PUCIT
Punjab University College of Information Technology

LOST2FIND App

Group Members:

BSEF20A020	Noor Fatima
BSEF20A049	Laiba Liaquat

Supervised By:

Sir Muhammad Nadeem

Second Deliverable for Object Oriented Approach

Version 1.0

TABLE OF CONTENTS

1 Introduction

Second deliverable is all about the usecase modeling and software design. In the previous deliverable, analysis of the system is completed. So we understand the current situation of the problem domain. Now we are ready to strive for a solution for the problem domain by using object-oriented approach.

Following artifacts must be included in this deliverable.

1. Key Features
2. Use case description
3. Use case diagram
4. Collaboration Diagram
5. Design Class Diagram
6. ERD Diagram
7. Sequence Diagram
8. Operation Contracts
9. DFD

Now we discuss these artifacts one by one as follows:

1.1 Key Features:

Following are the key features of our project:

- **User-Friendly Reporting:**

  The app offers an intuitive and user-friendly interface for reporting lost items. Users can quickly and efficiently provide detailed information about their lost possessions, enhancing the chances of a successful recovery**.**

- **Smart Search Functionality:**

  Utilizing advanced search algorithms, the app enables users to search for lost items based on various criteria such as category, location, and description. This intelligent search functionality maximizes the likelihood of reuniting individuals with their belongings.

- **Efficient Communication Channels:**

  The app serves as a communication hub, allowing users to interact with individuals who have found their lost items. Through secure and convenient messaging features, users can coordinate the safe return of their possessions.

- **Finder Reporting System:**
  Finders play a crucial role in the Lost2Find community. The app provides a dedicated interface for finders to report items they have found. This streamlined process ensures that valuable possessions are promptly registered in the system.

- **Real-Time Notifications:**
  To keep users informed and engaged, the app incorporates a robust notification system. Users receive real-time alerts about potential matches for their lost items, messages from finders, and updates on the status of their reports.

- **Secure and Privacy-Focused:**
  Security and privacy are paramount. The Lost2Find app employs cutting-edge encryption and authentication measures to safeguard user data, fostering a trustworthy and secure environment for interactions.

- **Administrative Oversight:**
  The app features administrative tools to monitor reported items, resolve disputes, and enforce community guidelines. This ensures a positive and responsible user experience within the Lost2Find community.

- **Continuous Improvement:**
  Committed to providing an exceptional user experience, the Lost2 Find app embraces continuous improvement. User feedback is valued, and the app evolves with updates and enhancements to meet the dynamic needs of its growing user base.

1.2 Use Case Description:

A use case diagram provides a high-level view of the functionality and interactions of a system from the perspective of its users. Below is a detailed description of a use case diagram for a Lost2Find app:

Description of Use Case Diagram for Lost2Find App:

Actors:

1. User:

- Description:

Represents individuals who use the Lost and Find app to report lost items, search for lost items, and communicate with others to return found items.
Roles and Responsibilities:
- Report Lost Item
- Search for Lost Item
- Communicate with Finder
- Receive Notifications

Offer a reward

2. Finder/ valuable user:
Description:
Represents individuals who use the app to report found items, search for owners of found items, and communicate with those who have lost items.
Roles and Responsibilities:
- Report Found Item
- Search for Item Owners
- Communicate with Owner
- Return Found Items

3. Admin/Moderator:
Description:
Represents administrators or moderators who oversee and manage the Lost and Found app, ensuring adherence to guidelines, resolving disputes, and maintaining the app's integrity.
Roles and Responsibilities:
- Review Reported Items

Manages User
- Resolve Disputes
- Enforce Guidelines
- Monitor App Activities

Use Cases:

1. **User Registration:**

Description: Allows a new user to create an account on the lost to find app.
Actors: User
Preconditions: The user is not registered.

Post conditions: The user is registered and has a unique account
Flow of Events:
User selects the "Register" option.
User provides required registration information (e.g., username, email, and password).
System validates and creates a new user account.
User receives a confirmation message.

2. **User Login:**

Description: Allows a registered user to log in to their account.
Actors: User
Preconditions: The user is registered.
Post conditions: The user is logged into their account.
Flow of Events:
User selects the "Login" option.
User enters their username and password.
System verifies the credentials.
If credentials are valid, the user is logged in; otherwise, an error message is displayed.

3. **Report Lost Item:**

Actors: User
Description: Users can report details about an item they have lost, including a description, category, and location.

4. **Search for Lost Item:**

Actors: User
Description: Users can search for lost items based on various criteria, such as category, location, or description.

5. **View Found Items:**

Description: This use case enables a user to view a list of found items.
Actors: User
Flow of Events:

- The logged-in user navigates to the "Found Items" section within the app.
- The app retrieves a list of found items from the database.
- The user can filter or search for specific found items based on criteria such as location or category.

