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 2: REQUIREMENT ANALYSIS

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

ABSTRACT

The Mobile-Based Archival and Retrieval of Missing Objects Application aims to revolutionize the way individuals manage and recover their lost belongings. Traditional methods of searching for lost items often prove ineffective and time-consuming. In response to this challenge, our project proposes the development of a mobile application that leverages cutting-edge image matching algorithms to facilitate the archival and retrieval process.

This System Requirements Specification (SRS) document serves as a comprehensive guide to understanding the requirements of the application. It outlines the functional and non-functional requirements, business objectives, stakeholder needs, and prioritization criteria essential for the successful development and implementation of the system.

By prioritizing user experience, operational efficiency, and revenue generation, the application seeks to provide a seamless and intuitive platform for users to upload images of lost items, receive real-time notifications for potential matches, and engage with a supportive community to enhance the search process.

With a clear understanding of the project's objectives and requirements, stakeholders can collaborate effectively to bring the Mobile-Based Archival and Retrieval of Missing Objects Application to fruition, ultimately providing a valuable solution to a common problem faced by individuals worldwide.

Table Of Content

INTRODUCTION

In this report, we delve deep into the requirements and specifications of our Mobile-Based Archival and Retrieval of Missing Objects Application. We explore the diverse needs of stakeholders, prioritize requirements based on their impact and feasibility, and lay out a detailed roadmap for the development and implementation of the system.

OBJECTIVE

Our objective is to systematically analyse and prioritize the requirements gathered for the Mobile-Based Archival and Retrieval of Missing Objects Application. We aim to define clear business goals, identify stakeholders and their needs, analyse requirements from various groups, establish prioritization criteria, categorize requirements based on priority, define comprehensive business requirements, and compile a detailed System Requirements Specification (SRS) document. Through these efforts, we seek to lay a solid foundation for the successful development and deployment of the application, ultimately providing users with an efficient and intuitive solution for retrieving lost items.

BUSINESS GOALS

The primary business goal of the Mobile-Based Archival and Retrieval of Missing Objects Application is to provide a user-friendly, efficient, and reliable platform for individuals to manage and retrieve their lost belongings. By leveraging advanced image matching technology and mobile capabilities, the application aims to streamline the process of locating lost items, reducing frustration and inconvenience for users. Additionally, the application seeks to enhance user engagement and satisfaction through features such as real-time notifications, community collaboration, and seamless navigation. Ultimately, the goal is to establish the application as a trusted solution for lost item retrieval, improving user experience and contributing to overall customer satisfaction.

LIST OF STAKEHOLDERS

- End Users
- Administrators
- Technical Team
- Project Sponsor and supervisor (client)

REQUIREMENTS GATHERED

Summary of Requirements from the user (User Requirements)

Functional.

1. **Upload Images of Lost Items:** Users should be able to upload images of lost items through the mobile application interface.
2. **Search for Potential Matches:** The application should allow users to search for potential matches of lost items within the database using uploaded images.
3. **Utilize Image Matching Algorithms:** The application must incorporate state-of-the-art image matching algorithms to compare uploaded images with archived objects.
4. **Implement Computer Vision Techniques:** Advanced computer vision techniques, such as feature extraction and similarity scoring, should be employed to identify potential matches with high accuracy and efficiency.
5. **Provide Notification System:** Users should receive notifications when potential matches for their lost items are found. Notification preferences, such as push notifications or email notifications, should be customizable.
6. **Ensure Data Privacy and Security:** The application must prioritize data privacy and security, ensuring that users' personal information and uploaded images are protected. Compliance with data protection regulations should be ensured.

7. **Location Tracking:** The app should have the functionality to track the last known location of the lost item, providing users with helpful information for retrieval.

