Software Requirements Specification

for

LOST AND FOUND INFORMATION SYSTEM

Version <5.0>

Prepared by

Group No. : 10

Rahul B Menon	**B210482CS**	**rahul_b210482cs@nitc.ac.in**
Pranav Prashant	**B210460CS**	**pranav_b210460cs@nitc.ac.in**
S Rishi Mohan	**B210552CS**	**rishi_b210552cs@nitc.ac.in**
Rahul P Aroli	**B210544CS**	**rahul_b210544cs@nitc.ac.in**
Ankesh Kumar	**B200821CS**	**ankesh_b200821s@nitc.ac.in**

INSTRUCTOR:	Prof. M Prabhu and Prof. Abdul Nazeer K A
COURSE:	DATABASE MANAGEMENT SYSTEM
LAB SECTION:	CS04
TEACHING ASSISTANT:	
DATE:	21/10/2023

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Ankesh Kumar	INTRODUCTION	10/10/2023
2.0	Rahul P Aroli	SOFTWARE REQUIREMENTS AND LOGS	12/10/2023
3.0	Rahul B Menon	FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	14/10/2023
4.0	S Rishi Mohan	USE CASE MODEL	16/10/2023
5.0	Pranav Prashant	PRODUCT DESCRIPTION AND INDEXING	20/10/2023

1.Introduction

1.1 Document Purpose

To outline the purpose and objectives of the Lost and Found Information System. The LFIS is designed to address the needs of organizations, institutions, or entities that manage lost and found items, ensuring efficient tracking, reporting, and recovery processes. This document provides a comprehensive understanding of the system's goals and the benefits it offers and also helps to understand the core functionalities of Database Management System used in this concept.

1.2 Product Scope

The objective of this project is to create a functional and efficient system for managing lost and found items within the educational institution. The LFIS will serve as a tool for students, staff, and administrators to report lost items, search for found items, and facilitate the recovery process.

1.3 Intended Audience and Document Overview

The intended audience for the Lost and Found Information System are:

- **Students:** Students are the primary users of the LFIS, both as individuals who may lose or find items and as contributors to the system by reporting lost or found items.
- **Faculty and Staff:** Faculty members and college staff may also use the system to report lost or found items, making them key users.
- **Administrators:** Administrative personnel responsible for managing the LFIS, including system administrators and database administrators.

1.4 Definitions, Acronyms and Abbreviations

- **DBMS:** Database Management System
  It is a software application or system that facilitates the creation, management, and manipulation of databases
- **LFIS**: Lost and Found Information System. A software system designed to manage and track lost and found items within the college, providing a platform for reporting, cataloging, and recovering lost items.
- **SRS**: Software Requirements Specification. A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform.

- **ReactJS**: A frontend development framework written in Javascript developed by Facebook.

- **MySQL**: An open-source relational database management system

1.5 Document Conventions

The document is written following the IEEE standard of preparing a software requirement specification.

