

REPUBLIC OF CAMEROON REPUBLIC DU CAMEROUN

Peace-Work-Fatherland Paix-Travail-Patrie

University of Buea Universite de Buea

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CEF440: Internet Programming and Mobile Programming**

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

**Academic year 2023/2024**

SUPERVISED BY

**Dr Nkemeni Valery**

**PRESENTED BY MEMBERS OF GROUP 9**

|  |  |
| --- | --- |
| **MATRICLE** | **NAME** |
| FE21A128 | AHOUMO TEMATEU ROXANE PHILIPPINE |
| FE21A152 | BOUCHUKE BABILA DANIEL |
| FE21A166 | DERRICK MBUNBO FORCHA |
| FE21A325 | TIANI PEKINS EBIKA |
| FE21A303 | SAMUEL OSOH |

**ACKNOWLEDMENTS**

**A**cknowledging the assistances of DR NKEMENI VALERY the course instructor and an elderly for his marvelous work and knowledge mounted upon me through this years of studies, my team and some close friends for their guidance and care. May the good LORD continues to guide and protect you all .Thanks

Table of Contents

[Table of Contents 5](#_Toc166508028)

[1. INTRODUCTION 7](#_Toc166508029)

[2. WHAT IS REQUIRMENT ANALYSIS 8](#_Toc166508030)

[3. PROCESSES INVOLVED IN REQUIEMENT ANALYSIS 8](#_Toc166508031)

[4 TECHIQUES USED IN REQUIMENT ANALYSIS 20](#_Toc166508032)

[5 BENEFIT OF REQUIREMENT ANALYSIS 22](#_Toc166508033)

[CONCLUSIONS 23](#_Toc166508034)

[REFERENCES 24](#_Toc166508035)

Table of figures

[Figure 1: Use case diagram for mobile base object archival application.](#_Toc155685950) 10

[Figure 2: Some key non-Functional Requirements 15](#_Toc155685952)

[Figure 3: Flow chart showing summarized processes of the system 20](#_Toc155685951)

## INTRODUCTION

The evolution of mobile computations and wireless communications technologies has embedded itself in our lives due to its rapid development. Ubiquitous wireless devices used in daily tasks have led to the revolution of Mobile Government (m-Government) systems aiming to deliver services and information anywhere and anytime to citizens through the utilization of mobile technologies and applications. However, only the performance of the delivered services establishes its usefulness with many undiscovered applications to such innovations, especially since this is a desired area for further development with relation to smart cities. One such application is the lost and found items retrieval utility. The loss or misplacement of personal belongings is a common occurrence that can cause inconvenience and distress to individuals. Traditional methods of searching for missing objects, such as posting flyers or contacting lost and found services, are often time-consuming and ineffective. To address this issue, this project proposes the development of a Mobile-Based Archival and Retrieval of Missing Objects Application using Image Matching. This project aims to leverage mobile technology and image matching algorithms to create a robust and user-friendly platform for archiving and retrieving missing objects. The mobile application will allow users to capture images of lost items, store them in a centralized database, and perform real-time image matching to identify potential matches.

## WHAT IS REQUIRMENT ANALYSIS

Requirements Analysis is the process of defining the expectations of the users for an application that is to be built or modified. It involves all the tasks that are conducted to identify the needs of different stakeholders which focuses on transforming the raw data gotten from requirement gathering into a clear and actionable set of instructions. This involves analyzing, prioritizing, and refining the requirements.

This processes of englobes the identification of inconsistencies, redundancies, and missing information in the gathered requirements.

## PROCESSES INVOLVED IN REQUIEMENT ANALYSIS

#### **3. 1: Identify Key Stakeholders and End-Users**

**How do we identify stakeholders**: the most user technique to identify stakeholders is through **stakeholder analysis?** Which is the process of collecting information about any person that will be impacted by (or can impact) your project. The various stake holders are;

* The end users
* Developer
* Testers and
* Administrators

#### **3.2: Capture Requirements**

##### 3.2.1: Hold one-on-one Interview

From the one-on-one interview carried out at “Mile 17 buea, with Over 2 "moto boys”, 8 bus drivers and 10 taxi drivers”, and in addition with the “two workers from Camtel office”

**QUESTIONS ASKED WERE OF THE FORM:**

* How often do u lose your lost document (ID card, birth certificate, death certificate and others)?
* If yes. What do you usually do to find the said lost document?
* Where do you often or occasionally do you recover lost document?
* What measures do you take after recovering the documents to make sure they are returned back to the owner?