Non-Functional

1. **User-Friendly Interface:** The application should feature an intuitive user interface and seamless navigation to enhance user experience.
2. **Efficiency and Performance:** The system must be efficient and capable of processing image comparisons and search queries quickly to provide timely results.
3. **Reliability and Accuracy:** The application should demonstrate high reliability and accuracy in identifying potential matches for lost items, minimizing false positives and negatives.

Summary of requirement gathered From Administrator

1. **Database Management:**
    - Ability to add, update, and remove archived objects from the database.
    - Features for categorizing and organizing archived items efficiently.
    - Tools for managing metadata associated with archived objects, such as descriptions, tags, and timestamps.
2. **User Management:**
    - Functionality for creating and managing user accounts with different access levels and permissions.
    - Options for assigning roles and responsibilities to users based on their job functions.
    - Tools for monitoring user activity, tracking login/logout sessions, and auditing changes made by users.
3. **System Configuration:**
    - Settings for configuring system preferences, such as notification preferences, search parameters, and default display options.
    - Customization options for tailoring the system interface to meet specific user preferences and workflow requirements.

4. **Monitoring and Maintenance:**
    - Tools for monitoring system performance, including real-time dashboards and reports on system usage, resource utilization, and performance metrics.
    - Features for conducting routine maintenance tasks, such as software updates, data backups, and system diagnostics.
5. **Data Security and Compliance:**
    - Measures for ensuring data security, including encryption protocols, access controls, and authentication mechanisms.
    - Compliance with relevant regulations and standards, such as GDPR, HIPAA, or industry-specific data protection requirements.
6. **User Support:**
    - Resources for providing user support and assistance, including documentation, help desk services, and training materials.

Outcome of the technical team brainstorming session.

Technical requirements

1. **Image Processing Algorithms:**
    - Implementation of feature/image machine
    - Implementation of image recognition algorithms for object detection and feature extraction.
    - Integration of image segmentation techniques to isolate objects of interest within images.
    - Utilization of image enhancement (image processing) methods to improve image quality and clarity.
2. **Database Management:**
    - Design and implementation of a scalable database schema to store archived objects and associated metadata.
    - Integration with a relational database management system NoSQL database for efficient data storage and retrieval.

- Implementation of data indexing and querying mechanisms for fast and accurate retrieval of archived objects.

3. **System Architecture:**
   - Design of a MVC (model, view, controller) architecture coupled with micro service architecture for a modular and scalable development.
4. **Integration with External APIs or Services:**
   - Integration with third-party APIs for accessing external data sources, such as relocation services for mapping lost item locations.
   - Incorporation of APIs for accessing external image databases or repositories for comparison and matching.
   - Integration with communication APIs for sending notifications to users about potential matches or updates on lost items.
5. **Data Security and Privacy:**
   - Implementation of encryption mechanisms to secure sensitive data, such as user information and uploaded images.
   - Implementation of access control measures to restrict unauthorized access to system resources and data.
6. **Scalability and Performance**
   - Design for scalability to accommodate a growing user base and increasing volumes of archived objects.
   - Optimization of algorithms and data processing pipelines for improved performance and response times.

Business Requirements

Enhance Customer Experience

- **Description:** The application should provide a seamless and intuitive user experience to increase customer satisfaction and retention.
- **Rationale:** A positive user experience will lead to higher adoption rates and customer loyalty.

Increase Operational Efficiency

- **Description:** Automate the lost-and-found process to reduce the time and effort required to manage lost items.
- **Rationale:** This will free up resources and reduce operational costs.

Leverage Advanced Technology

- **Description:** Utilize state-of-the-art image matching and computer vision technologies to offer accurate and reliable matching of lost items.
- **Rationale:** Advanced technology will differentiate the application from competitors and provide a unique value proposition.

Ensure Data Security and Compliance

- **Description:** Implement robust data security measures to protect user data and comply with relevant data protection regulations.
- **Rationale:** Ensuring data privacy and security is crucial for building user trust and avoiding legal issues.

