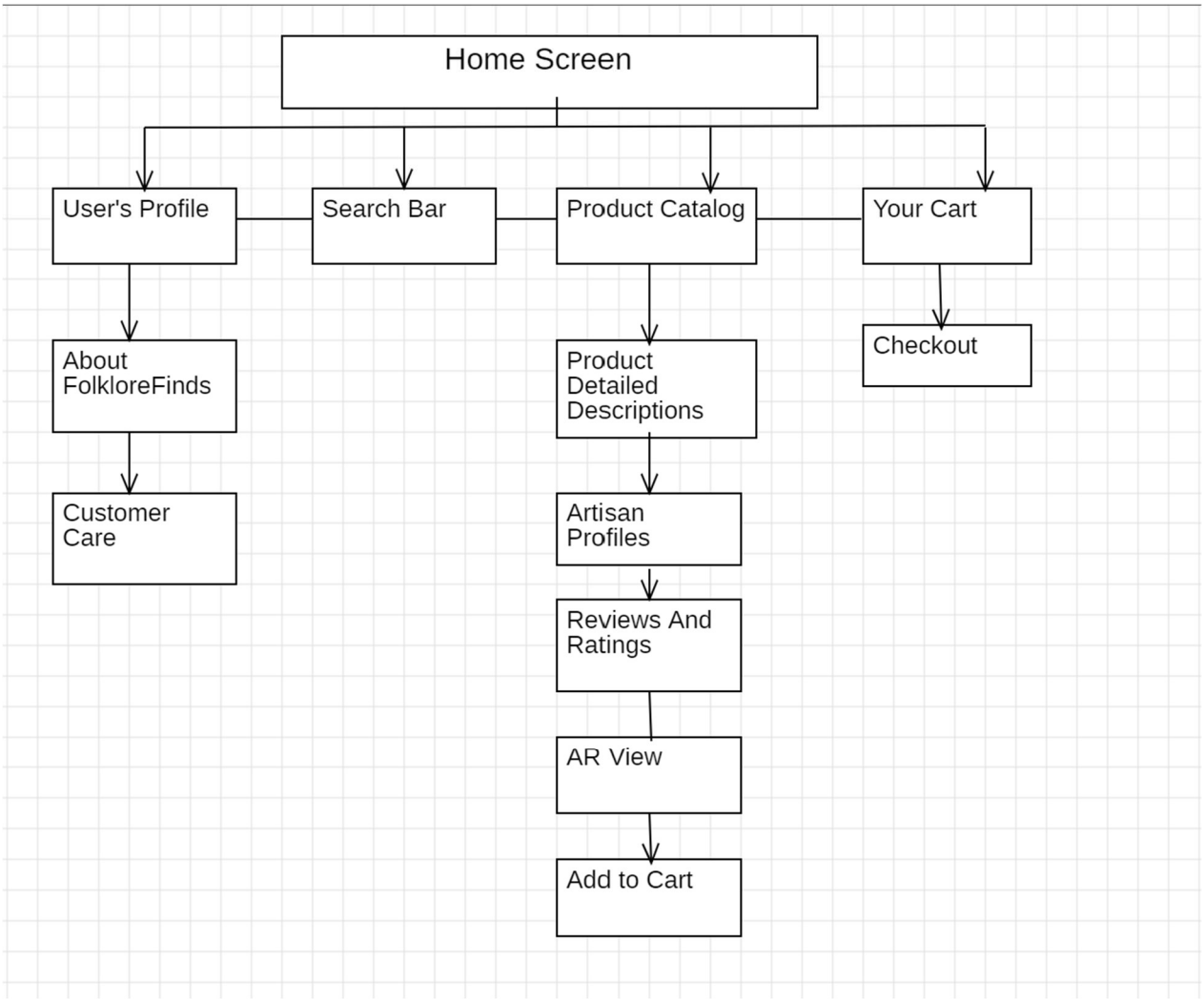
Data Layer: The Data Layer comprises various databases that store information about lost and found items, users, messages, and transactions. The Application Logic layer retrieves and updates data in these databases as required.

This structural model outlines the architecture of the Lost and Found App, detailing the main components, subcomponents, and their interactions. It represents how the app's user interface, application logic, and data layer work together to provide a seamless experience for users, allowing them to report, search for, view details, and recover lost items efficiently.