Later on, results and analysis from the interview were gotten as listed below;

QUESTION 1:

Out of the 20 people,

* 15 and more had never lost their documents.
* 3 and less had their documents stolen from their car and other places.

QUESTION 2:

* Those whose document was stolen reported to the police for further investigation and findings

QUESTION 3:

Out of the 20 people,

* 16 of these persons typically find missing documents in their vehicle ( more than 5 times)
* 3 of them have rarely found missing documents in their vehicle ( < 2 times )
* None of them have never seen or picked any missing documents

QUESTION 4:

* 7 of these people return the lost document to the "drivers association"
* 8 of the bus drivers return the missing document to the nearest bus stop

Less than 3 of them tried to personally contact the owners due to the numerous "drivers association" locations making them pay a fine before searching for their missing documents.

At the camtel office,

* Both workers reported they have never lost their documents. They reported they are always carefully when handling their documents ( especially ID card ) as it has become a second nature to them
* Both mostly found missing documents on waiting chairs or on the floor and they usually placed the missing document on the front of the desk visible to everyone in search of.

##### 3.3.2Utilize Use Cases

Use cases describe the interaction between the system and external users that leads to achieving particular goals.  
Each use case includes three main elements:

**Actors:** These are the external users that interact with the system  
  
 **System**: The system is described by functional requirements that define the intended behaviours of the application.  
  
 **Goals:** The purposes of the interaction between the users and the system are outlined as goals.  
  
There are two ways to represent use cases:

**A use case diagram** and a use **case specification**

**A Use case diagram**.

The use **case diagram** doesn’t contain a lot of details. It shows a high-level overview of the relationships between actors, different use cases, and the system. With respect to the functional and non-functional requirements listed above, the use case diagram includes the following main elements.

* **Use cases**. (Log in, registration, archiving objects etc.).
* **Actors**. Users/registered, System Administrator etc.
* **Associations.** E.g. “Registered user to archiving objects “,”user to register .here, Associations are drawn with lines showing different types of relationships between actors and use cases.

System boundaries. Boundaries are outlined by the box that groups various use cases in a system.

