UNIVERSITY OF BUEA

REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

P.O. Box 63,

Buea, South West Region

CAMEROON

Tel: (237) 3332 21 34/3332 26 90

Fax: (237) 3332 22 72

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING ALGORITHMS AND ADVANCED MACHINE VISION TECHNIQUES

PHASE 4: SYSTEM MODELING AND DESIGN

By:

Group 6

Supervisor:

Dr Nkemeni Valery

University of Buea

2nd Semester 2023/2024 Academic Year

LIST OF PARTICIPANTS

NAMES	MATRICULE	Speciality
KOUETE TEPE KENNETH	FE21A220	Software
ZELEFACK MARIE	FE21A330	Software
EFUETAZOH ASONG RODERIC	FE21A179	Network
TAKO DIEUDONNEVOJONG	FE21A310	Software
BUINFUN MONIE JULIUS	FE21A154	Software

TABLE OF CONTENTS

1. INTRODUCTION

The problem of lost and found items can be mitigated by leveraging advanced image matching algorithm. This document outlines the system modelling and design approach approach for developing and efficient and robust system for archival and retrieval of missing objects using image matching algorithms. This system aims to provide a seamless user experience for reporting lost items, uploading images and searching for potential matches and receiving notifications.

We will employ various UML diagram. These diagram will help in visualizing the system components, their relationships and the flow of operations within the system. The following UML diagram will be used

- Context diagram
- UseCase diagram
- Class diagram
- Sequence diagram
- Deployment diagram

2. CONTEXT DIAGRAM

A context diagram is a diagram that provide a high-level overview of a system and how the system interacts with external entities and also the processes of the system.

2.1 Components

A context diagram is made up of 3 components that is the system boundary, external entities and data flow.

1. System Boundary

The system boundary represents a single process or entity in the center of the diagram and clearly label the system to indicate what is being analysed.

2. External Entities

These are the external actors that interact with the system. Below are the external entities.

- User

Users interact with the system by singing in or signup to the system, upload images of missing object and search for the missing object.

- Admin

The admin interacts with the system by signing in, manages the system by adding, updating and removing image object and users.

- Database

The database interacts with the system by storing and retrieving datas of users and Image objects.

- Cloud Infrastructures

The system interacts with cloud infrastructure to backup data, data storage and also for the deployment of the system.

- Matching Algorithm

The system interacts with matching algorithm service to search and retrieve potential image match of the missing object