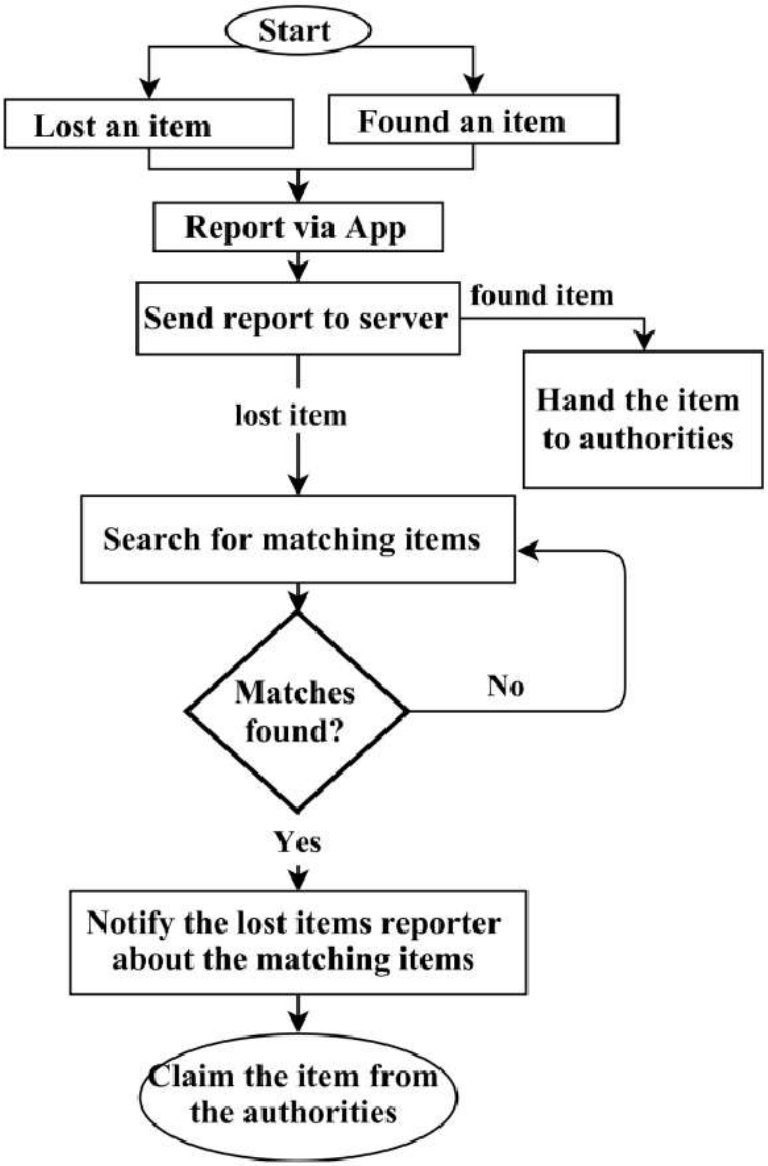
1.6 References and Acknowledgments

- R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7/e, Pearson Education, 2016
- R. Ramakrishnan and J. Gehrke, Database Management Systems,3/e, McGraw Hill, 2003
- P. Rob and C. Coronel, Database SystemsDesign, Implementation and Management, 7/e, Cengage Learning, 2007
- Wikipedia
- Stack Exchange
- W3Schools

2. Overall Description

2.1 Product Overview

Lost and Found Information System (LFIS) is a cutting-edge web application designed to address the complex needs of organizations, institutions, or entities that manages lost and found items.This innovative system is built using React for the frontend and SQL for the backend, ensuring a comprehensive and efficient solution for tracking, reporting, and recovering lost items.



2.2 Product Functionality

Efficient Item Tracking: LFIS provides a user-friendly and intuitive platform for tracking lost and found items. Through a streamlined interface, administrators can easily record found items, update their status, and categorize them, all in real-time.

Seamless Recovery Process: LFIS simplifies the process of returning lost items to their owners. It offers a search feature that allows users to quickly locate their misplaced belongings by entering a description, category, or other identifying information.

User-Friendly Interface: The React-based frontend ensures that LFIS is accessible to users of all technical backgrounds. The responsive design and intuitive navigation make it easy for staff and students to use, promoting widespread adoption.

Robust Database Management: The system is built on a SQL database, providing a solid foundation for data storage and retrieval. It ensures data integrity and security, making it a reliable solution for organizations with critical data needs.

Customization and Scalability: LFIS is highly adaptable and can be customized to meet the specific requirements of any organization. Whether you're a small college or a large corporation, LFIS can scale to accommodate your needs.

Cost-Effective Solution: Implementing LFIS is a cost-effective alternative to developing a similar system from scratch. It saves organizations time and resources, making efficient lost and found management accessible even for institutions with limited budgets.

2.3 Design and Implementation Constraints

Budget Constraints: Limited financial resources may constrain the choice of technologies, development time, and available features.

Technology Stack: The choice of React for the frontend and SQL for the backend, as specified, may limit the use of other technologies.

Scalability: The system must be designed to handle growth and accommodate increasing numbers of lost and found items. Scalability constraints can affect architecture and data storage choices.

Mobile Responsiveness: The application may need to be responsive for various devices, imposing constraints on the user interface and user experience.

Browser Compatibility: Compatibility with different web browsers (e.g., Chrome, Firefox, Safari) may be required, which can impact frontend design and testing.