- The app displays a list of found items with relevant details, such as descriptions and images.
- The user can click on a specific item to view more details and contact the finder if needed

5. Contact with Finder:

Actors: User

Description: Users can communicate with individuals who have found their lost items to coordinate the return process.

6. Offer a Reward for Found Item:

Description: Allows a user to offer a reward for a found item they want to retrieve.

Actors: User

Preconditions: The user is logged into their account, and they have found an item.

Post conditions: The reward offer is associated with the found item.

Flow of Events:

User selects the "Offer Reward" option.

User provides details about the found item (e.g., description, location).

User specifies the reward amount they are willing to offer.

System associates the reward offer with the found item in the database.

User receives confirmation of the reward offer.

7. Receive Notifications:

Actors: User

Description: Users receive notifications when there is a potential match for their lost item or when someone communicates with them about a found item.

8. Report Found Item:

Actors: Finder/valuable user

Description: Finders can report details about items they have found, including a description, category, and location.

9. Search for Item Owners:

Actors: Finder

Description: Finders can search for owners of items they have found based on various criteria, such as category, location, or description.

10. Communicate with Owner:

Actors: Finder

Description: Finders can communicate with individuals who have lost items to coordinate the return process.

11. Return Found Items:

Actors: Finder

Description: Finders can arrange and carry out the process of returning found items to their owners.

12. Rating and give feedback:

Description: Enables users to rate and provide feedback on the Lost and Found App. This feedback can help improve the app's features and user experience.

Actor: Registered User

Trigger: User wishes to share their experience or suggest improvement

13.Admin Login:

Actor: Admin

Description: Allows administrators to log in to the system using their credentials, granting access to admin-specific functionalities.

14. Review Reported Items:

Actors: Admin/Moderator

Description: Admins or moderators can review reported items, ensuring they adhere to guidelines and taking appropriate actions.

15. Manage Users:

Actor: Admin

Description: Allows administrators to view, edit, or deactivate user accounts. This feature ensures user management and system security.

16. Resolve Disputes:

Actors: Admin/Moderator

Description: Admins or moderators can intervene and resolve disputes or conflicts between users.

17. Enforce Guidelines:

Actors: Admin/Moderator

Description: Admins or moderators can enforce community guidelines to maintain a safe and respectful environment within the app.

18. Monitor App Activities:

Actors: Admin/Moderator

Description: Admins or moderators can monitor overall app activities, ensuring compliance with rules and addressing any issues that may arise.

19.Logout:

Actor: User or Admin

Description: Allows users and administrators to log out of their accounts, ensuring the security of their information.

Relationships:

- Association Relationships:

- Users are associated with multiple use cases, including reporting lost items, searching for items, and communicating with finders.
- Finders are associated with use cases related to reporting found items, searching for owners, and communicating with owners.
- Admins/Moderators are associated with use cases related to reviewing reported items, resolving disputes, enforcing guidelines, and monitoring activities.

- Include Relationships:

- Communication with Finder and Communication with Owner include the Receive Notifications use case.

- Extend Relationships:

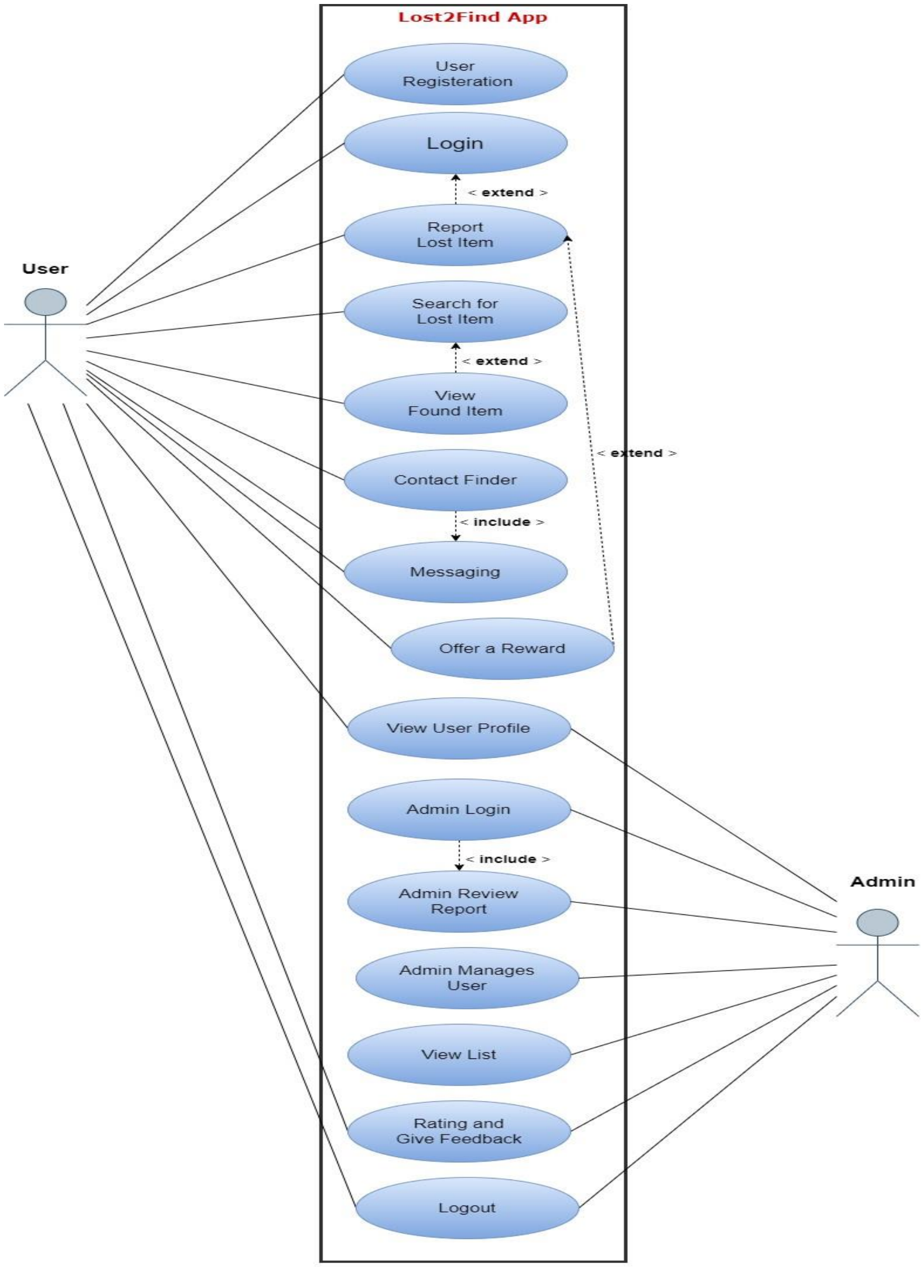
- Use case report lost to item extend to login Use case
- Use case View found item extends to search for lost items Use case
- Use case offer a reward extends to report to lost item Use case

Summary:

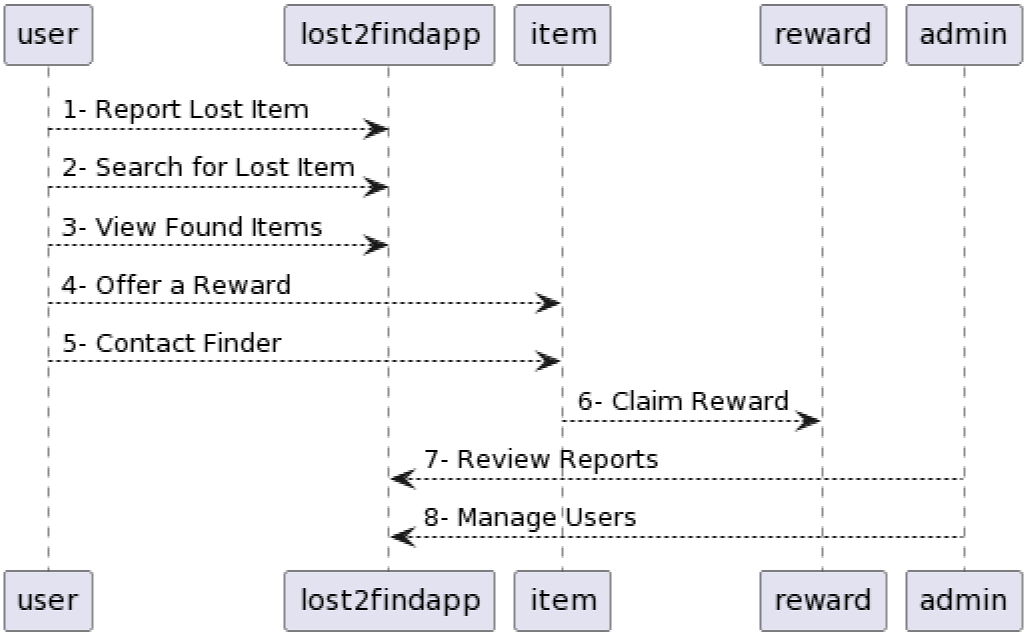
- The use case diagram provides an overview of the interactions between actors and use cases in the Lost and Found app.
- It highlights key functionalities such as reporting lost/found items, searching, communication, and administrative actions.