Generate Revenue

- **Description:** Explore monetization options such as premium features, in-app advertisements, and partnerships with insurance companies or retailers.
- **Rationale:** Creating revenue streams will help sustain the application financially and contribute to the business's profitability.

Support Scalability and Growth

- **Description:** Design the application architecture to support future scalability and the addition of new features.
- **Rationale:** A scalable solution will accommodate growing user numbers and evolving business needs without significant redevelopment.

Outcome of requirement extraction From Client Problem definition

Functional Requirements:

1. **User Registration and Authentication**
    - Users should be able to create accounts and log in securely to access the application's features.
2. **Image Upload and Storage**
    - Users should be able to capture and upload images of lost items to the application.
    - The application should store uploaded images securely in a centralized database.
3. **Image Matching and Retrieval**

- The application should perform real-time image matching using advanced algorithms to compare uploaded images with archived objects.
- Identified matches should be retrieved and displayed to users in a timely manner.

4. **Intuitive User Interface**

   - The application should feature user-friendly interfaces and seamless navigation to enhance usability.
   - Users should be able to easily navigate through different sections of the application, including image upload, search, and match results.

5. **Search Functionality**

   - Users should be able to search for potential matches of their lost items using various search criteria, such as keywords or visual similarities.
   - The search functionality should be fast, accurate, and intuitive for users to use effectively.

6. **Notification Functionality**

   - Users should receive real-time notifications via push notifications or email when potential matches for their lost items are found.

7. **Crowdsourcing and Community Engagement**

   - Users should have the option to share information about their lost items on social media platforms and enlist the help of friends, family, and other users in the search process.

Non-Functional Requirements:

1. **Security and Privacy**

   - The application should implement robust security measures to protect user data and privacy.
   - Measures should include encryption of data transmission, secure storage of user information, and authentication mechanisms.

2. **Performance**

   - The application should be responsive and per formant, even under high load conditions.

- Image matching algorithms should be optimized for speed and efficiency to ensure quick retrieval of matches.

3. **Scalability**

   - The application should be designed to handle a growing user base and increasing volumes of uploaded images.
   - Scalability measures should include efficient database management, resource allocation, and load balancing.

4. **Reliability**

   - The application should be reliable and available for users to access whenever needed.
   - Measures should be in place to mitigate downtime, such as redundant systems and regular maintenance.

5. **Compatibility**

   - The application should be compatible with a wide range of mobile devices and operating systems.
   - Compatibility testing should be conducted to ensure consistent performance across different platforms.

REQUIREMENT PRIORITIZATION

Requirement prioritization is a crucial step in the development process of any project, ensuring that resources are allocated effectively to meet the most important needs and objectives. In this section, we will delve into the criteria and methods used to prioritize the requirements gathered for the Mobile-Based Archival and Retrieval of Missing Objects Application. By establishing clear guidelines for prioritization, we aim to focus our efforts on addressing the most critical and impactful requirements first, thereby maximizing the success and efficiency of the development process.

Prioritization Criteria.

User Impact: We prioritize requirements that directly impact the user experience, such as core functionalities related to uploading lost item images, searching for matches, and receiving notifications. Features that enhance usability, accessibility, and overall satisfaction should be given high priority.

Business Value: We Evaluated requirements based on their potential to generate value for the business, such as revenue generation, cost reduction, or competitive advantage. Features that align with strategic objectives and contribute to the application's market viability should be prioritized accordingly.

Technical Feasibility: We considered the technical complexity and feasibility of implementing each requirement within the project timeline and resource constraints. Requirements that can be easily implemented with existing technology and resources should be prioritized over those that require significant development effort or specialized expertise.

Time Sensitivity: Assess the urgency of each requirement in terms of project timelines, deadlines, and stakeholder expectations. Requirements that are time-sensitive or critical for meeting project milestones should be prioritized to ensure timely delivery and project success.