Network Connectivity: Constraints on network reliability and speed can influence the application's ability to function offline or in low-bandwidth conditions.

2.4 Assumptions and Dependencies

- *Windows 7 and above*
- *Android*
- *Decent Internet Connectivity*
- *Member of the NITC Community*

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

User Registration and Login:

- **Purpose:** Allows new users to create accounts and existing users to log in.
- **Components:** Input fields for username, email, password, and a "Sign Up" button for registration. For login, username/email and password fields with a "Log In" button.

Lost Item Reporting:

- **Purpose:** Allows users to report lost items.
- **Components:** Forms for item details (description, date, location, category), option to upload images, and a "Submit" button.

Found Item Reporting:

- **Purpose:** Allows users to report found items.
- **Components:** Similar to the lost item reporting interface with fields for item details, images, and a "Submit" button.

Search and Browse:

- **Purpose:** Permits users to search the item list for lost and found items.
- **Components:** Search bar, filters (e.g., category, location, date), and a list/grid view of search results with item summaries.

Claiming Process:

- **Purpose:** Guides users through the process of claiming a found item.
- **Components:** A form to send a message to the finder/owner and a "Claim Item" button. It may also include a meeting scheduler.

Administrator Dashboard:

- **Purpose:** Offers administrators tools for managing the system and moderating content.
- **Components:** User management, content moderation, reporting, analytics, and system settings.

3.1.2 Hardware Interfaces

- *Your local machine (desktop, laptop) or mobile devices (browser application)*
- *A stable internet connection.*

3.1.2 Software Interfaces

- Browser renders the application in your desktop/mobile devices. Android application interface can also be used to access the database managements system

- The LOST AND FOUND INFORMATION SYSTEM is rendered as a web application. Hence, only a web browser is required for its functioning

3.2 Functional Requirements

3.2.1 USER AUTHENTICATION: Users must have the capability to log in securely and verify their identities in order to gain access to the system

3.2.2 USER REGISTRATION: The system should provide a mechanism for new users to register their profiles by furnishing the requisite information.

3.2.3 ITEM LISTING: The system should display lost and found items with relevant information.

3.2.4 REPORTING FOUND ITEMS: Users should be able to report found items in the item listing by providing a detailed description of the item, specifying the location where it was found and indicating the date of discovery.

3.2.5 SEARCH AND REQUEST FUNCTIONALITY: Users should be able to search for lost and found items based on various criteria. In the event that the individual who lost the item locates the item within the item listing, they should have the option to submit a formal request for its retrieval.

3.2.6 REPORTING LOST ITEMS: In the event that an individual who has lost an item cannot locate the item within the lost and found listing, an option should be provided to formally submit the item as lost, including comprehensive details about it.

3.2.7 RESPONDING TO LOST ITEMS: Should an individual come across an item that has been in the lost items registry, they should have the option to officially designate the item as found, and provide details of its discovery.

3.2.8 ITEM STATUS UPDATES: Users and administrators should be able to update the status of items.

3.2.9 ITEM EXPIRY AND DELETION: Lost and found item listings should have a time limit. All the items found and received by the owner should be deleted from the listing.

3.2.10 MESSAGING AND COMMUNICATION: Users must be able to communicate with other users or administrators for item recovery.

3.3 Use Case Model

A use case model for a Lost and Found system is a structured representation that identifies and describes the key interactions and functionalities required to support the seamless management of reported lost items, their storage, search, and the claiming process. To illustrate these crucial aspects, a simplified use case model for a Lost and Found system is shown below :

3.3.1 USER IDENTITY VERIFICATION AND ACCESS CONTROL :

AUTHOR :-

- Pranav Prashant

PURPOSE :-

- Access Control
- User Identity Verification
- User Account Management
- User Convenience

PRIORITY : -

- User Registration (Priority: High)
- User Login (Priority: High)
- Authenticate User (Priority: High)

ACTORS : -

- User
- System
- Administrator (Optional)

FLOW OF EVENTS : -

- Use Case: User Login
- User Requests Password Reset

3.3.2 LISTING ITEMS:

AUTHOR :-

- Pranav Prashant

PURPOSE :-

- Record Lost and Found Items
- Recovery of Lost Items
- Documentation and Accountability
- User Engagement
- Search and Retrieval