- This diagram serves as a foundation for more detailed use case descriptions and further system design.

1.3 Use Case Diagram:



1.4 Collaboration Diagram:



1.5 Collaboration Diagram Description:

The collaboration diagram for a lost and found app illustrates the dynamic interactions between different actors and objects within the system. Here's a detailed description of the interactions depicted in the collaboration diagram:

1. User Reports Lost Item:

Actor Involved: User

Interaction: The user interacts with the "Lost2FindApp" by reporting a lost item. This involves sending a message to the app, providing details about the lost item.

2. User Searches for Lost Item:

Actor Involved: User

Interaction: The user interacts with the app by searching for lost items. This involves sending a message to the app to retrieve information about lost items based on certain criteria.

3. User Views Found Items:

Actor Involved: User

Interaction: The user interacts with the app to view found items. This involves sending a message to the app to retrieve and display a list of found items.

4. User Offers a Reward:

Actors Involved: User, Item

Interaction: The user interacts with a specific "Item" by offering a reward. This involves sending a message to the item, updating its reward information.

5. User Contacts Finder:

Actors Involved: User, Item

Interaction: The user interacts with a specific "Item" by sending a message to contact the finder. This may involve messaging functionality within the app.

6. Finder Claims Reward:

Actors Involved: User, Item, Reward

Interaction: The user (finder) interacts with the "Item" to claim the reward. This involves sending a message to the item, and potentially interacting with the "Reward" associated with the item.

7. Admin Reviews Reports:

Actor Involved: Admin

Interaction: The admin interacts with the "Lost2FindApp" by reviewing reports of lost items. This involves sending a message to the app, accessing and reviewing information about reported lost items.

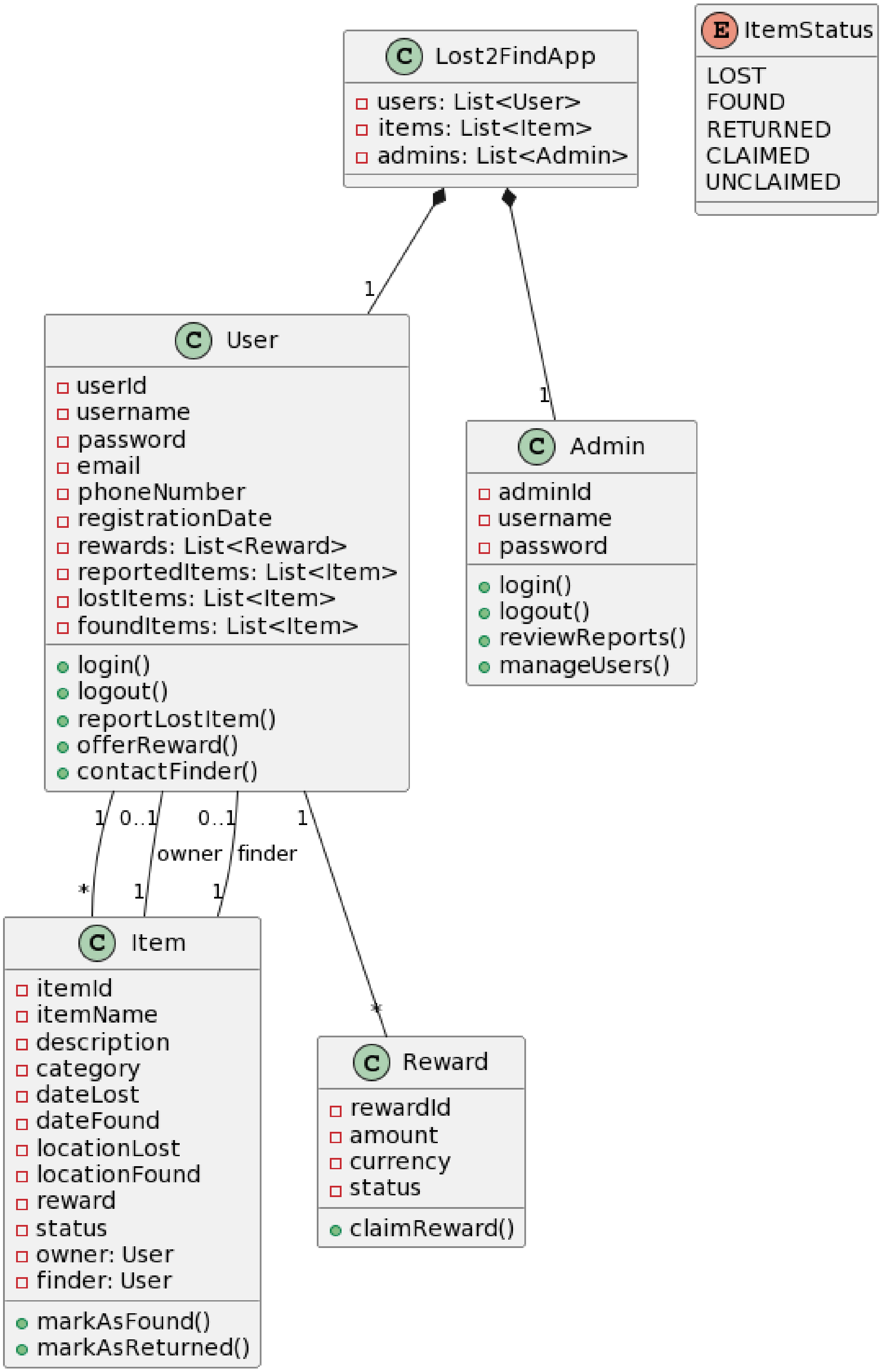
8. Admin Manages Users:

Actor Involved: Admin

Interaction: The admin interacts with the "Lost2FindApp" by managing user-related tasks. This involves sending a message to the app, performing actions such as user authentication, user account management, etc.

In summary, the collaboration diagram provides a visual representation of the dynamic interactions between users, administrators, and various components within the lost and found app. It highlights the flow of messages and actions during key scenarios, including reporting lost items, searching for items, offering rewards, contacting finders, and administrative tasks.

1.6 Class Diagram:



1.7 Class Diagram Description:

Class Descriptions:

Lost2FindApp: Represents the main application class containing lists of users and items.

User: Represents a user of the app with attributes such as userId, username, password, email, phoneNumber, and methods like report LostItem and viewFoundItems.

LostItem: Represents an item reported as lost with attributes like itemId, itemName, description, category, location, dateLost, reward, owner, finder, and status. It has a method updateStatus to change the status of the item.

FoundItem: Represents an item that has been found with attributes similar to LostItem and a method claimItem to handle the claiming process.

Status: Represents the status of an item, allowing for tracking its current state. It includes attributes statusId and statusName.

Associations:

The Lost2FindApp class has associations with the User and LostItem classes.

The User class has associations with the LostItem and FoundItem classes.

The LostItem and FoundItem classes have associations with the User and Status classes.

The LostItem and FoundItem classes have a bidirectional association with the Status class.

Lost2FindApp Class:

Attributes:

Users: List<User>: A collection of User objects representing individuals using the app.

Items: List<Item>: A collection of Item objects representing lost or found items.

Admins: List<Admin>: A collection of Admin objects representing administrators overseeing the app.

User Class:

Attributes:

- userId: Unique identifier for the user.
- username: User's username for authentication.
- password: User's password for authentication.
- email: User's email address.
- phoneNumber: User's contact phone number.
- registrationDate: Date when the user registered in the app.

- rewards: List<Reward>: A collection of Reward objects representing rewards associated with found items.
- reportedItems: List<Item>: A collection of Item objects representing items reported as lost by the user.
- lostItems: List<Item>: A collection of Item objects representing items owned by the user and reported as lost.
- foundItems: List<Item>: A collection of Item objects representing items found by the user.

Methods:

- login(): Authenticates the user into the app.
- logout(): Logs the user out of the app.
- reportLostItem(): Allows the user to report a lost item.
- offerReward(): Allows the user to offer a reward for a found item.
- ContactFinder (): Enables the user to contact the finder of a lost item.

Admin Class:

Attributes:

- adminId: Unique identifier for the admin.
- username: Admin's username for authentication.
- password: Admin's password for authentication.

Methods:

- login(): Authenticates the admin into the app.
- logout(): Logs the admin out of the app.
- reviewReports(): Allows the admin to review reports of lost items.
- manageUsers(): Enables the admin to manage user accounts.

Item Class:

Attributes:

- itemId: Unique identifier for the item.
- itemName: Name or description of the item.
- description: Additional details about the item.
- category: Category of the item (e.g., electronics, clothing).
- dateLost: Date when the item was reported as lost.
- dateFound: Date when the item was found.
- locationLost: Location where the item was reported as lost.
- locationFound: Location where the item was found.
- reward: Reward associated with the item.
- status: Current status of the item (e.g., lost, found).

- owner: User: User object representing the owner of the item.
- finder: User: User object representing the finder of the item.

Methods:

- markAsFound(): Changes the status of the item to "found."
- markAsReturned(): Marks the item as returned to the owner.

Reward Class:

Attributes:

- rewardId: Unique identifier for the reward.
- amount: Amount of the reward.
- currency: Currency of the reward (e.g., USD, EUR).
- status: Current status of the reward (e.g., pending, claimed).

Methods:

- claimReward(): Allows a user to claim a reward associated with a found item.