4.2 CONTROL MODEL

The control model focuses on how different components interact and control the flow of operations within the app.

4.3.1 CONTROL MODEL-EVENT DRIVEN



Event-Driven Characteristics:

User Actions Trigger Events: The model highlights how user interactions (such as reporting a lost item, searching for items, viewing item details, and communicating with other users) trigger specific events within the app. Each component of the UI layer (e.g., Home Screen, Item Reporting, Search Bar) responds to user actions, which are essentially events that initiate processes in the application logic layer.

Decentralized Control: The control logic is distributed among various components (Item Management, User Management, Search Logic, etc.), which handle specific tasks when relevant events occur. The system responds dynamically to events generated by user interactions rather than following a predefined, centralized control sequence.

Interaction Between Components: Components interact with each other based on the occurrence of events. For example, reporting a lost item triggers the addition of the item to the database and notification to relevant users, while searching for items triggers the search logic to fetch relevant results. This interaction pattern is typical of event-driven architectures, where the flow of control depends on events rather than a central controller.

Control Flow:

User Initiates Action:

- User Launches the App: The user opens the Lost and Found App and lands on the Home Screen.
    - Control: The Home Screen displays options to report lost items, search for items, and view recent activity.

User Reports a Lost Item:

- User Submits a Report: The user fills out a form to report a lost item.
    - Control: Item Reporting Logic processes the input and adds the item to the Item Database. A notification is sent to users who have reported similar items.

User Searches for Items:

- User Enters Search Query: The user inputs search criteria in the Search Bar.
    - Control: Search Logic processes the query and fetches results from the Item Database. Search results are displayed to the user.

User Views Item Details:

- User Selects an Item: The user selects an item from the search results or recent reports.
    - Control: Item Management fetches item details, including descriptions, photos, and the last known location. Data is displayed on the Item Detail Page.

User Interacts with Location Services:

- User Requests Location Details: The user views the location of lost items or nearby found items.
    - Control: Location Services fetches and displays location data from the Item Database and provides a map view.

User Communicates with Other Users:

- User Sends/Receives Messages: The user sends a message regarding a found or lost item.
    - Control: Messaging Logic processes the message and updates the Message Database. Notifications are sent to involved users.

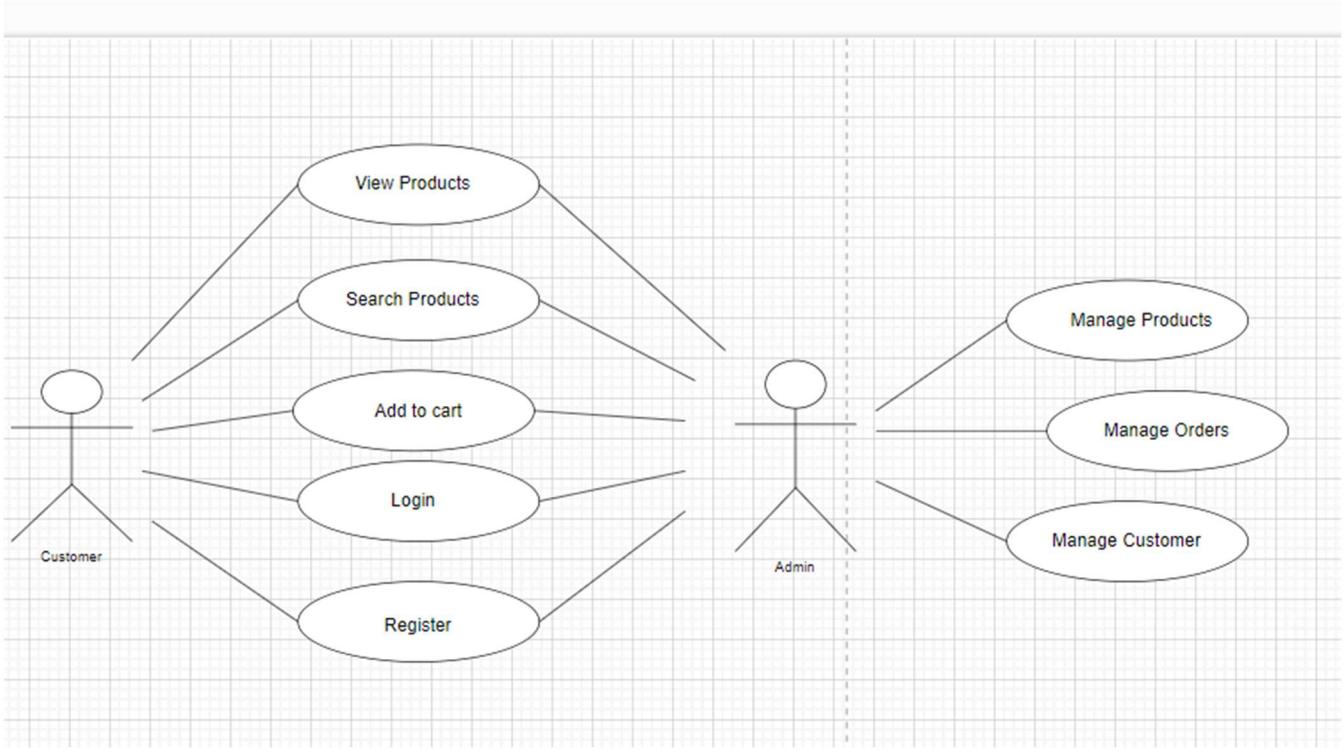
User Updates or Deletes Reports:

- User Modifies Report: The user updates or deletes a previously reported lost or found item.
    - Control: Update Logic processes changes and updates the Item Database accordingly.

This control model describes how the Lost and Found App manages user interactions through an event-driven architecture. By distributing control among various components and responding dynamically to user actions, the system ensures a responsive and efficient user experience.

5. DETAILED DESCRIPTION OF MODEL

5.1 Use Case Diagram:



Search Items:

- **Customer:** Searches for specific lost or found items using the app's search functionality.
- **Administrator:** Not typically involved in searching for items.

Report Item:

- **Student:** Reports a lost or found item by filling out the necessary details and submitting the report.
- **Administrator:** Not typically involved in reporting items.

Update Report:

- **Student:** Updates or deletes their own reports if there are changes or if the item is recovered.
- **Administrator:** Manages item reports for accuracy and handles any discrepancies.

Register/Login:

- **Student:** Registers a new account or logs into an existing account to manage their reports and interactions.
- **Administrator:** Logs into the admin panel to manage app functionalities and user accounts.

Manage Reports:

- **Administrator:** Reviews, updates, or deletes reported lost or found items as needed to ensure accuracy and relevance.

Manage Users:

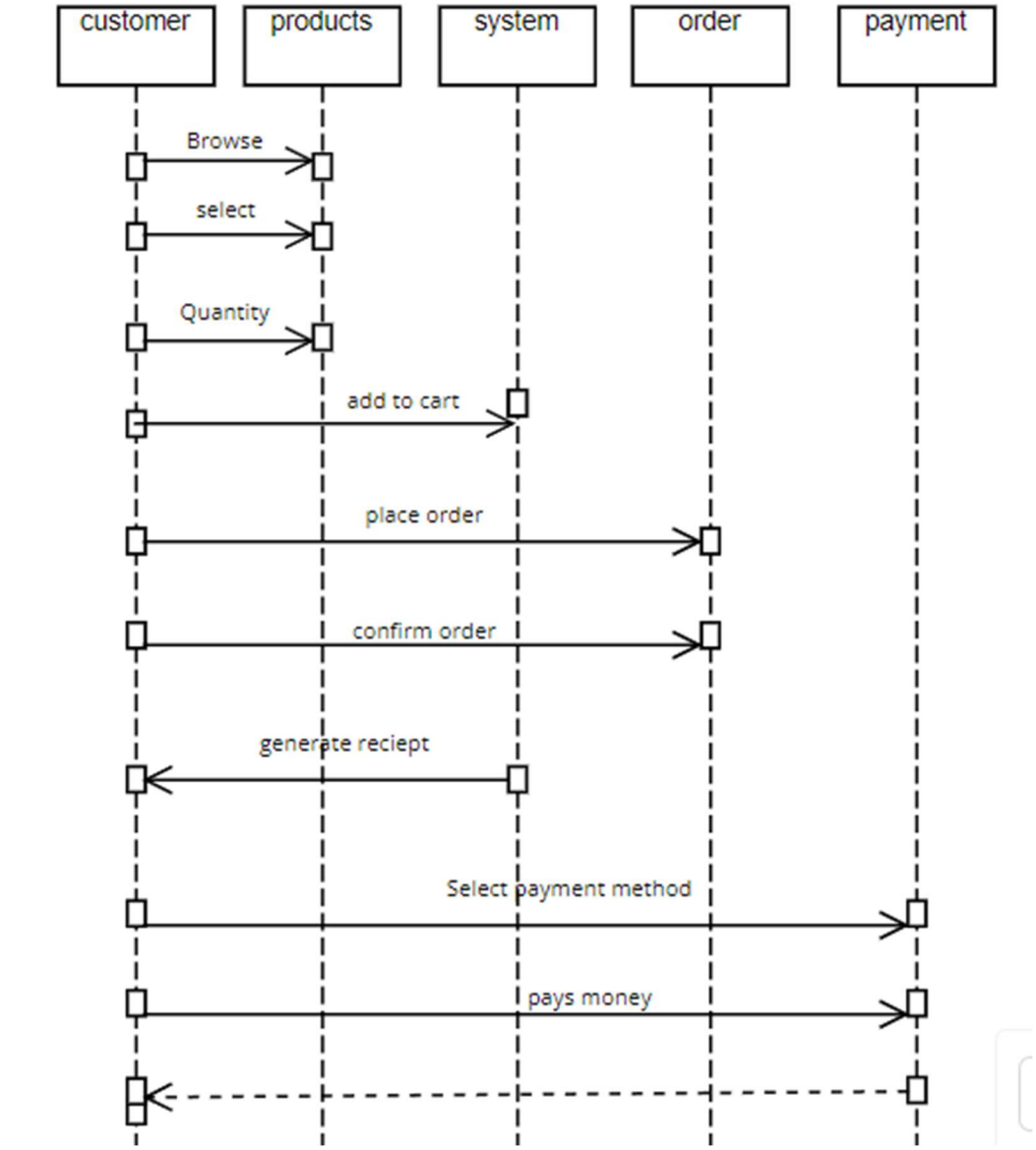
- **Administrator:** Manages user accounts, including activation, deactivation, and resolving issues.

Process Notification:

- **Notification System:** Sends alerts and notifications to users about matching items or updates on their reports.

This updated control model for the Lost and Found App outlines the responsibilities and actions of different roles within the system, demonstrating how various users interact with the app and how their actions trigger specific processes.