Risk Mitigation: Prioritize requirements that address critical risks or dependencies, such as security vulnerabilities, regulatory compliance, or performance limitations. Features that mitigate potential risks and ensure the stability, reliability, and security of the application should be given high priority.

Stakeholder Input: Consider feedback and input from stakeholders, including users, administrators, technical teams, and project sponsors. Requirements that align with stakeholder preferences, expectations, and strategic objectives should be prioritized to ensure stakeholder satisfaction and engagement.

Prioritization criteria on the requirements collected from Stakeholders

Functional Requirements:

1. **Upload Images of Lost Items**

    - Importance: High
    - Justification: This requirement directly impacts the user experience by enabling users to report lost items, aligning with the "User Impact" criterion. It also

enhances operational efficiency by automating the process of item submission, contributing to "Increase Operational Efficiency."

2. **Search for Potential Matches**

   - Importance: High
   - Justification: Crucial for users to find potential matches efficiently, enhancing the utility of the application ("User Impact"). Additionally, efficient search functionality contributes to the application's reliability and accuracy ("Reliability and Accuracy").

3. **Utilize Image Matching Algorithms**

   - Importance: High
   - Justification: Integral to the core functionality of matching uploaded images with archived objects, ensuring the effectiveness of the application ("User Impact"). It also leverages advanced technology, aligning with "Leverage Advanced Technology."

4. **Implement Computer Vision Techniques**

   - Importance: High
   - Justification: Advanced computer vision techniques are necessary for accurate matching of lost items, enhancing the reliability of the application ("Reliability and Accuracy"). Additionally, it aligns with "Leverage Advanced Technology."

5. **Provide Notification System**

   - Importance: Medium.
   - Justification: While important for user engagement, this feature can be implemented after the core functionalities are in place, contributing to "Time Sensitivity." It also enhances user experience and engagement, aligning with "User Impact."

6. **Location Tracking**

   - Importance: Medium
   - Justification: Helpful for users in locating their lost items, but not as critical as other functionalities. It enhances the user experience and provides additional value ("User Impact").

7. **Crowdsourcing and Community Engagement**
    - Importance: Medium
    - Justification: While beneficial for enhancing user engagement and expanding the application's reach, this functionality may not be immediately critical for the core functionality of matching lost items.
8. **User Support**
    - Importance: Medium
    - Justification: Important for providing assistance to users and ensuring a positive user experience, but may not be as urgent as other functionalities.
9. Register **and Login**
    - Importance: High
    - Justification: Fundamental for user authentication and access control, essential for ensuring the security and privacy of user data.
10. **Database Management**
    - Importance: High
    - Justification: Crucial for managing archived objects and associated metadata efficiently, ensuring the integrity and accessibility of the data.

11. User Management

- Importance: High
- Justification: Essential for creating and managing user accounts, assigning roles and permissions, and monitoring user activity to maintain security and accountability.

12. System Configuration

- Importance: Medium
- Justification: Important for customizing system preferences and user interface settings, but may not be as critical as core functionalities.

13. **Monitoring and Maintenance**
    - Importance: Medium
    - Justification: Necessary for ensuring system reliability, performance, and security through routine maintenance tasks, but may not be as urgent as other (high) functionalities.

Non-Functional Requirements

Ensure Data Privacy and Security

- Importance: High
- Justification: Critical for user trust and compliance with data protection regulations, ensuring the privacy and security of user data ("Risk Mitigation"). It also aligns with "Business Value" by protecting the business from legal and reputational risks.

User-Friendly Interface

- Importance: High
- Justification: An intuitive interface is crucial for user adoption and satisfaction, enhancing the overall user experience ("User Impact"). It also contributes to "Business Value" by increasing customer satisfaction and retention.

Efficiency **and Performance**

- Importance: High
- Justification: The system must be efficient and responsive to provide timely results, ensuring user satisfaction and usability ("User Impact"). Additionally, it aligns with "Business Value" by improving operational efficiency and reducing rework.