Associations:

Lost2FindApp to User and Admin:

- The Lost2FindApp class has associations with both User and Admin classes, indicating that the app contains collections of users and administrators.

User to Item and Reward:

- The User class is associated with the Item and Reward classes, representing that a user can own or find multiple items and can be associated with multiple rewards.

Item to User:

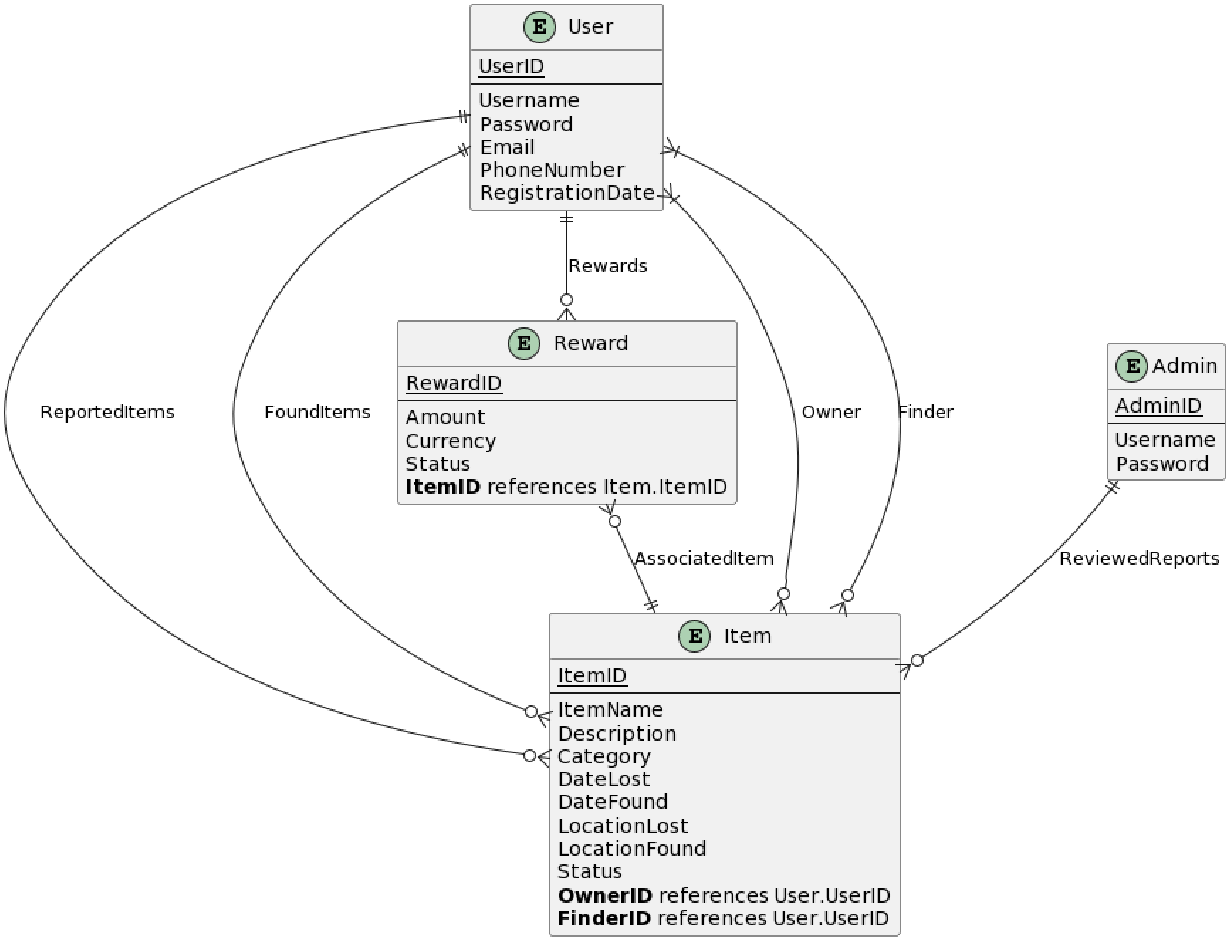
- The Item class has an association with the User class, indicating the owner of the item.

Item to Reward:

- The Item class also has an association with the Reward class, representing the reward associated with the item.

This class diagram provides a visual representation of the key classes, attributes, methods, and associations in a lost 2 find app. It demonstrates how users, administrators, items, and rewards are structured and interconnected within the application's system.This class diagram provides a foundation for modeling the core entities and relationships in a lost 2 find app.

1.8 ERD:



1.9 ERD Description:

Designing an Entity-Relationship Diagram (ERD) involves identifying the main entities, their attributes, and the relationships between them.