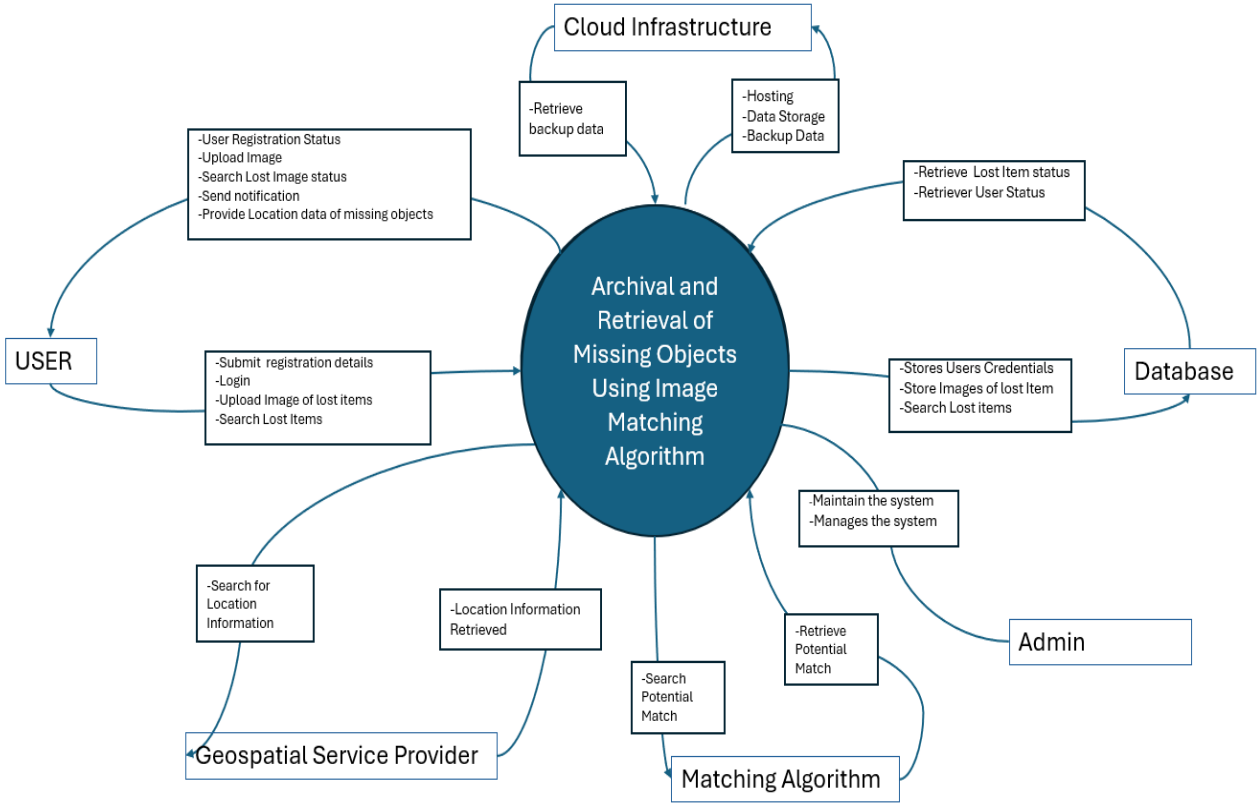
- Geospatial Service Provider

The system interacts with geospatial service providers to track and retrieve the location of the missing objects.

3. Data Flows

These are arrows which are used to show the flow of information between the system and the external entities.

2.2 CONTEXT DIAGRAM OF ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING ALGORITHM



3. USE CASE DIAGRAM

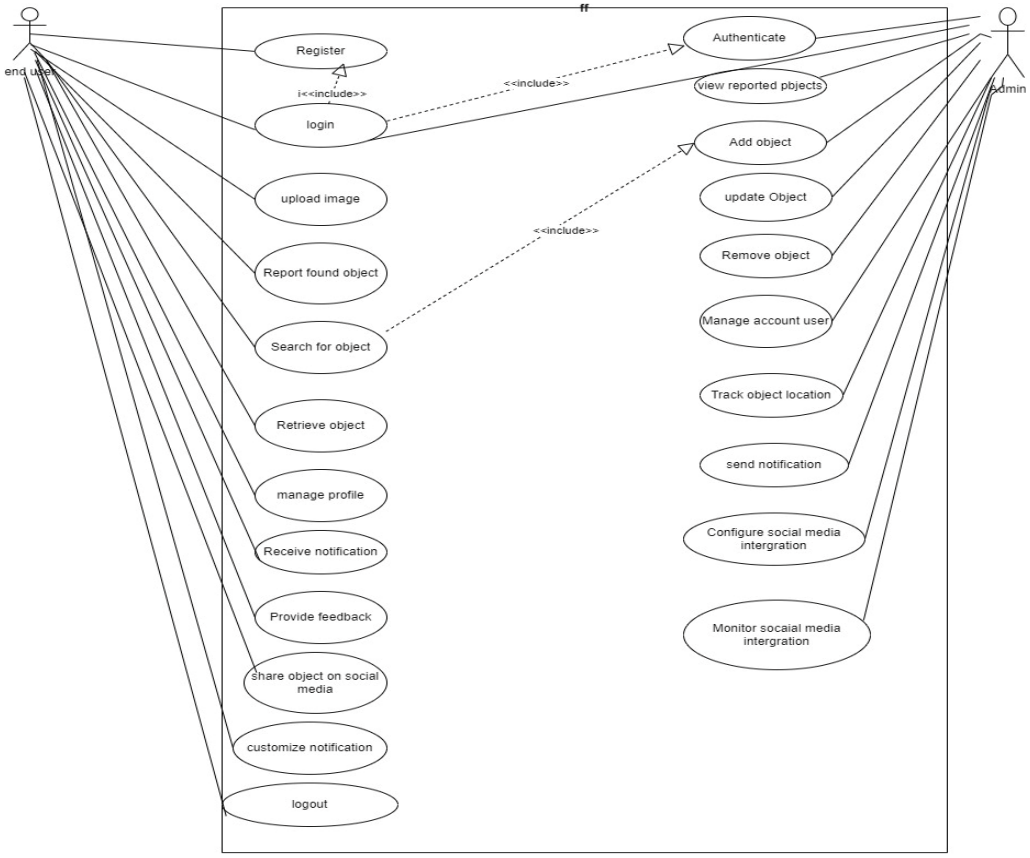
A use case diagram is a graphical representation of a user's possible interactions with a system. A use case diagram, shows various use cases and different type of users the system has .