5.2 Sequence Diagram:


customer
products
system
order
payment
Browse
select
Quantity
add to cart
place order
confirm order
generate reciept
Select payment method
pays money

Components:

- Customer: Represents a user who interacts with the Lost and Found App to report or search for lost and found items.
- Notification System: Manages the delivery of notifications to users regarding their reports and item matches.
- Item Database: Represents the database or service that stores and manages information about lost and found items.
- Report Management System: Handles the processing and management of item reports.
- User Management System: Manages user accounts, including registration and login.

Sequence of Interactions:

Customer Reporting and Searching Process:

1. Reporting an Item:
    - Customer: Reports a lost or found item by submitting details through the app.
    - Sequence:
        1. Customer sends a request to report an item.
        2. Report Management System receives the report and stores it in the Item Database.
        3. Notification System alerts relevant users about the new report.
2. Searching for Items:
    - Customer: Searches for lost or found items using the app's search functionality.
    - Sequence:
        1. Customer sends a search query to the app.
        2. Report Management System retrieves matching items from the Item Database.
        3. Search results are displayed to the customer.

Registration and Login:

1. Guest Registration/Login:
    - Guest: Chooses to register or log in to access additional features.
    - Sequence:
        1. Guest decides to register or log in.
        2. System prompts for registration or login details.
        3. Guest enters the required information.
        4. User Management System verifies credentials and grants access.