Entities and Attributes:

1. **User:**
 - Attributes: UserID (PK), Username, Password, Email, PhoneNumber, RegistrationDate

2. Admin:
 - Attributes: AdminID (PK), Username, Password

3. Item:
 - Attributes: ItemID (PK), ItemName, Description, Category, DateLost, DateFound, LocationLost, LocationFound, Reward, Status
 - Foreign Key: OwnerID (FK referencing UserID), FinderID (FK referencing UserID)

4. Reward:
 - Attributes: RewardID (PK), Amount, Currency, Status
 - Foreign Key: UserID (FK referencing UserID)

Relationships:

User-Item Relationship:
 - A User can report multiple lost items (One-to-Many).
 - An Item is reported by one User.

User-Reward Relationship:
 - A User can claim multiple rewards (One-to-Many).
 - A Reward is claimed by one User.

Item-User (Owner) Relationship:
 - An Item is owned by one User (Many-to-One).
 - A User can own multiple items.

Item-User (Finder) Relationship:
 - An Item can be found by one User (Many-to-One).

- A User can find multiple items.

Cardinality and Connectivity:
User-Item Relationship:

- User (1) --< Item (Many)
- Connectivity: Optional (a User may not necessarily report an item, and an item may not have an owner immediately)

User-Reward Relationship:

- User (1) --< Reward (Many)
- Connectivity: Optional (a User may or may not claim a reward)

Item-User (Owner) Relationship:

- Item (Many) >-- User (1)
- Connectivity: Mandatory (an item must have an owner, but a user may own zero or more items)

Item-User (Finder) Relationship:

- Item (Many) >-- User (1)
- Connectivity: Optional (an item may or may not have a finder, and a user may find zero or more items)

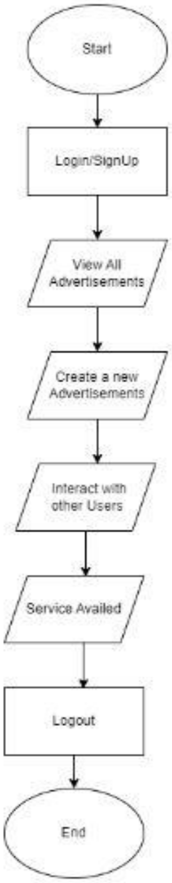
Key Points:

- Each entity has a primary key (PK) that uniquely identifies instances.
- Foreign keys (FK) establish relationships between entities.
- Relationships are annotated with cardinality (One/Many) and connectivity (Optional/Mandatory).

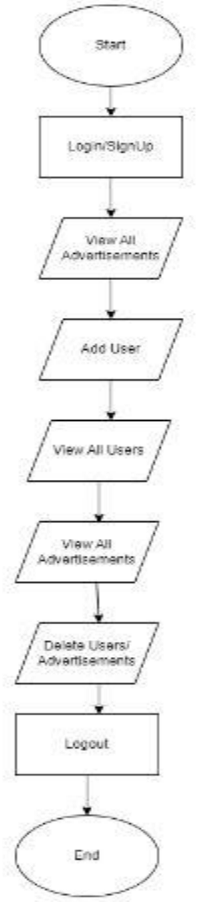
This ERD captures the essential entities, attributes, and relationships for a Lost 2 Find app. It represents the user-item interactions, the claiming of rewards, and the ownership and finding of items within the application. The cardinality and connectivity annotations provide additional context about the nature of these relationships. Below the simplest form of ERD Diagram for user and Admin side.

1.10 Sequence Diagram:

User Side:



Admin Side:



1.11 Sequence Diagram Description:

Sequence DIAGRAM Description:

A sequence diagram is a visual representation of the interactions between different components or objects in a system over time. Below is a description of a sequence diagram for a Lost and Found app, focusing on the interactions between key components during the process of reporting and searching for lost items:

Actors/Components:

1. **User:** Represents an individual using the Lost and Found app.

2. Finder: Represents an individual who found a lost item and is using the app.
3. System/Server: Represents the backend server handling the app's functionality.

Sequence of Interactions:

1. User Reports a Lost Item:
User Action: Initiates the process by selecting "Report Lost Item" in the app.
System Action: Receives the report request and stores the details of the lost item.
Communication: User communicates with the server.

2. System Sends Confirmation:
-System Action: Generates a confirmation message indicating the successful report of the lost item.
- Communication: Server sends a confirmation message to the user.

3. User Searches for Lost Items:
- User Action: Initiates a search by entering criteria such as category, location, or description.
- System Action: Processes the search request and retrieves relevant lost items.
- Communication: User communicates with the server.

4. System Sends Search Results:
System Action: Sends a list of search results to the user, including detailed information about each item.
Communication: Server sends the search results to the user.

