

UNIVERSITY OF BUEA

**Faculty Of Engineering and Technology**

CEF440

**INTERNET PROGRAMMING AND MOBILE PROGRAMMING**

**Design and Implementation of a Mobile-Based Archival and Retrieval of Missing Objects Application using Image Matching**

**System Modeling and Design**

*Submitted to:*

Dr. Nkemini Valery

2023/2024

**Members**

1. DJIALEU TANKOUA KEANE RANDY FE21A170
2. MBELI NGO JOEL  FE21A229
3. NANGLEFACK LEODIA FIETSOP FE21A244
4. NKEMZI FOLEFACK GIL FE21A276
5. NSIELA SUHBISI DENZEL FE21A286

Contents

[1. Introduction 1](#_Toc167745486)

[2. System Overview 1](#_Toc167745487)

[3. System Requirements 1](#_Toc167745488)

3.1. Functional requirements………………………………………………...1

3.2. Non-functional requirements……………………………………………2

[4. System Modeling 3](#_Toc167745489)

4.1. Use Case diagram……………………………………………………….4

4.2. Context diagram…………………………………………………………4

4.3. Class diagram……………………………………………………………6

4.4. Sequence diagram……………………………………………………….6

4.5 Deployment diagram…………………………………………………….7

[5. Implementation Technologies 10](#_Toc167745490)

[6. Testing 11](#_Toc167745491)

[7. Conclusion 12](#_Toc167745492)

# 1. Introduction

This report details the system modeling and design for a mobile application focused on facilitating the retrieval of lost objects using image recognition technology. This application aims to bridge the gap between lost and found objects by creating a platform for users to report missing items and browse reported found objects.

## 2. System Overview

The application leverages image matching to compare reported lost objects with images of found objects uploaded by other users or authorities. The core functionalities include:

* Reporting lost items with image and details.
* Reporting found items with image and details.
* Browsing reported found items based on various criteria.
* Claiming found items and facilitating communication between finder and potential owner.
* Administrator functionalities for managing user accounts and reported items.

## 3. System Requirements

**3.1 Functional Requirements**

Users should be able to register and create profiles.

- Users should be able to register and create profiles.

- Users should report lost items by capturing an image and providing details.

- Users should report found items by capturing an image and providing details.

- Users should be able to browse reported found items based on various criteria.

- Users should be able to express interest in a found item and initiate communication with the finder.

- Users should be able to manage their profiles and update information.

- Administrators should manage user accounts and review reported items.

**3.2 Non-Functional Requirements**

* The application should be user-friendly and intuitive.
* The image matching algorithm should be reliable and accurate.
* The application should have adequate security measures to protect user data and image privacy.
* The application should be performant and responsive on various mobile devices.

## 4. System Modeling

In this section, we dive into the fascinating world of system modeling, exploring how we dissect complex systems into manageable pieces, visualize their interactions and predict their behaviors. So system modeling acts as a blue print that maps out all what we want our system to be and functions.

**4.1 Use Case Diagram**

The use case diagram showcases the functionalities of the application from the user's perspective.

**Actors**

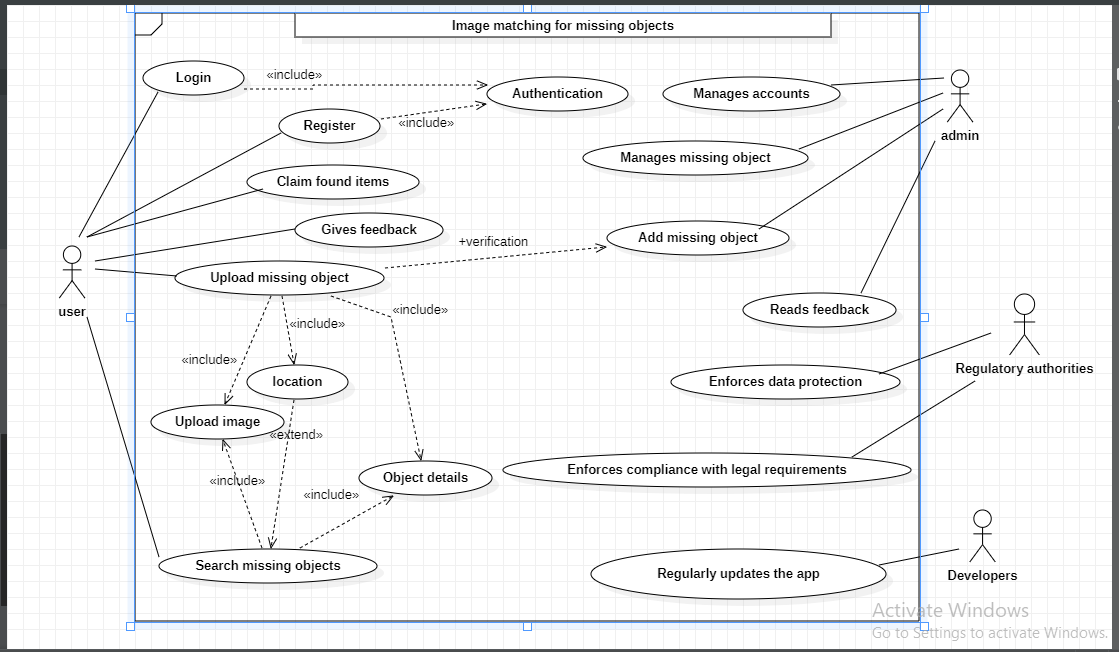
-Users( person who has missing objects and person who found missing objects).\