PRIORITY : -

- Report Lost and Found Items (Priority: High)
- Search for Lost and Found Items (Priority: Medium)
- View Lost and Found Items (Priority: Low)
- Manage Lost and Found Items (Priority: Low)

ACTORS : -

- User
- Administrator
- System
- The Lost and Found system itself is also considered an actor

FLOW OF EVENTS : -

- Report Lost Item

3.3.3 REPORTING LOST/FOUND ITEMS:

AUTHOR :-

- Rahul B Menon

PURPOSE :-

- Reuniting Owners with Belongings
- Documenting Found Items
- User Empowerment
- Security and Transparency
- Efficient Item Tracking

PRIORITY : -

- Report Lost Item (Priority: High)

- Report Found Item (Priority: High)
- Automatic Matching and Notifications (Priority: Medium)
- User-Friendly Interface (Priority: High)
- View Reported Items (Priority: Medium)
- User Notifications (Priority: Medium)

ACTORS : -

Primary Actors:

- User
- Administrator
- System

FLOW OF EVENTS : -

- Use Case: Report Lost Item
- Use Case: Report Found Item

3.3.4 SEARCH AND REQUEST FUNCTIONALITY:

AUTHOR :-

- Rahul B Menon

PURPOSE :-

- Item Retrieval
- Efficient Search
- Match Notification
- Request Item Return
- Administrative Oversight

PRIORITY : -

- Search for Lost Items (Priority: High)
- Search for Found Items (Priority: High)
- Automatic Matching and Notifications (Priority: Medium)
- Request Item Return (Priority: High)
- Manage Item Return Requests (Priority: Medium)
- User Notifications (Priority: Medium)

ACTORS : -

Primary Actors:

- User
- Administrator
- System

FLOW OF EVENTS : -

- Use Case: Search for Lost Items
- Use Case: Search for Found Items
- Use Case: Request Item Return
- Use Case: Manage Item Return Requests (Admin)

3.3.5 HANDLING LOST ITEMS:

AUTHOR :-

- S Rishi Mohan

PURPOSE :-

- Item Retrieval
- User Interaction
- Finder Interaction
- Notification and Communication
- Item Verification

PRIORITY : -

- Respond to Lost Items (Priority: High)
- Automated Notifications (Priority: Medium)
- Communication with Finders (Priority: High)
- Item Verification and Validation (Priority: Medium)
- Recording and Tracking (Priority: Medium)

ACTORS : -

Primary Actors:

- User
- Administrator
- System

FLOW OF EVENTS : -

- Use Case: Respond to Lost Items
- Use Case: Initiate Item Return Process
- Use Case: Communication (User and Finder)
- Use Case: Record Communication

3.3.6 STATUS OF ITEMS:

AUTHOR :-

- S Rishi Mohan

PURPOSE :-

- Visibility and Transparency
- Status Notification
- User Empowerment
- Efficient Communication
- Record Keeping

PRIORITY : -

- Item Status Updates Notification (Priority: High)
- User-Initiated Communication (Priority: Medium)
- Status Updates Record Keeping (Priority: Medium)

ACTORS : -

Primary Actors:

- User
- Finder
- System

FLOW OF EVENTS : -

- Use Case: Item Status Updates Notification
- Use Case: User-Initiated Communication
- Use Case: Record Communication

3.3.7 EXPIRY AND DELETION OF ITEMS:

AUTHOR :-

- Rahul P Aroli

PURPOSE :-

- Maintain System Integrity
- Efficient Resource Management
- User Experience
- Compliance and Data Security
- System Performance

PRIORITY : -

- Item Expiry and Deletion (Priority: High)
- Archiving Expired Items (Priority: Medium)
- Notification to Users (Priority: Medium)
- Secure Data Deletion (Priority: High)
- Grace Period Handling (Priority: Low)

ACTORS : -

Primary Actors:

- User
- Administrator
- System

FLOW OF EVENTS : -

- Use Case: Item Expiry and Deletion
- Use Case: Notification to Users
- Use Case: User Action
- Use Case: Grace Period Handling