Managing Reports:

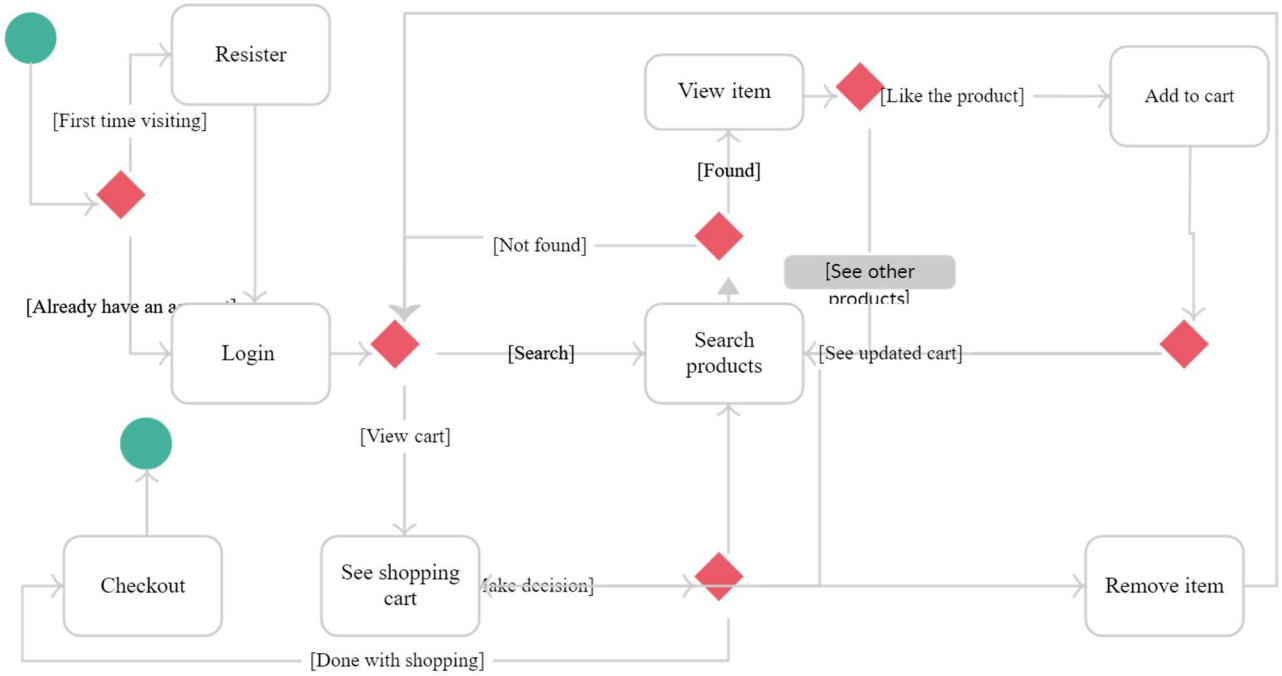
1. Updating or Deleting Reports:
    - Customer: Updates or deletes their own reports if there are changes.
    - Sequence:
        1. Customer requests to update or delete a report.
        2. Report Management System processes the request.
        3. Updated information is reflected in the Item Database.
2. Admin Review:
    - Administrator: Reviews and manages reports for accuracy.
    - Sequence:
        1. Admin accesses the admin panel.
        2. Admin reviews pending reports.
        3. Admin updates or deletes reports as necessary.

Notification Process:

1. Sending Notifications:
    - Notification System: Sends alerts to users about matching items or updates.
    - Sequence:
        1. System triggers a notification event based on user actions or item matches.
        2. Notification System sends the alert to the relevant users.

This sequence of interactions outlines how the Lost and Found App operates, detailing the flow from user actions to system responses and ensuring effective management of lost and found items.

5.3 State Diagram:



Customer States:

1. **Guest:** Initial state when a user visits the app without logging in.
2. **Logged In:** Customer has successfully logged into their account.
3. **Searching:** Customer is actively searching for lost or found items.
4. **Reporting:** Customer is in the process of reporting a lost or found item.
5. **Viewing Item:** Customer is viewing the details of a specific item.
6. **Communicating:** Customer is communicating with another user about a lost or found item.

Administrator States:

1. **Logged In:** Administrator is authenticated and logged into the admin panel.
2. **Managing Reports:** Administrator is reviewing, updating, or deleting item reports.
3. **Managing Users:** Administrator is managing user accounts and permissions.
4. **Logged Out:** Administrator logs out of the admin panel.