-Administrator

-Developer

-Regulatory authorities.

The use case diagram for our system is given below:



So for our application, a user must be registered and gets to login if and only if he/she gets authenticated. After logging in, they can upload their missing objects which includes an image, where it was lost and more details about the object so as to facilitate image matching of the object. From there, if the object is seen and found by someone else, he provides all necessary infos when uploading it.

While the admin manages all user accounts, manages all found item (finds and verifies that the missing object has been found.

Then the developers regularly update the app so as to solve issues and bring about more user satisfaction.

**4.2 Context Diagram**

The context diagram depicts the high-level interactions between the application and external entities:

* **Actors:**

\* User (Lost Item Reporter)

\* User (Found Item Reporter)

\* Administrator

* **Data Flows:**

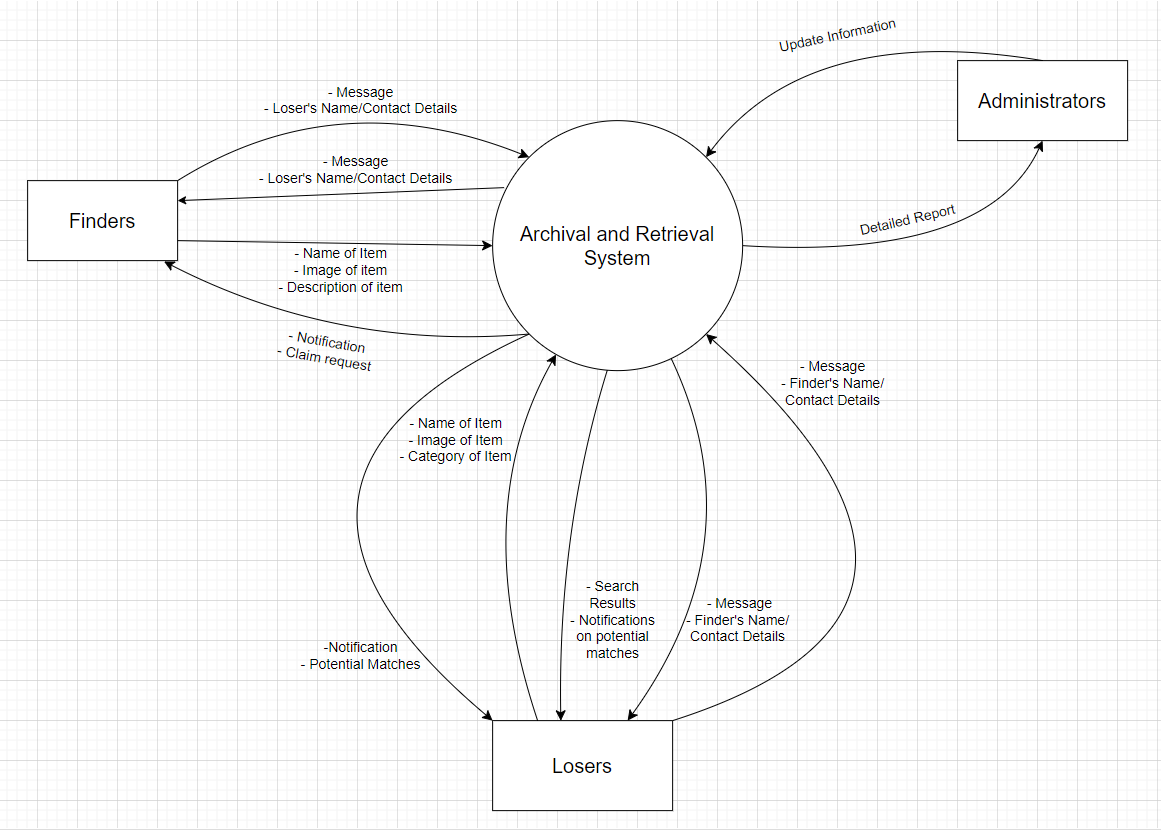
\* User reports lost/found items with images and details.