3.3.8 MESSAGING AND COMMUNICATION:

AUTHOR :-

- Ankesh Kumar

PURPOSE :-

- Facilitate User Interaction
- Item Reunification
- Resolving Item Status
- User Support
- Security and Privacy

PRIORITY : -

- Messaging and Communication (Priority: High)
- Item Reunification (Priority: Medium)
- Resolving Item Status (Priority: Medium)
- User Support (Priority: Low)
- Security and Privacy (Priority: High)

ACTORS : -

Primary Actors:

- User
- Administrator
- System

FLOW OF EVENTS : -

- Use Case: Messaging and Communication

USE CASE DIAGRAMS:-

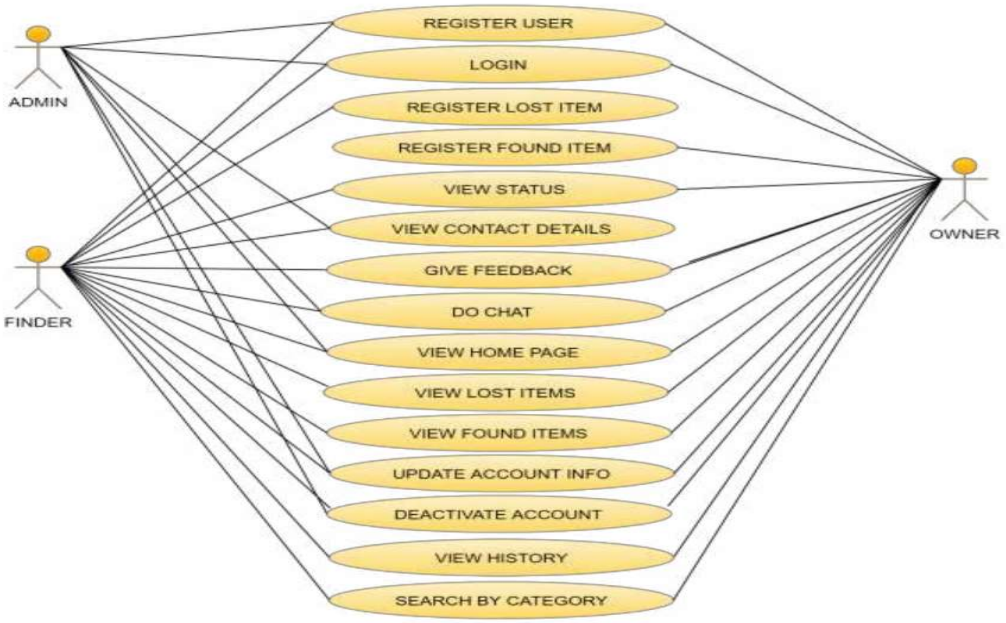


Fig 1 : Use Case Diagram for whole Project

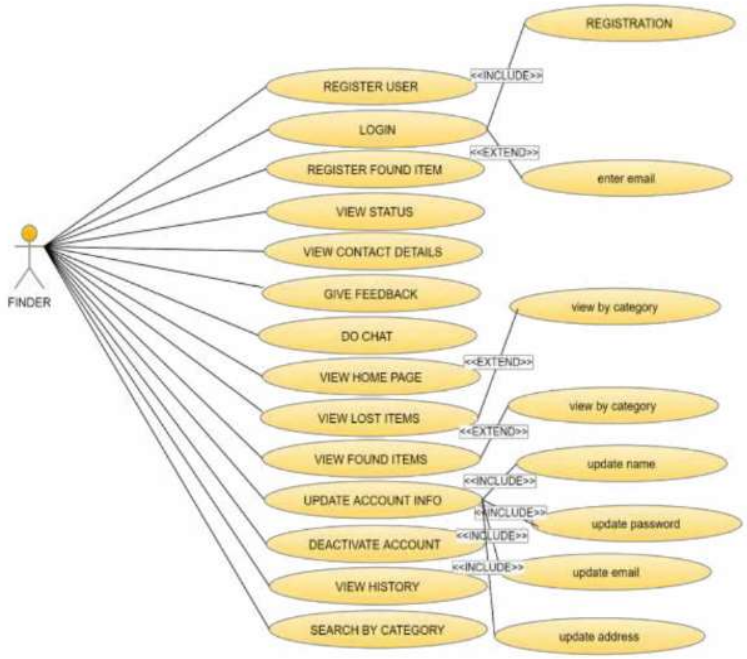


Fig 2 : Use Case Diagram for Finder

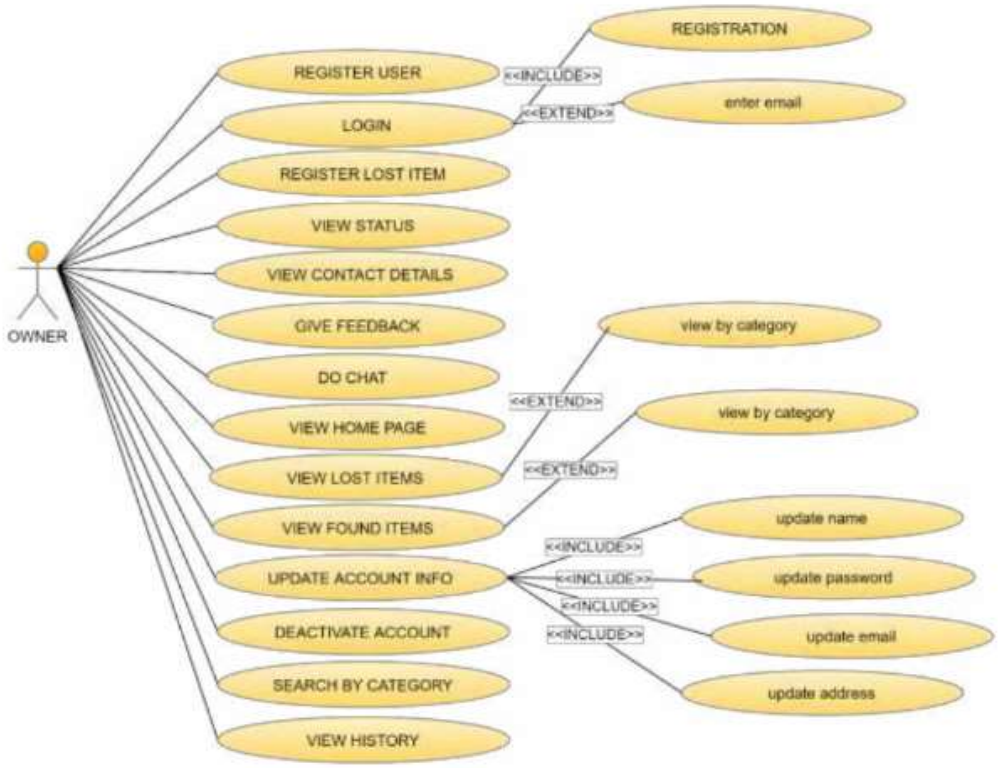


Fig 3 : Use Case Diagram for Owner

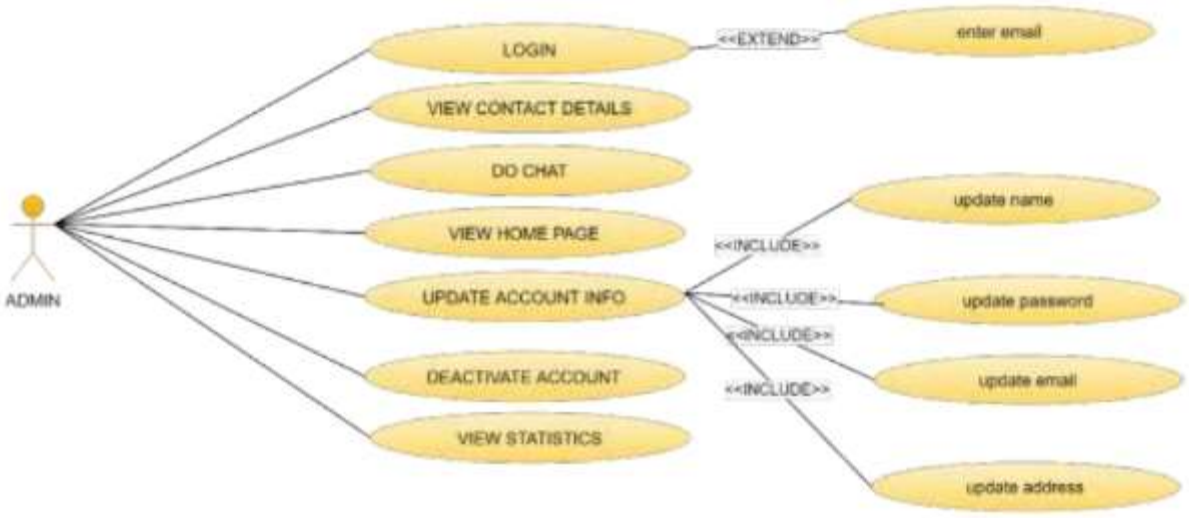


Fig 4 : Use Case Diagram for Admin

4. Other Non-Functional Requirements

4.1 Performance Requirements

The LOST AND FOUND INFORMATION SYSTEM should be able to handle a few hundred transactions every day. The system should have low response time, high search speed and should be able to handle concurrent users. The system should be scalable and should be able to accommodate an increase in users and data volume over time. Efficient data archiving and cleanup procedures have to be implemented to maintain optimal system performance over time.

4.2 Safety and Security Requirements

The LFIS should have the following safety and security requirements:

- **ADMINISTRATOR FUNCTIONS:** Administrators should have the necessary tools and functionalities for the purpose of overseeing user accounts and item listings and for facilitating the resolution of disputes.