Item States:

1. **Reported:** Item has been reported as lost or found.
2. **Under Review:** Item report is under review by the administrator.
3. **Verified:** Item report has been verified by the administrator.
4. **Resolved:** Item has been returned to its owner or the report has been resolved.

Report States:

1. **New:** Report is newly created and awaiting review.

2. **Processing:** Report is being processed by the system and/or administrator.
3. **Matched:** Report has been matched with a similar item report.
4. **Closed:** Report has been resolved and closed.

State Diagram Layout:

States:

- Represented as rounded rectangles with meaningful names (e.g., "Logged In", "Searching", "Reported").
- Initial State: Indicated by a filled circle at the starting state.

Transitions:

- Represented by arrows with labels indicating the event triggering the transition (e.g., "Log In", "Search Items", "Report Item").

6.Estimation and Schedule

6.1. Cost and Effort Estimation

Phase 1: Planning and Research

- **Market Research and Analysis**
  - **Duration:** 2 weeks
  - **Cost:** ₹20,000 (Research Analyst)
- **Define Features and Requirements**
  - **Duration:** 1 week
  - **Cost:** ₹10,500 (Product Manager, Stakeholder Meetings)

Phase 2: Design

- **Wireframing and Mockups**
  - **Duration:** 2 weeks
  - **Cost:** ₹30,000 (UI/UX Designer)
- **User Interface Design**
  - **Duration:** 3 weeks
  - **Cost:** ₹40,500 (UI/UX Designer)

Phase 3: Development

- **Backend Development**
  - **Duration:** 6 weeks
  - **Cost:** ₹120,000 (Backend Developer)
- **Frontend Development**
  - **Duration:** 6 weeks (concurrent with backend)
  - **Cost:** ₹100,000 (Frontend Developer)
- **Integration and API Development**
  - **Duration:** 3 weeks

    - **Cost:** ₹60,000 (Backend Developer)

Phase 4: Testing

- **Unit Testing**
    - **Duration:** 2 weeks (concurrent with development)
    - **Cost:** ₹20,500 (QA Engineer)
- **User Acceptance Testing**
    - **Duration:** 2 weeks
    - **Cost:** ₹30,000 (QA Engineer, Test Users)

Phase 5: Deployment

- **Beta Launch**
    - **Duration:** 1 week
    - **Cost:** ₹10,000 (DevOps Engineer)
- **Final Launch**
    - **Duration:** 1 week
    - **Cost:** ₹10,500 (DevOps Engineer)

Phase 6: Marketing and Promotion

- **Pre-Launch Marketing**
    - **Duration:** 2 weeks
    - **Cost:** ₹5,000 (Marketing Specialist)
- **Post-Launch Marketing**
    - **Duration:** 4 weeks
    - **Cost:** ₹10,000 (Marketing Specialist)

Phase 7: Maintenance and Support

- **Ongoing Maintenance**
    - **Duration:** 6 months
    - **Cost:** ₹10,000/month = ₹60,000 (Maintenance Team)
- **User Support**
    - **Duration:** 6 months
    - **Cost:** ₹5,000/month = ₹30,000 (Support Team)