5. User Communicates with Finder:
User Action: Selects a specific lost item and initiates communication with the finder.
System Action: Facilitates communication between the user and the finder.
Communication: User communicates with the server, which relays messages to the finder.

6. Finder Responds to User:

Finder Action: Receives the user's message and responds accordingly.

System Action: Relays the finder's response to the user.

Communication: Finder communicates with the server, and the server communicates with the user.

7. User Confirms Match:

User Action: If a potential match is confirmed, the user communicates their confirmation.

System Action: Processes the confirmation and initiates the process of arranging the return of the found item.

Communication: User communicates with the server.

8. System Coordinates Return:

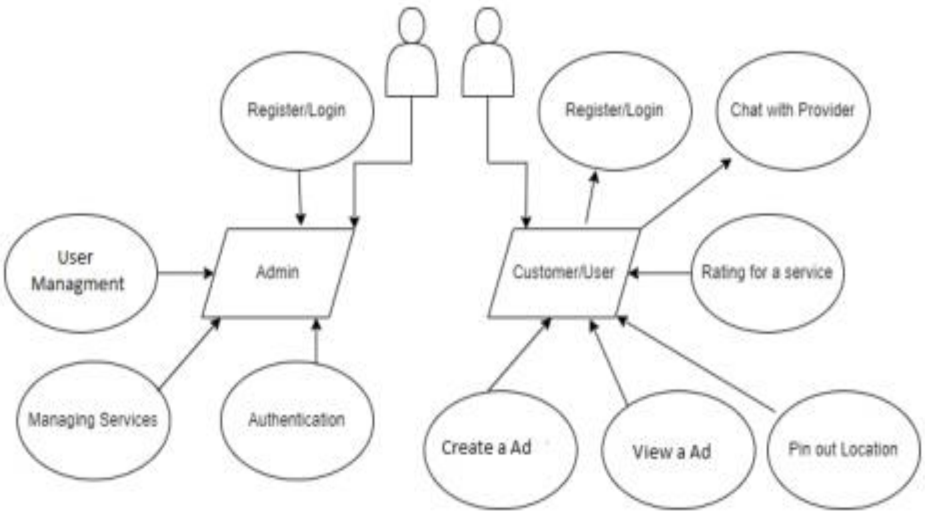
System Action: Initiates the process of coordinating the return of the found item between the user and the finder.

Communication: Server facilitates communication between the user and the finder.

9. End of Sequence:

- The sequence concludes when the user successfully retrieves the lost item or when the process ends without a match.

1.12 Data Flow Diagram:



1.13 Data Flow Diagram Description:

A Data Flow Diagram (DFD) is a visual representation of how data flows through a system. In the context of a Lost and Found app, the DFD can illustrate how information moves between different components or processes. Here's a description of a DFD for the Lost 2 Find app:

Components:

1. User Interface (UI):

- Represents the app's interface where users interact with the system.

2. Lost Item Reporting Process:

- Manages the process of users reporting lost items.

3. Search Process:

- Handles user requests to search for lost items based on criteria.

4. Communication Process:

- Facilitates communication between users and finders.

5. Finder Interface:

- Represents the interface through which finders interact with the system.

6. Database:

- Stores information about reported lost items, user details, and finder details.

7. Notification System:

- Manages the sending of notifications to users about potential matches or communications.

Data Flows:

1. User Reports Lost Item:

- Flow: From UI to Lost Item Reporting Process
- Description: The user provides details about a lost item through the app's interface, initiating the reporting process.

2. Lost Item Data to Database:

- Flow: From Lost Item Reporting Process to Database
- Description: The information about the reported lost item is stored in the database for future reference.

3. User Searches for Lost Item:

- Flow: From UI to Search Process
- Description: The user initiates a search by entering criteria such as category, location, or description.

4. Search Results to User Interface:

- Flow: From Search Process to UI
- Description: The search process retrieves relevant lost items from the database, and the results are displayed in the user interface.

5. User Communicates with Finder:

- Flow: Between UI, Communication Process, and Finder Interface
- Description: The user communicates with the finder through the app's interface, facilitated by the communication process.

6. Communication Data to Database:

- Flow: From Communication Process to Database
- Description: Communication data, such as messages between users and finders, is stored in the database.

7. Notification to User:

- Flow: From Notification System to UI
- Description: The notification system sends notifications to users about potential matches or messages from finders.

8. Finder Reports Found Item:

- Flow: From Finder Interface to Database
- Description: Finders report details about found items through the app's interface, and this information is stored in the database.

9. Return Coordination Process:

- Flow: Between UI, Communication Process, and Finder Interface
- Description: When a match is confirmed, the system initiates a process to coordinate the return of the found item between the user and the finder.