For the mobile based Archival and Retrieval of missing item we have the admin and end user as our main actors , the use cases and lastly the system.

The primary actor is the **"End User" ,**the user can perform various actions such Registering, Loggin in , upload image, Report found object, Search for an object , Retrieve object , manage his or her profile, receive notification, provide

feedback, receive notification,share object on social media and customize his or her notification, logout, and our primary actor is the **“Admin”,** and he should be able login , authenticate , view reported objects, add object, update object, remove object, manage account user, track object location, send notification, configure social media intergration, monitor social media intergration etc , The use case below shows a clear illustration

3.1 USECASE DIAGRAM OF ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING ALGORITHM



4. CLASS DIAGRAM

The class diagram for an archival and retrieval of missing objects using Image matching algorithm illustrates the structures of the system by showing it classes, attributes, operations and the relationships among them. It provides a high-level

overview of how different components of the system interacts with each other. The following are the classes for the archival and retrieval of missing object system.

4.1 CLASSESS AND THEIR COMPONENTS

1. USER CLASS: Manages user-related activities like registration, authentication image upload etc.

Attributes

UserID datatype int – This is the unique identifier for the user

Username datatype “String” : The Username for the user

Password datatype “String”: Password for user authentication

Email datatype “string”: The users email

Phone number datatype “string”: The users phone number

Operations

The operations that the user class can perform are registration, login, manage profile, upload Image, delete Image, search for object by uploading image, report found object, get notification and provide feedback

2. ADMIN CLASS: Manages the system and provides supports.

Attributes

AdminID datatype “int”: This is the unique identifier for the Admin

Username datatype “string”: The username for the Admin

Password datatype “string”: The password for admin authentication

Operations

The operations that the ADMIN CLASS perform are manages users, add missing Object Image, update Missing Object status, Delete Missing Object Image, send Notifications.

3. DATABASE CLASS: Manages data storage and retrieval

Attributes

databaseName datatype “String”: The name of the database example User database

databaseType datatype “string”: The type of database example MongoDB or MySQL

Operations

The operations the database class perform is to store, retrieve and authenticates informations that will be inputted into the system.

4.IMAGE CLASS: Represent images that will uploaded by the users.

Attributes

imageID datatype “string”: The unique identifier for the image

UserID datatype “int”: The unique identifier for the user that uploaded the image.

imageData datatpye “Blob”: To store binary data of the image

uploadDate datatype “Date”: The date of the uploaded image

Operations

The operation that the IMAGE CLASS perform is to analyse the image uploaded by the users

5. LOCATION CLASS: Provides the locations of the missing objects

Attributes

LocationID datatype “int”: Unique identifier for the Location

Latitude datatype “double”: Coordinates for the precise geographical positioning

longitudes datatype “double”: Coordinates for the precise geographical positioning

description datatype “string”: description for the geographical area

Operations

The CLASS LOCATION gets the location, provide location of the missing objects.

6. NOTIFICATION CLASS: Represents notification sent to the users

Attributes

NotificationID datatype “int”: Unique identifier for the notification service

UserID datatype “int”: Unique identifier for the user

Message datatype “string”: Message sent to the user

sendDate datatype “Date”: The date of the message

Operation

The operation of the notification class is to send notifications.