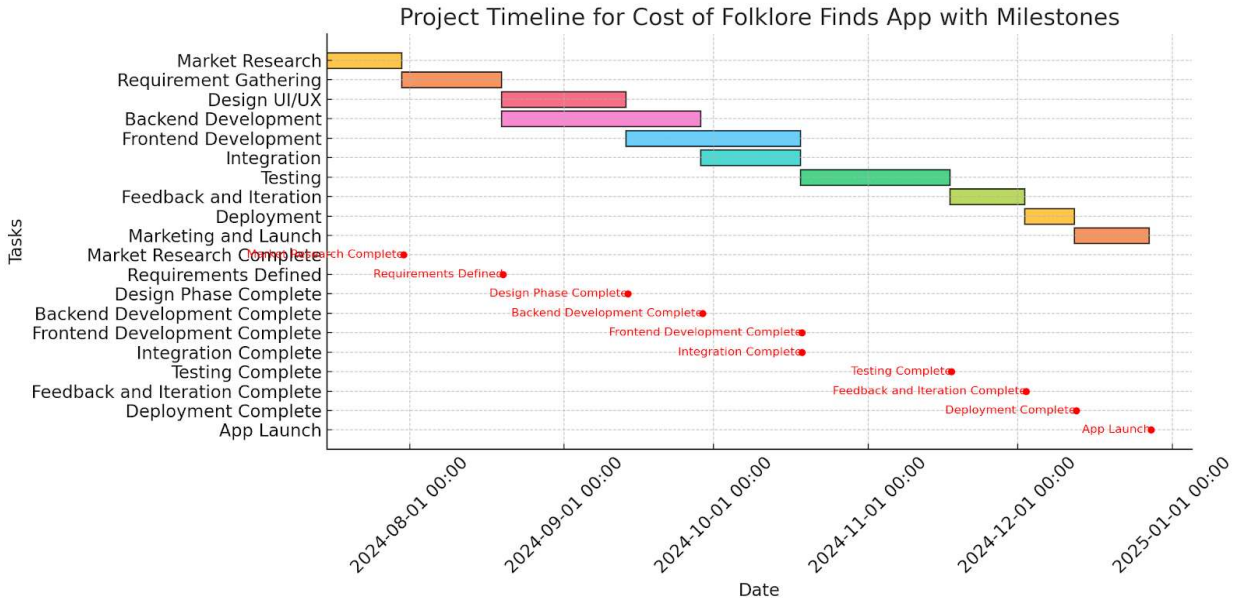
Summary Timeline and Costs

- **Total Duration:** Approximately 20 weeks (5 months) for initial launch, with 6 months of maintenance and support.
- **Total Estimated Cost:** ₹550,000 for development and initial launch, plus ₹90,000 for 6 months of maintenance and support.

6.2 Time-line Chart

Project Timeline with Milestones

1. Milestone 1: Market Research Complete
    - Market Research: 2024-07-15 09:00 AM to 2024-07-30 05:00 PM
2. Milestone 2: Requirements Defined
    - Requirement Gathering: 2024-07-30 09:00 AM to 2024-08-19 05:00 PM
3. Milestone 3: Design Phase Complete
    - Design UI/UX: 2024-08-19 09:00 AM to 2024-09-13 05:00 PM
4. Milestone 4: Backend Development Complete
    - Backend Development: 2024-08-19 09:00 AM to 2024-09-28 05:00 PM
5. Milestone 5: Frontend Development Complete
    - Frontend Development: 2024-09-13 09:00 AM to 2024-10-18 05:00 PM
6. Milestone 6: Integration Complete
    - Integration: 2024-09-28 09:00 AM to 2024-10-18 05:00 PM
7. Milestone 7: Testing Complete
    - Testing: 2024-10-18 09:00 AM to 2024-11-17 05:00 PM
8. Milestone 8: Feedback and Iteration Complete
    - Feedback and Iteration: 2024-11-17 09:00 AM to 2024-12-02 05:00 PM
9. Milestone 9: Deployment Complete
    - Deployment: 2024-12-02 09:00 AM to 2024-12-12 05:00 PM
10. Milestone 10: App Launch
    - Marketing and Launch: 2024-12-12 09:00 AM to 2024-12-27 05:00 PM



6.1 Project timeline for folklore finds

Project Milestones:

1. Market Research Complete: 2024-07-30 05:00 PM
2. Requirements Defined: 2024-08-19 05:00 PM
3. Design Phase Complete: 2024-09-13 05:00 PM
4. Backend Development Complete: 2024-09-28 05:00 PM
5. Frontend Development Complete: 2024-10-18 05:00 PM
6. Integration Complete: 2024-10-18 05:00 PM
7. Testing Complete: 2024-11-17 05:00 PM
8. Feedback and Iteration Complete: 2024-12-02 05:00 PM
9. Deployment Complete: 2024-12-12 05:00 PM
10. App Launch: 2024-12-27 05:00 PM

7. TEST CASES

The test cases outlined below for the Lost and Found App cover critical functionalities like user authentication, item reporting, and searching for items. Each test case ensures that users can seamlessly log in with valid credentials, report lost or found items, and search the database effectively.

7.1 Login Authentication

Test **ID:** #LA01
Test Scenario: User login authentication

Test Steps:

1. Open the login page on the Chrome browser on a device.
2. Enter a valid username in the "Username" field.
3. Enter a valid password in the "Password" field.
4. Click the "Login" button.

Precondition:

- User has a valid account with a known username and password.