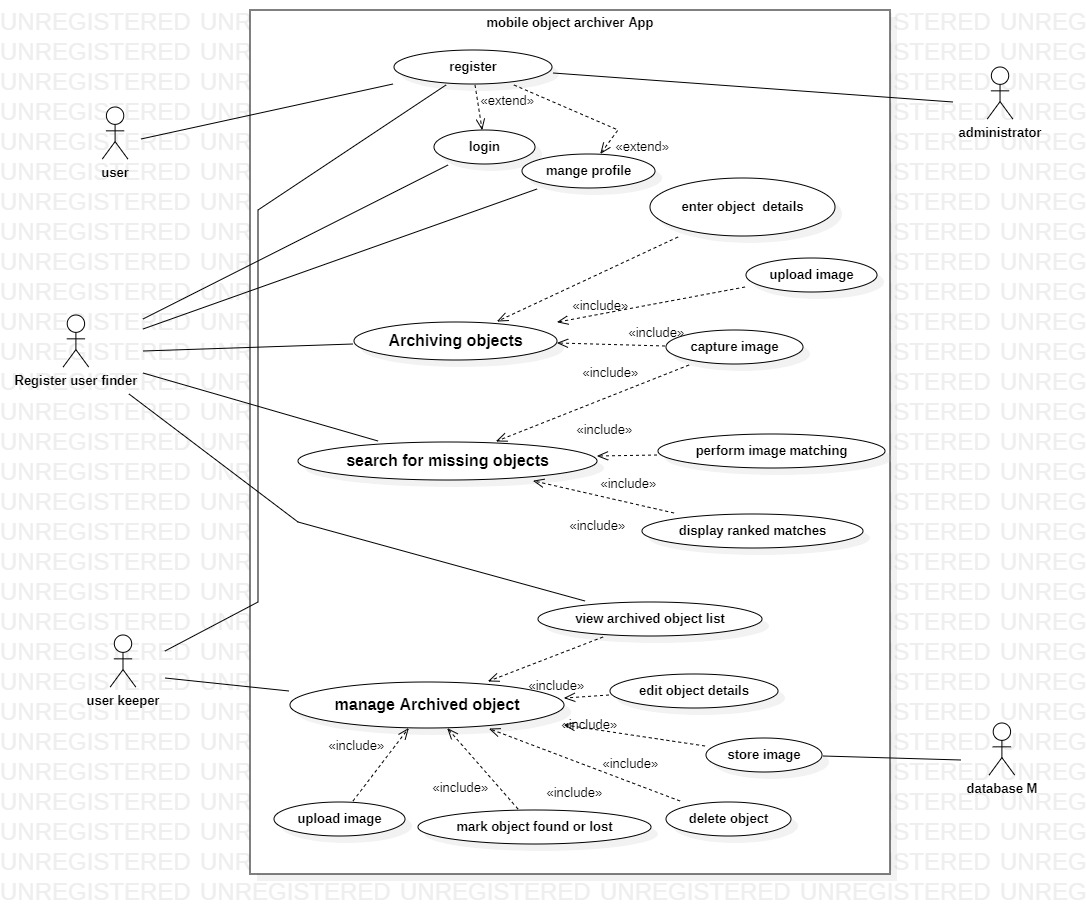


FIG 1: Use case diagram for mobile base object archival application.

B **Use case specification**

Represents the sequence of events and other information related to this use case.

These use cases delve deeper into the specific interactions between actors and the system, outlining the detailed steps involved in achieving their goals.

**Use Case 1: Archive a Found Object (Keeper)**

**Actor**: Keeper (Person who found an object)

**Goal**: Archive a found object in the system to potentially reunite it with its owner.

**Preconditions:**

* The keeper has a mobile device or computer.
* The keeper has an image of the found object.

**Steps:**

* The keeper launches the system's mobile app or web interface.
* The system prompts the keeper to sign in or create an account (if not already done).
* The system presents an interface for adding a new object.
* The keeper uploads an image of the found object.
* The keeper confirms the information and submits it.
* The system validates the uploaded image format and size.
* The system performs basic image pre-processing (e.g., resizing, noise reduction) on the uploaded image.
* The system stores the image and associated object details in the object archive.
* The system displays a confirmation message indicating successful object archiving.

**Postconditions:**

* The image and details of the found object are stored in the system's archive.
* The keeper can access the archived object information and details later if needed.
* If the image upload fails due to size or format limitations, the system displays an error message prompting the keeper to upload a valid image.
* The keeper can choose to cancel archiving the object at any step during the process.
* Use Case 2: Search for a Lost Object (Finder)

**Actor:** Finder (Person who lost an object)

**Goal**: Search the system's archive for a lost object using an image.

**Preconditions:**

* The finder has a mobile device or computer
* The finder has an image of the lost object.

Steps:

* The finder launches the system's mobile app or web interface.
* The system prompts the finder to sign in or create an account (if not already done).
* The system presents an interface for searching lost objects.
* The finder uploads an image of the lost object.
* (Optional) The finder can filter the search results by category (based on object details archived by keepers).
* The finder confirms the search criteria and initiates the search.
* The system validates the uploaded image format and size..
* The system utilizes an image matching algorithm to compare the search image with all archived object images.
* The system ranks the archived objects based on the degree of image similarity to the search image.
* The system displays a ranked list of potential matches, prioritizing objects with the highest similarity scores.
* The finder can view details and potentially a preview image of each retrieved object.

**Postconditions:**

* The system presents the finder with a ranked list of potentially matching objects based on image similarity.
* The finder can review the search results and identify potential leads for retrieving their lost object.

**Alternative Flows:**

* If the image upload fails due to size or format limitations, the system displays an error message prompting the finder to upload a valid image.
* If no matches are found within the archive, the system informs the finder and allows them to refine their search criteria (if applicable) or try again with a different image.
* Keeper marking a found object as "**retrieved" if claimed by the owner**.
* System administrator managing user accounts and system settings.
* Advanced search options based on object details provided by keepers.