7.MATCHING ALGORITHM: Represent the algorithm used to find potential matches

Attributes

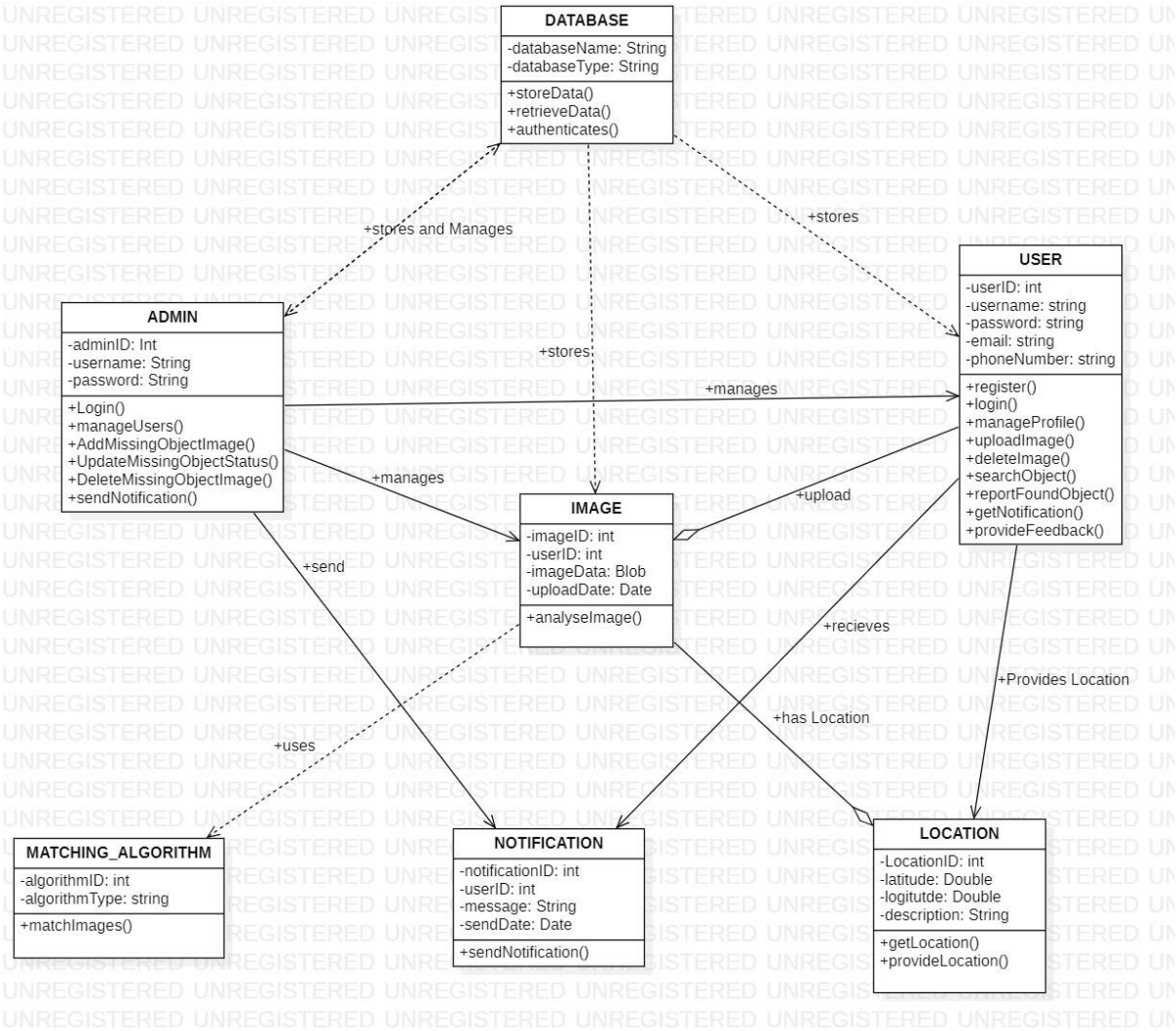
algorithmID datatype “int”: Unique identifier for the matching algorithm

algorithmType datatype “string”: the type of matching algorithm that will be use example feature-based algorithm based on key features or points in the image

operations

To match images that will be uploaded into the system and the images of the found objects

4.2 CLASS DIAGRAM OF ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING ALGORITHM



3. SEQUENCE DIAGRAM

The sequence diagram shows how the users (actors) interact with the system components during the archival and retrieval of missing objects. It shows the sequence of messages exchanged between objects to perform a specific function, providing a clear view of the interactions and the order in which they occur.