\* Administrator manages user accounts and reviews reports.

\* The application receives data from users and transmits it to the database.

\* The application retrieves data from the database to display relevant information to users.

In the diagram it is clear to see that the main actors in the system are the end users, and the administrators act as secondary users, performing routine system checks and doing background maintenance in coordination with the developers.



**4.3 Class Diagram**

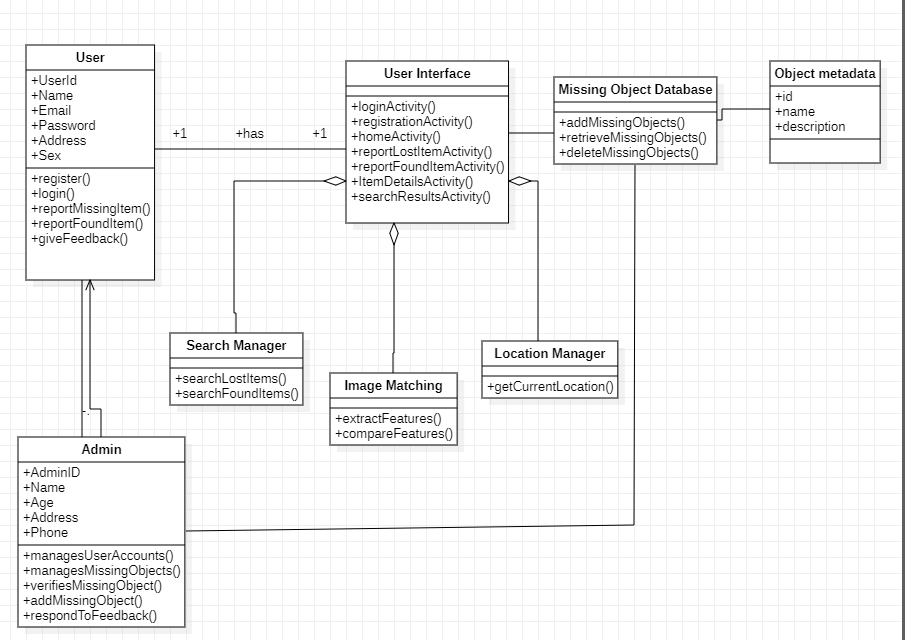
The class diagram represents the application's core classes and their relationships in our system.

* Classes:

\* User

\* User interface

\*Admin



*Class diagram*

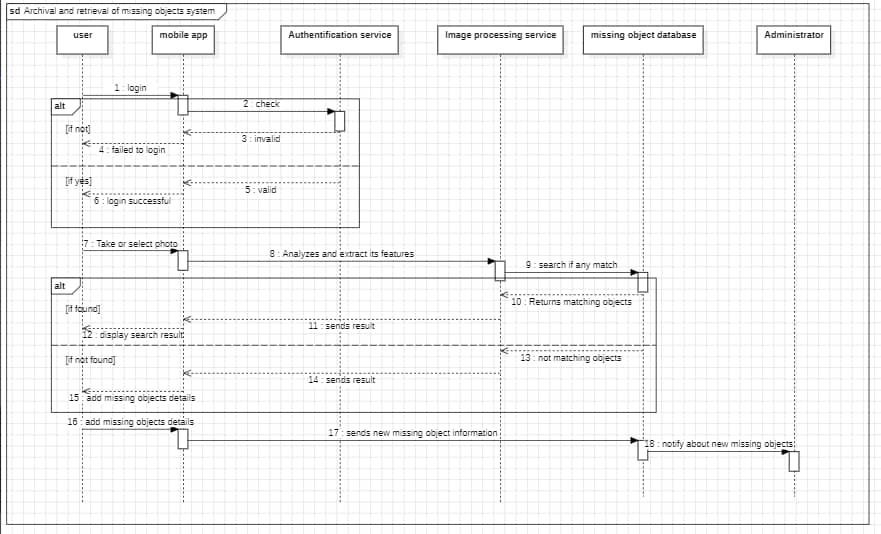
**4.4 Sequence Diagram**

The sequence diagram demonstrates the message flow between the application and its internal components while a user reports a lost item.

**Actors:** User, Administrator

**Objects:** Mobile app, Authentication service, Image Processing Service, Missing Object Database

Below is the diagram



*Sequence diagram*

The sequence diagram shows the process of a user searching for a missing object using the mobile app. The mobile then interacts with an authentication service which verify the user’s login credentials. Then, it interacts with an image processing service to analyze a photo of the missing object. The image processing service then interacts with a missing object database to see if there is a match. Finally, the mobile app displays the results to the user. If there is no match, the user can add information about the missing object to the database.