##### 3.2.3: User story

It describes what exactly the stockholders want the system to do .this user centric approach helps to ensure that the final product aligns with the need of the people who at all time will interact with the application

These user stories describe the functionalities of the system from the perspective of different stakeholder:

* **User 2: As a finder who lost an object**
* As a finder who lost an object, I want to upload a picture of the lost object so the system can search for similar items in the archive and help me find it.
* As a finder, I want to be able to provide additional details about the lost object, like its brand, description, or category, to further refine the search.
* As a finder, I want to see a ranked list of potential matches based on how similar their images are to the image of my lost object.
* **User 2: Person who found an Object (Keeper)**
* As a keeper who found an object, I want to be able to archive the object in the system by uploading its picture and details like its category or any distinguishing marks.
* As a keeper, I want the option to mark an archived object as "found" if the rightful owner retrieves it, so it's no longer displayed in search results.
* **User 3: System Administrator**
* As the system administrator, I want to be able to manage user accounts and access permissions to ensure the security of the archived objects.
* As an administrator, I want to monitor system performance and storage usage to identify any potential bottlenecks and ensure smooth operation.
* As an administrator, I want to have tools to analyse archived object data and user search trends to improve the overall system effectiveness.
* **User 4: System tester**
* As the system testing unit , the system must properly be tested under varying conditions making sure that the system is responsive, easy to use and navigate, with clear instructions for uploading images and searching for objects.
* **User 5: System developers**
* As part of the development team, providing a platform for confidentiality that is users uploaded images and object information are stored securely within the system.

##### 3.2.4: Build prototypes

Before the product is ready, there’s often a need to see how functional requirements are implemented – or how the future product will work.

That’s why prototypes, wireframes, and mock-ups are created. They represent how the solution will look and give an idea of how users will interact with it.

This way, they help bridge the vision gaps and let stakeholders and teams get a shared understanding of products in development.

1. **prototypes**

Is actually an umbrella term for different early-stage deliverables that are built to showcase how requirements can be implemented? Its most basic form is a wireframe.

1. **Wireframes.**

Wireframes are low-fidelity graphic structures of a website or an app. They help map different product pages with sections and interactive elements.

1. **Mock-ups.**

Wireframes can be turned into mock-ups – visual designs that convey the look and feel of the final product.

Eventually, mock-ups can become the final design of the product.

#### **3.3: Categorizing the Requirements**

Categorizing the requriment into functional, nonfunctional and technical requirement simplifies the process of analysis

##### 3.3.1 Functional requirements

Are product features or functions that developers must implement to enable users to accomplish their tasks. So it’s essential to make them clear both for the development team and the stakeholders. Generally, functional requirements describe system behaviour under specific condition

**A User Management**

1. **Register and account creations**

* The app should allow users to create accounts and log in using credentials like email and password or through social media integration.

1. **Authentication**

* Users can log in to their account using their credentials
* Users can manage their profile information (e.g., name, email).

Also the user would be able to

* Capture or upload images of their belongings.
* Search for missing items by capturing a new image or uploading an existing one.
* Users can edit or delete archived object information.
* Users can filter and sort archived objects (e.g., by category, date added).

**B Object archival (keeper)**

The preservation and management of digital representations of physical objects.

Here,

* Users (keeper) can capture images of objects using the device camera.
* Users can upload images of objects from their device gallery.

**Object Retrieval (finder):**

* Users can search for missing objects using a new image captured by the camera.
* Users can search for missing objects by uploading an existing image from the device gallery.
* The search results should display a ranked list of potential matches from the archived database based on image similarity scores.
* Users can view detailed information associated with each matched object in the search results globally

##### 3.3.2Non functional requirements

Are not related to the system's functionality but rather define how the system should perform. They are crucial for ensuring the system's often influencing the overall experience. Gathering of these requirements could be done by talking to people, conducting workshops, and using surveys etc.