3.1 KEY COMPONENTS FOR THE SEQUENCE DIAGRAM

Actors:

- **User:** The person using the system to upload, report and search for missing objects using image
- **Admin:** The Person that manages the system and users of the system

Objects

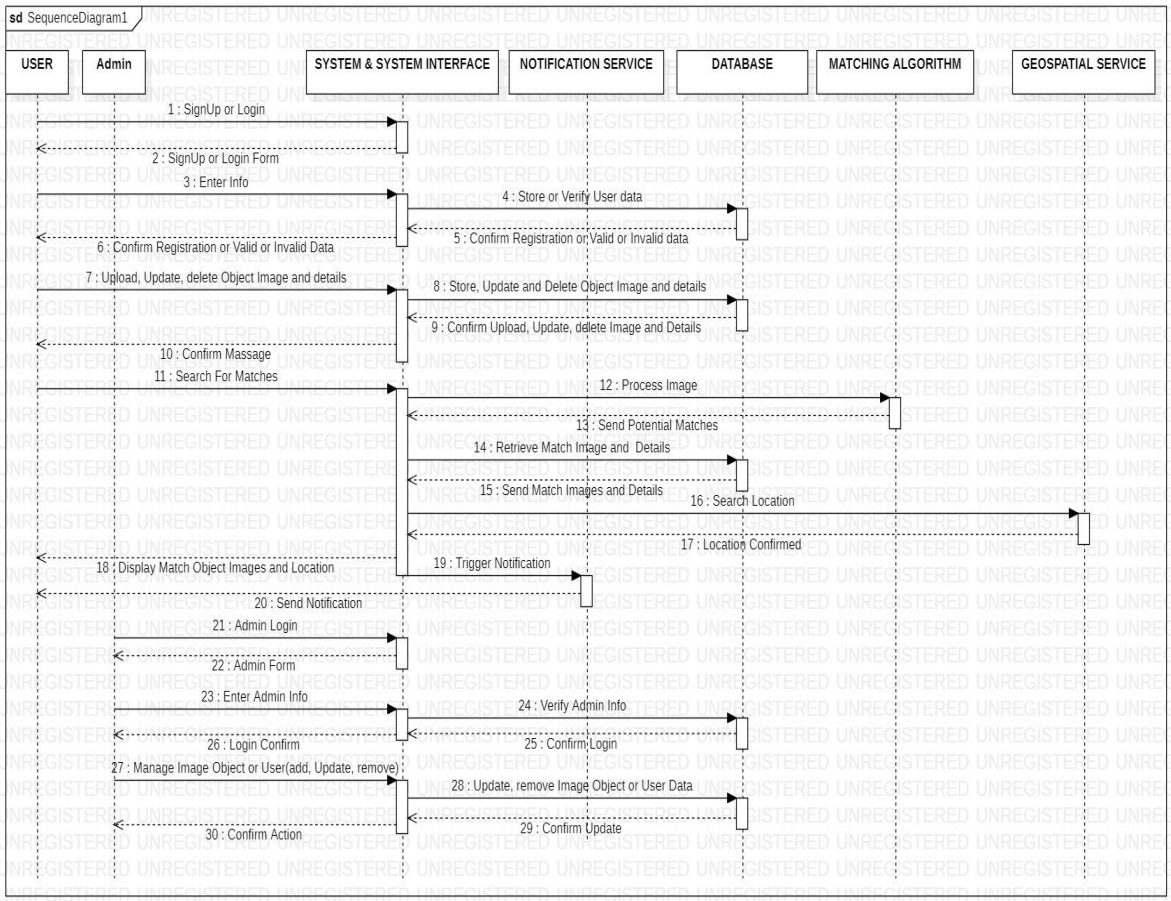
- **System Interface:** The front-end interface through which the user interacts with the system
- **System:** Performs a set of actions in response to interaction from external actors
- **Notification Service:** Sends Notifications to the users regarding the status of their uploads and search results
- **Database Service:** Manages data storage and retrivals. It store relevant information and retrieves when needed
- **Matching Algorithm:** Executes the image matching process to find potential matches by comparing image to find similarities and determine possible matches.
- **Geospatial Service:** Provides the Location of the missing objects