Reliability **and Accuracy**

- Importance: High
- Justification: Essential for providing accurate matches and reliable service to users, minimizing false positives and negatives ("User Impact"). It also aligns with "Risk Mitigation" by reducing the likelihood of errors and system failures.

Prioritizing Business Requirements:

1. **Enhance Customer Experience**

    - **Importance: High**
    - **Justification:** Critical for increasing user satisfaction and retention, contributing to the long-term success and growth of the application

2. **Increase Operational Efficiency**

    - **Importance: Medium**

- **Justification:** Important for streamlining processes and reducing operational costs, although it may not directly impact user experience or value.

3. **Leverage Advanced Technology**

   - **Importance: High**
   - **Justification:** Essential for staying competitive and providing innovative solutions to users, enhancing the application's value proposition and differentiation

4. **Ensure Data Security and Compliance**

   - **Importance: High**
   - **Justification:** Critical for building trust with users and complying with legal and regulatory requirements, ensuring the protection and privacy of user data

5. **Generate Revenue**
   - **Importance: Medium**
   - **Justification:** Important for sustaining the application financially, although it may not be immediately critical for the initial development phase.

6. **Support Scalability and Growth**

   - **Importance: High**
   - **Justification:** Essential for accommodating future expansion and increasing user numbers, ensuring the long-term viability and scalability of the application.

Requirement Categorization

High Priority:

Functional Requirements:

1. Upload Images of Lost Items
2. Search for Potential Matches
3. Utilize Image Matching Algorithms
4. User Support.
5. Implement Computer Vision Techniques
6. Register and Login
7. Database Management

8. User Management

Non-Functional Requirements:

1. Ensure Data Privacy and Security
2. User-Friendly Interface
3. Efficiency and Performance
4. Reliability and Accuracy

Business Requirements:

1. Enhance Customer Experience
2. Leverage Advanced Technology
3. Ensure Data Security and Compliance
4. Support Scalability and Growth

Medium Priority:

Functional Requirements:

1. Crowdsourcing and Community Engagement.
2. System Configuration.
3. Monitoring and Maintenance.
4. Provide Notification System.

Business Requirements:

1. Increase Operational Efficiency
2. Generate Revenue

DETAILED REQUIREMENT SPECIFICATION DOCUMENT

Goal Of The Implementation

The goal of this system is to develop a comprehensive and efficient Lost and Found mobile application that facilitates the process of reporting and retrieving lost items. This application aims to provide users with a user-friendly platform for uploading images of lost items, searching for potential matches, and receiving notifications about found items. Additionally, the system aims to utilize advanced image matching algorithms and computer vision techniques to enhance the accuracy and reliability of matching lost items. The overarching goal is to enhance user satisfaction by streamlining the lost and found process, leveraging technology to improve efficiency, and ensuring the security and privacy of user data.

Functional Requirements:

R1: Login

Input: User enters their username/email and password.

Process: The system validates the entered credentials against the stored user data. If the credentials are correct, the system generates an authentication token. If the credentials are incorrect, the system returns an error message.

Output: On successful login, the user is granted access to their profile and relevant features. On failed login, the user receives an error message indicating invalid credentials.

R2: Register

Input: User provides details such as name, email, password, and optionally other personal information (e.g., phone number, address).

Process: The system checks the provided email for uniqueness. If the email is unique, the system encrypts the password and stores the user details in the database. If the email already exists, the system returns an error message.

Output: On successful registration, the user receives a confirmation message. On failed registration, the user receives an error message indicating the issue (e.g., email already in use).

R3: Update Profile

Input: User enters updated personal information (e.g., name, email, phone number, address) and possibly a new password.

Process: The system validates the new information. The system updates the user's details in the database. If updating the email, the system checks for uniqueness before updating. If updating the password, the system encrypts the new password before storing it.

Output: The user receives a confirmation message indicating that their profile has been successfully updated. The system reflects the updated information in the user's profile.