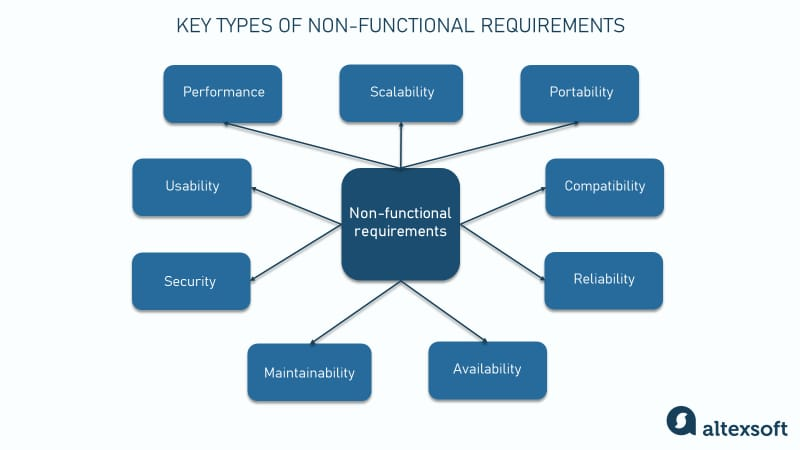


FIG2: Some key non-Functional Requirements

For

**Performance:**

* The application needs to be responsive and perform actions within a reasonable timeframe.

Example,

* + The app should launch quickly on various models of the mobile device.
  + Image capture and upload should be efficient and not take excessive time.
  + Image matching and search results should display within a tolerable wait time.

**Availability:**

* The application needs to be available for use most of the time with an uptime target of about 98% and more.
* Since it is a mobile based application, thus the need for it to function properly even under moderate network conditions (3G/4G).

**Security:**

* User credentials must be securely stored using hashing or encryption to prevent unauthorized access.
* User data (object details, images) must be securely stored in the applications database or on the device (with user consent for device storage) using encryption at rest and in transit.
* The application should implement secure communication protocols (HTTPS) for data transmission.
* The application should be regularly tested for vulnerabilities and patched promptly.

**Usability:**

* The user interface have to be intuitive and user-friendly for people with varying technical skills
* The application should be accessible for users with disabilities (considerations for visual impairments, etc.).
* The application ought to be localized to support different languages (if targeting a global audience).

**Reliability:**

* The application should function consistently and reliably with minimal crashes or errors.
* Also, the application should recover gracefully from unexpected errors and provide informative feedback to users.

**Scalability:**

The application needs to able to handle an increasing number of users and archived objects without performance degradation.

* The underlying infrastructure (image stored in the database,) should be scalable to accommodate future growth.

**Privacy:**

* Users’ needs to have control over their privacy settings and the data associated with archived objects.
* The application should collect and store only the data necessary for its functionality and comply with relevant data privacy regulations.

**Offline Functionality (Optional):**

* The application should allow users to capture images offline and store them locally for upload when an internet connection becomes available.

**Battery Consumption:**

The application needs to be energy-efficient and minimize battery drain on mobile devices.

**Maintainability:**

* The application code should be well-documented, modular, and easy to maintain for future updates and bug fixes.

These non-functional requirements detail the "how” to address the overall qualities and characteristics of your application. By considering these alongside the Functional requirements detailing the "what" (features) you can ensure a well-rounded mobile application for retrieving and archiving missing objects that provides a positive user experience.

##### 3.3.3 Technical requirements

Here's a breakdown of the technical requirements for the mobile application:

**A Mobile App Development:**

1. **Platform:** Choose the target platform(s) for the application (Android, iOS, or both)
2. **Development Framework**: Select a suitable mobile development framework (e.g., React Native, Flutter, native development) that aligns with the chosen platform(s) and developer expertise.
3. **User Interface (UI) Design**: a user-friendly and intuitive interface for efficient user interaction with the application's functionalities.

**B Image Management:**

1. **Image Capture**: The application should utilize the device camera to capture images of objects.
2. **Image Upload**: The application should enable users to upload images of objects from their device gallery.
3. **Image Storage:** A secure storage solution is needed for user-uploaded images. Cloud storage services (e.g., Firebase Storage, AWS S3) or local device storage (with user consent) are options to consider. Security measures like encryption for images at rest and in transit are crucial.
4. **Image Preprocessing (Optional**): Techniques like resizing, cropping, or colour adjustments might be implemented to improve image quality and search performance.

**C Image Recognition and Search:**

1. **Image Recognition Library**: Integrate an image recognition library or API (e.g., TensorFlow Lite, Clarifai) capable of performing image similarity matching. This library should be optimized for mobile device performance.