3.2 EXPLANATION OF INTERACTION

- **SignUp/Login:** The user or admin interacts with the system interface to signup or login using the signup or login form that will be provided by the system.
- **Enter Info:** The users or admin will enter his/her information for Login or SignUp.
- **Store/Verify User data:** The entered data from the user or admin will stored in the databased during signup while during login, the entered data from the user or admin will be verified in the database if any potential data exist
- **Confirm registration/ Valid / Invalid data:** A message will be sent to the user or admin to confirm their registration during signup or to provide an invalid or valid message during login if their credentials exist or not in the database
- **Upload/Update/delete Object Image and details:** The user can then upload, Updates, deletes Image Objects and details after a successful login using the system interface
- **Store/Update/Delete** Object Image and details: The Object image is then stored or updates or deleted from the database

- **Confirm Upload, Update, delete Object Image and Details:** A confirmation message is then sent to user that the Object image and details has been uploaded, updated and deleted through the system interface
- **Search For Matches:** After uploading or updating the object image, the user can now search for potential match.
- **Process Image:** The search imaged is then process using matching algorithm to find any match.
- **Send Potential Matches:** The potential matches are then send to the system.
- **Retrieve Match Image and Details:** The system then retrieve match image and details from the database which the database sent the match images and details for verification
- **Search Loaction:** After that the system then search for the location of the missing object using geospatial service. The location is then confirmed which is then sent to the system interface then the matched Image object and location is the displayed to the user.
- **Trigger Notification:** The system will now trigger a notification using the notification service which will be sent to the user
- **Manage Image Object or User(add, Update, remove):** The user can add, update or remove Object image or User using the system interface which will be updated in the database
- **Confirm Action:** A confirm message will be sent to the admin that the Object image or user has been added, updated or removed.

3.3 SEQUENCE DIAGRAM OF ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING ALGORITHM



4. DEPLOYMENT DIAGRAM

A deployment diagram provides picture of how the physical system will look when all it's put together.It

Shows how artifacts are deployed on system hardware and how pieces of hardware connect to one another

It comprises of nodes, in our case (that of a mobile base archival and Retrieval of missing objects) we have

- Mobile device
- Application Server

- Database Server
- Notification Server
- Social Media

So a node is basically a computational resource upon which artifacts are deployed for execution and artifacts.

Below we have our various nodes and artifacts containing all the modules

- Mobile device

  Artifacts:

  - ✓ **Mobile App** : this artifact indicates that the mobile app is deployed on a mobile device

    and it contains UI, image capture, notification, and image matching client functionalities

    - **User interface:** The part of the app where users interact with the system. It includes screens and controls for uploading images, viewing notifications, etc.
    - **Image capture**: Allows users to take photos of lost items using the device's camera.
    - **Notification:** Manages push notifications and email alerts to inform users of potential matches.
    - **Image matching Client**: Preprocesses images before sending them to the server for matching.

- **Application server**

  Artifact:

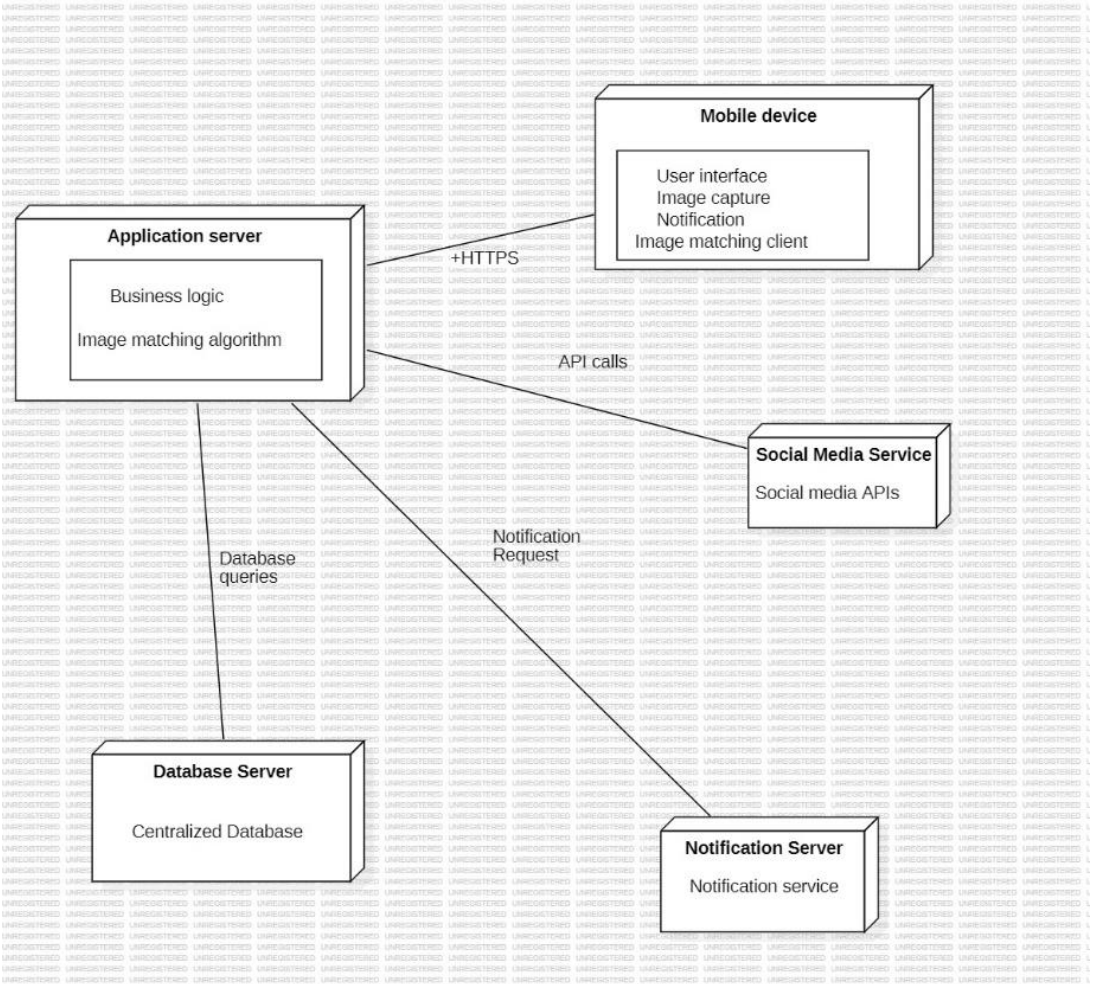
  - ✓ **Application Backend** : it is deployed on the application server and it contains modules such as Business logic and Image matching algorithm

- **Business logic:** it handles the core functionalities and operations of the application, handling requests from the mobile app.
- **Image matching algorithm:** Processes the images sent by the client, using computer vision techniques to find matches in the database.

- **Database Server**
  - ✓ **Database Schema and Data:** it is deployed on the database server, it includes the structure(schema) of the databases and the actual data (images and meta data).It contains modules such as Centralized Database.
    - **Centralized Database :** Stores images of lost items along with metadata such as descriptions, timestamps, and locations. This database supports efficient indexing and retrieval of stored information

- **Notification Server**
  - ✓ **Notification System:** this artifact includes the service responsible for sending push notifications and emails . It is deployed on the notification server. It includes modules like notification service
    - **Notification Service**: Sends push notifications and emails to users when potential matches for their lost items are found.
- **Social Media Service**
  - ✓ **API intergration:** this artifact includes the code and configuration required to interact with social media platforms . It is not directly deployed but it is used by the application server to make API calls
    - Social Media APIs: Provides integration with various social media platforms to share information about lost items, leveraging crowdsourcing for enhanced search effectiveness.

4.1 DEPLOYMENT DIAGRAM OF ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING ALGORITHM



For the connection between the nodes ,

- **HTTPS**: This connection represents secure communication between the mobile app and the application server. HTTPS ensures that data transferred between the client and server is encrypted and secure.
- **Database queries**: The application server queries the database server to store and retrieve images and metadata. This connection handles all database operations needed for the application to function.
- **Notification Requests:** The application server sends notification requests to the notification server. When a potential match is found, the application server instructs the notification server to send a push notification or email to the user.

- **API Calls:** The application server makes API calls to various social media platforms. This allows users to share information about their lost items on social media, engaging the community in the search process.

5. CONCLUSION

By using these UML diagram, we will create a comprehensive and detailed model of the system, ensuring clarity in design. This structured approach will help in the systematic development and deployment of the system addressing the problem of lost and found items with precision and efficiency