4. View Notification

Input: User accesses the notification section.

Process: The system retrieves the user's notifications from the database. Notifications are filtered by read/unread status and sorted by date.

Output: The user sees a list of notifications, including details such as message content, date, and read/unread status. The user can mark notifications as read, delete them, or take actions suggested by the notifications (e.g., follow a link).

Reporting Lost Items

R4: Upload image into the system

Description: This requirement entails providing users with the functionality to upload images of lost items into the system. Users should be able to capture or select images from their device's gallery and upload them through the application interface. The user has two options directly provided to him: capture with camera and selecting from gallery. Once the image is selected, the user should initiate the upload process by clicking on the upload button, transferring the image data securely to the server for processing and storage. The upload process should include error handling mechanisms to address any issues such as network connectivity issues or file format compatibility problems. Additionally, users should receive feedback on the upload status, including confirmation messages upon successful upload and error notifications in case of failures.

R4-1: Capturing image with camera

Input: click camera icon Input: User selects the option to capture an image using the device's camera.

Process: The system activates the camera interface on the user's device, allowing them to frame the desired subject and capture the image. After capturing the image, the system processes it to ensure image quality and compatibility with the application's requirements. This may involve resizing, compression, or format conversion.

Output: The captured image is displayed on the screen for user review. If satisfied, the user can proceed with uploading the image to the system. If not, they may have the option to retake the image before proceeding. Additionally, the system may provide feedback on the image quality or file size to guide the user's decision.

R4-2: Uploading image to the system

Input: Image and Additional data

Process: The system attaches metadata such as the upload timestamp, user ID, and any additional user-provided details to the image. The image and metadata are securely transmitted to the server using encryption protocols. Upon receipt, the server stores the image and metadata in the appropriate database, updating the database index for quick retrieval. The system then verifies the successful upload and storage of the image and metadata.

Output: The user receives a confirmation message indicating the successful upload of the image and details. The uploaded image and associated details become accessible in the user's profile or upload history, where the user can view, edit, or delete the uploaded image and details if necessary.

Search for Potential Matches

R4: Search for Potential Matches

Description: The "Search for Potential Matches" function allows users to find possible matches for their lost items within the system's database. Users can initiate a search by submitting images or specifying certain criteria. The system employs advanced image matching algorithms and computer vision techniques to compare the submitted images with those stored in the database. Upon identifying potential matches, the system presents the results to the user, enabling them to review and verify the matches. This function aims to streamline the process of locating lost items, enhancing user satisfaction and operational efficiency.

R5-1: select or capture the item

Input: User selects an existing image or captures a new image of the lost item.

Process: If capturing a new image, the system activates the device camera. The user captures or selects the image.

Output: The selected or captured image is displayed and ready for submission to the search process.

R5-2: provide Additional information

Input: additional info

Process: enter the info in the text fields

Output: none

R5-3: Search for potential match

Input: Image and additional information (e.g., item description, last known location).

Process: The system uses advanced image matching algorithms to compare the uploaded image with images in the database. Computer vision techniques, such as feature extraction and similarity scoring, are employed to identify potential matches. The system cross-references additional information to refine the search results.

Output: A list of potential matches is generated and displayed to the user, along with details of each match for further review.

R5-3: Verifying Object ownership.

Input: check the checkbox below the image that matches your item

Process: A notification is send to the admin with the claimed image additional info entered when uploading the found object and those entered by the user for the search. The status of the object in updated to ‘pending claim’. When these two information match, the admin sends a notification to the user with a code that he will present to collect the object. If not further verification will be require that may include prove of identity and ownership. Finally the object is added to the user claimed objects history and the status is updated if necessary.

Output: user receive an message confirming ownership or requesting for prove

R6: Track Matched Items on the Map

Input: User selects the option to track matched items on the map. The system retrieves the location data associated with the matched items.