**4.5 Deployment Diagram**

The deployment diagram depicts the physical architecture of the application system.

**Components:**

\*User’s mobile device

\* Web-based admin portal

\* Application server

\* Database

\* Image processing service

\* Notification service

NB: This components are connected by TCP/IP

**Communication:**

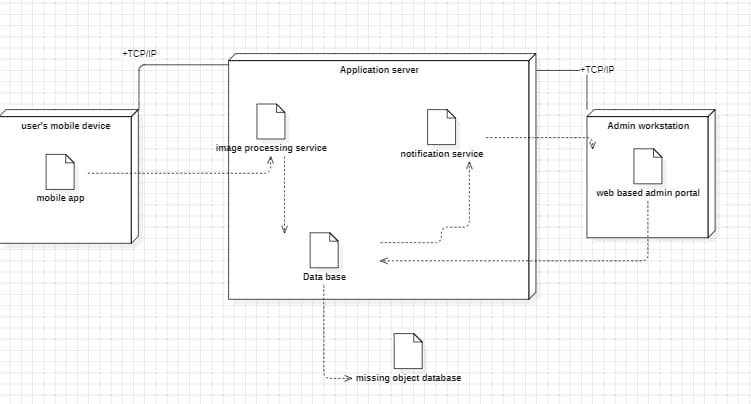
-The mobile app runs on the user’s mobile device. It communicates with the application server to access data.

-The web-based admin portal allows administrators to manage the system. It also communicates with the application server.

-The application server runs the business logic of the application. It communicates with the database, image processing and notification service.

-The image processing service analyses images from the mobile app.

- the notification service sends notifications to users.



*Deployment Diagram*

## 5. Implementation Technologies

* **Programming Language: Java (Android)**

Developing with Java ensures our application runs on a vast majority of android devices which was found to be the most owned devices among interviewed users. Also java comes with a very good ecosystem of libraries such as libraries for user interface creation and data handling which of course saves us time , thereby allowing us focus on the innovative features that make the app unique from others.

* **Development Framework: IntelliJ IDEA**

Having a large and active community, this means a wealth of resources such as plugins and developer community is readily available. This therefore will be useful as working with a team spirit will help us quickly achieve our goals for our application.

* **Database: Cloud-based database (e.g., Firebase)**

Firebase in particular provides a secured, scalable and real time reunion solution to our ever-needed growing database for lost and found object, which in turn encourages more users to register their lost items and also increasing the chances of users also seeing their lost items all within a well secured database system.

* **Image Processing Library: OpenCV**

OpenCV unlocks the magic of image recognition which is the very focus of our application, thereby increasing the chances of the rightful owners matching their missing items.

## 6. Testing

The application will undergo various testing phases, including unit testing, integration testing, system testing, and user acceptance testing (UAT). Unit testing will focus on testing individual components such as the image processing module and database interactions. Integration testing will ensure that different modules of the application work together seamlessly.

**System testing:** This will verify the application's compliance with functional and non-functional requirements. The complete app will be tested in a simulated real-world environment to assess overall functionality and performance. (Manual testing using emulators or real devices)

**User acceptance testing:** This will involve real users interacting with the application to evaluate its usability, reliability, and performance. Test cases will cover scenarios such as reporting lost and found items, browsing found items, claiming items, and profile management.

**Functionality Testing**: Verify core functionalities like user registration, login, item addition (lost and found), image processing, search, and match notifications.This will be done manually.

**Performance Testing:** Assess app responsiveness, loading times, and scalability under various user loads.

**Usability Testing**: Evaluate user interface (UI) design, ease of navigation, and overall user experience.

**Security Testing**: Identify potential vulnerabilities related to data security, user authentication, and authorization.

**Unit Testing:** Individual app modules will be tested in isolation to ensure they function as designed. (Implemented within the development process using tools provided by IntelliJ IDEA)

**Integration Testing**: Modules will be integrated and tested as a whole to identify any integration issues. (Manual testing and potentially using additional testing frameworks)

## 7. Conclusion

In conclusion, this report has outlined the system modeling and design for a mobile application aimed at facilitating the retrieval of lost objects through image recognition technology. By leveraging image matching algorithms and providing intuitive user interfaces, the application seeks to connect users who have lost items with those who have found them, thereby increasing the chances of successful retrieval. With thorough testing and implementation of robust security measures, the application aims to provide a reliable and secure platform for users to report and reclaim their lost belongings.