**2 Image Search Algorithm**: with its high level of Accuracy and advanced features, these algorithms are often more sophisticated and can achieve better image matching results, especially for complex objects or challenging scenarios (e.g., variations in lighting, angles). Thus leading to a higher chance of successfully reuniting lost objects with their owners.

**D Testing:**

**1 Unit Testing:** Test individual software components to ensure they function as intended.

**2 Integration Testing**: Test how different components of the application work together.

**3 Usability Testing:** Involve users in testing the application to identify usability issues and ensure a smooth user experience.

**4 Performance Testing**: Evaluate the application's performance under various load conditions to ensure responsiveness and stability.

By considering these technical requirements, the development of a robust and functional mobile application that leverages image recognition for object archiving and retrieval is possible.

#### **3.4 prioritize the Requirements**

Here prioritization in been level into high, medium and low based on what the system need to offer as without excluding its core functionalities.

* **High priority**: Core functionalities (object archiving, retrieval), performance, usability, security, reliability.
* **Medium priority**: Edit/delete objects, filter/sort, and mark objects as found/lost, offline functionality (optional).
* **Low priority**: Advanced search options, social media integration (considered for future iterations).

**A) High Priority:**

These are essential functionalities that address the core purpose of the application:

**Functional Requirements**

* Object Archiving (capture/upload images, add object details - optional)
* Object Retrieval (search by image, ranked results based on image similarity)

**Non-Functional Requirements:**

* Performance (responsiveness for actions like image capture, upload, search)
* Usability (intuitive UI, clear instructions)
* Security (encryption for data storage and transmission), as Data security is paramount to gain user trust and protect sensitive information
* Reliability (minimal crashes, informative error messages)

**B Medium Priority**:

These features enhance the application's usability and value proposition but can be considered in later development stages:

**Functional Requirements:**

* Edit/Delete archived objects
* Filter/Sort archived objects
* Mark objects as "found"/"lost"
* Offline functionality (optional)

**Non-Functional Requirements:**

* Accessibility features for users with disabilities
* Data export functionality (optional)

**C Low Priority:**

These features might be considered for future iterations based on user feedback and business goals

* Advanced search options (e.g., by object category, brand)
* Social media integration.

By prioritizing requirements effectively, you can ensure your mobile application delivers a valuable user experience

#### **Understanding the Business Needs**

The business needs for a Mobile-Based Archival and Retrieval of Missing Objects Application using image recognition can be multifaceted, depending on the specific business model. Here's a breakdown of some potential considerations:

**A Core Value Proposition**:

1. **Addressing a Pain Point**: The application addresses the common problem of lost or stolen belongings by providing a tool for secure archiving and potential retrieval using image matching.

**B Potential Business Models**:

1. **Freemium Model**: Offer a basic version of the application for free with limited storage or features. Premium tiers could offer increased storage, advanced search functionalities, or cloud backup options.
2. **Subscription Model**: Users pay a monthly or annual subscription fee for full access to the application's features.

**C Monetization Strategies (depending on model):**

* In-app purchases for premium features or storage upgrades.
* Subscription fees for ongoing access.
* Revenue sharing with partners offering the application as a service.

## TECHIQUES USED IN REQUIMENT ANALYSIS

Flowcharts, Gantt Charts and gap analysis are advanced methods that helps along processes of the requirement analysis;

1. **Flowcharts**

Flowcharts depict sequential flow and control logic of a related set of activities. They are useful for all stakeholders of the system. Below is a summarized flow chart of the system.