Test Data:

- Valid username: `nithinkoushik.cs22@bmsce.ac.com`
- Valid password: `Password123`

Expected Result:

1. When a valid username and password are entered, the user is successfully logged in, and the user is redirected to the homepage.
2. When an invalid username or password is entered, an error message is displayed indicating incorrect username or password.

3. When the "Username" or "Password" field is left blank, an error message is displayed indicating that all fields are required.

Actual Results: As expected
Test Status: Pass/Fail: Pass

7.2 Report Lost Item

Test ID: #RI01
Test Scenario: User reports a lost item

Test Steps:

1. Log in to the app.
2. Navigate to the "Report Lost Item" page.
3. Fill in the item details (name, description, last seen location, date).
4. Click the "Submit" button.

Precondition:

- User is logged in.

Test Data:

- Item name: Wallet
- Description: Black leather wallet with multiple cards
- Last seen location: Central Park
- Date: 2023-07-15
- Finder’s USN:1BM22CS185
- Finder’s Name: Monish.P

Expected Result:

1. The lost item is successfully reported, and a confirmation message is displayed.
2. The reported item is added to the "Lost Items" database.

Actual Results: As expected
Test Status: Pass/Fail: Pass

7.3 Report Found Item

Test ID: #RF01
Test Scenario: User reports a found item

Test Steps:

1. Log in to the app.
2. Navigate to the "Report Found Item" page.
3. Fill in the item details (name, description, found location, date).
4. Click the "Submit" button.

Precondition:

- User is logged in.

Test Data:

- Item name: Watch
- Description: Silver wristwatch
- Found location: Library
- Date: 2023-07-18

Expected Result:

1. The found item is successfully reported, and a confirmation message is displayed.
2. The reported item is added to the "Found Items" database.

Actual Results: As expected
Test Status: Pass/Fail: Pass

7.4 Search Lost Items

Test ID: #SL01
Test Scenario: User searches for a lost item

Test Steps:

1. Log in to the app.
2. Navigate to the "Search Items" page.
3. Enter search criteria (item name, description).
4. Click the "Search" button.

Precondition:

- User is logged in.

Test Data:

- Search criteria: `Wallet`

Expected Result:

1. A list of lost items matching the search criteria is displayed.
2. User can view details of the items listed.

Actual Results: As expected
Test Status: Pass/Fail: Pass

7.5 Search Found Items

Test ID: #SF01
Test Scenario: User searches for a found item

Test Steps:

1. Log in to the app.
2. Navigate to the "Search Items" page.
3. Enter search criteria (item name, description).
4. Click the "Search" button.

Precondition:

- User is logged in.

Test Data:

- Search criteria: `Watch`

Expected Result:

1. A list of found items matching the search criteria is displayed.
2. User can view details of the items listed.

Actual Results: As expected
Test Status: Pass/Fail: Pass

7.6 Add Lost Item

Test ID: #ALI01
Test Scenario: Administrator adds a lost item

Test Steps:

1. Log in to the app as an administrator.
2. Navigate to the "Admin Panel".
3. Go to the "Add Lost Item" section.
4. Fill in the item details (name, description, last seen location, date).
5. Click the "Submit" button.

Precondition:

- Administrator is logged in.

Test Data:

- Item name: `Smartphone`
- Description: `Black iPhone with a cracked screen`
- Last seen location: `Bus Station`
- Date: `2023-07-20`

Expected Result:

1. The lost item is successfully added to the database, and a confirmation message is displayed.
2. The new lost item appears in the "Lost Items" list visible to all users.

Actual Results: As expected
Test Status: Pass/Fail: Pass

8. Conclusion

The development and launch of the Lost and Found App involve a comprehensive process encompassing multiple phases, from planning and design to development, testing, and deployment. The projected timeline for the initial launch is approximately 20 weeks (5 months), with an additional 6 months allocated for maintenance and support.

By investing in thorough user research, robust design, meticulous development, and strategic marketing, the Lost and Found App can establish a reliable platform for users to report, search for, and recover lost items. The structured timeline and budget allocation ensure that the project is completed efficiently and effectively, paving the way for a successful launch and sustained growth. This app aims to streamline the process of finding lost items and returning found items to their rightful owners, providing a valuable service to the community.