- **DATA RETENTION:** Even subsequent to the removal of an item from the item listing following its return to the owner, all the information pertaining to the item, along with the details of the individual who retrieved it, should be retained within the system for a specified duration.

- **DISPUTE RESOLUTION:** Dispute resolution procedures must be implemented.

4.3 Software Quality Attributes

There are many quality attributes of a database, but some of the most important ones are:

- **Correctness:** The data in the database must be accurate and consistent. The software that we intend to develop would be consistent with respect to the data collected and would be updated regularly to correct any sort of anomaly in the data collected.

- **Completeness:** The database should contain all the data that is relevant to the task at hand. The user would be required to enter all the relevant details that are stored in the database through forms.

- **Usability:** The database should be easy to use and understand. The dynamic frontend as well as the application server interface would provide necessary

features that enhance the ease of use of the application and improve the understanding about the database.

- **Performance:** The database should be able to handle large amounts of data and be able to process queries quickly. A distributed database system with clusters in different locations would enhance the accessibility and improve the performance of the given application

- **Scalability:** The database should be able to grow and change as the needs of the users change. Scalability is partly the developers job as the developer is responsible for adding functionality to the database that makes the database more scalable.

5. Other Requirements

- **USER FEEDBACK:** A feedback option should be provided wherein users can provide their feedback and suggestions regarding the functioning of the system.

- **REPORTING AND ANALYTICS:** The system should generate reports on the status and activity of lost and found items. Analytic tools can help administrators identify trends and improve the system.

Appendix A – Data Dictionary

Table	Field	DataType	Description
Items	item_ id	INT	Primary Key
Items	name	VARCHAR(255)	Name of the Item
Items	description	TEXT	Description of the item
Items	found_date	DATE	Date the item was found
Items	found_location	VARCHAR(255)	Location where the item was found
Items	status	VARCHAR(255)	Status of the item (e.g., lost, found, claimed, disposed of)
Claims	claim_id	INT	Primary key
Claims	item_id	INT	Foreign key to the Items table
Claims	claimant_name	VARCHAR(255)	Name of the person claiming the item
Claims	claimant_contact_information	VARCHAR(255)	Contact information for the person claiming the item
Claims	claim_date	DATE	Date the claim was made
Claims	claim_status	VARCHAR(255)	Status of the claim (e.g., pending, approved, denied)

Appendix B - Group Log

Primary Author(s)	Description of work done	Date Completed
Ankesh Kumar	Introduction	10/10/2023
Rahul P Aroli	Software Requirements & Logs	12/10/2023
Rahul B Menon	Functional & Non-Functional Requirements	14/10/2023
S Rishi Mohan	Use Case Model	16/10/2023
Pranav Prashant	Product Description & Indexing	20/10/2023