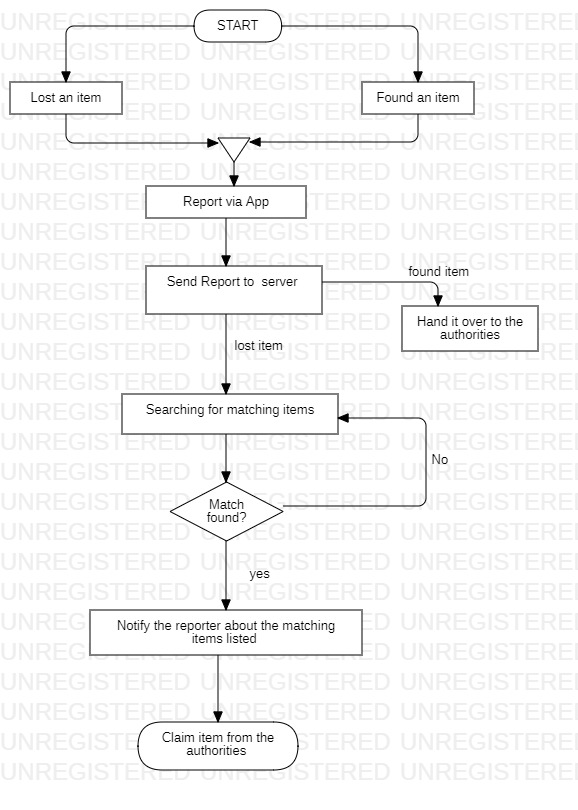


FIG: flow chart showing summarized processes of the system.

1. **Gantt Charts**

Gantt Charts provide a visual representation of tasks along with their scheduled timelines. They help business analysts visualize the start and end dates of all the tasks in a project

## BENEFIT OF REQUIREMENT ANALYSIS

Requirement analysis brings numerous benefits in the development of our system.

Below is a rundown of how beneficial this stage is as it:

1. **Improved User Experience (UX):**

**Focus on User Needs**: Requirement analysis ensures the app caters to the needs of both finders (lost object retrieval) and keepers (archiving found objects). By understanding user goals and pain points, the app can be designed with features and functionalities that truly address their challenges.

**Usability and Efficiency**: Analysing user interactions helps create an intuitive and user-friendly interface. This makes the app easy to navigate for both tech-savvy and non-technical users, streamlining the process of archiving and retrieving objects.

1. **Reduced Development Risks:**

**Clear Project Scope:** Requirement analysis defines the app's functionalities and features in detail. This reduces ambiguity and ensures everyone involved (developers, designers, stakeholders) is on the same page about what the app should achieve. This clear scope minimizes the risk of scope creep (adding unnecessary features) during development.

**Feasibility Assessment:** Identifying technical constraints and limitations early on helps determine the feasibility of implementing certain functionalities. This avoids wasting time and resources on features that might not be technically viable or practical for the mobile platform.

1. **Enhanced System Quality:**

**Error Prevention**: Requirement analysis helps identify potential issues and errors early in the development process. By addressing these issues upfront, you can build a more robust and reliable app with fewer bugs and glitches.

**Testability:** Well-defined requirements serve as the foundation for creating effective test cases. These test cases ensure the app functions as intended and meets all the specified requirements, resulting in a higher quality product.

1. **Stronger Project Management:**

**Prioritization and Estimation:** By analysing requirements, you can prioritize features based on their importance and user impact. This helps development teams focus on critical functionalities first and estimate development effort more accurately.

Overall, requirement analysis is a crucial step in developing a successful mobile app for archival and retrieval of missing objects. It ensures the app is user-centric, addresses real needs, and is built on a solid foundation, leading to a high-quality and effective solution.

## CONCLUSIONS

By effectively analysing, and documenting requirements, a solid groundwork for a successful mobile application is perceived as a clear understanding of stake holders, functional and non-functional requirements. Also with the design of the use case diagram to visualize core functionalities, ultimately leading to a valuable tool for users looking to archive and potentially recover their belongings.

## REFERENCES

[1] T. Alhussain, S. Drew, and O. Alfarraj, Biometric authenti-cation for mobile government security, Intelligent Computing And Intelligent Systems (ICIS) IEEE International Conference,vol.2, pp. 114-118, October 2010.

[2] N. Ghazel and R. Rizal, A preliminary review of interface design elements for mobile electronic government systems, In-formation and Communication Technologies (WICT), Fourth World Congress, pp. 217223, December 2014

[3] S. Chutichudet, T. Kanthathasiri, I. Ritsakunchai, and D. Wong-sawang, LFD: Lost and Found Dog application on mobile,Student Project Conference (ICT-ISPC), 2014. Third ICTInternational, March 2014.

[4] S. Chan, A. Connell, E. Madrid, P. Dongkuk, and R. Kamoua,RFID for personal asset tracking, Systems, Applications andTechnology Conference, 2009. LISAT 09. IEEE Long Island,pp. 1-7, May 2009.