Process: The system utilizes geolocation data stored in the database for each matched item. It then plots the locations of the matched items on the map interface. The system may employ mapping APIs or services to generate the interactive map. Users may zoom in/out and pan across the map to explore the locations of matched items.

Output: The user is presented with an interactive map displaying the locations of matched items. Each matched item is represented by a marker or icon on the map. Users can click on individual markers to view more details about the matched item, such as description and status.

R7: User Support

Description: The User Support function provides assistance to users regarding the various features and functionalities of the system. This includes offering help with user registration, image upload, search operations, and notifications. Users can access support through multiple channels such as in-app help, FAQs, and customer service contact options. The goal is to ensure a positive user experience by addressing user queries and issues promptly and effectively, enhancing overall user satisfaction and engagement with the application.

R7-1: Access Help Section

Input: Click help related button

Description: The user accesses the help section of the application for assistance. Input: Selection of the help option Output: Help resources and support contact information displayed on the application interface.

Output: interface displayed

R7-2: Submit Support Request.

Input: User fills out a support request form including details such as their name, contact information, the issue they are facing, and any relevant screenshots or images.

Description: The system prompts the user to provide necessary details about their issue through a structured support request form. The user submits the form, and the data is securely transmitted to the support team. The system automatically categorizes the request based on the provided details and assigns it a priority level. The support request is logged into the support ticketing system, where it is assigned to a support agent for resolution.

Output: conversation with customer support lunched.

Non-functional Requirements

- Ensure Data Privacy and Security.
- User-Friendly Interface.
- Efficiency and Performance.
- Reliability and Accuracy.

Business Model

The Advertising Model and Freemium Model align well with the context of Cameroon and offer a flexible approach to revenue generation:

1. **Advertising Model**: Given the diverse range of businesses and service providers in Cameroon, targeted advertisements within the application can cater to local businesses, insurance companies, and relevant service providers. For example, local shops,

transportation services, and insurance companies could advertise their services to users searching for lost items, providing them with relevant solutions and options.

2. **Freemium Model**: Offering basic functionalities for free ensures wider accessibility and adoption among users in Cameroon. Basic features such as uploading images of lost items, searching for potential matches, and receiving notifications can be provided at no cost, encouraging users to engage with the platform. Premium features, such as advanced search filters, priority customer support, and additional security measures, can be offered through subscription plans, providing a source of revenue while maintaining a free-to-use option for users.

Combining these models allows for both immediate revenue generation through advertising and potential future revenue streams through premium subscriptions and commission-based transactions. As the platform grows and establishes itself in the market, exploring additional revenue models such as commissions can further enhance profitability and sustainability.

CONCLUSION

In conclusion, the System Requirements Specification (SRS) document serves as a comprehensive blueprint for the development of the lost-and-found application. By meticulously outlining the functional and non-functional requirements, prioritizing key features, and proposing a viable business model, this document lays the foundation for a robust and user-centric solution. The detailed analysis of stakeholder requirements, coupled with the prioritization criteria, ensures that development efforts are focused on delivering maximum value to users while addressing critical business objectives. Furthermore, the inclusion of implementation goals provides strategic guidance for future enhancements and scalability, fostering continuous improvement and innovation. Overall, the SRS document serves as a guiding framework to steer the development process, aligning technical capabilities with user needs and business goals to create a compelling and sustainable solution for the lost-and-found ecosystem.

GLOSSARY

- **Lost-and-Found Application**: A software solution designed to facilitate the reporting, matching, and retrieval of lost items through user-contributed images and database search functionalities.
- **SRS (System Requirements Specification)**: A document that specifies the functional and non-functional requirements of a system, including user interactions, data processing, security measures, and performance criteria.
- **Stakeholder**: Any individual or group with an interest in the development or outcome of the lost-and-found application, including users, administrators, developers, and business stakeholders.
- **Use Case**: A description of interactions between a user (actor) and the system to achieve a specific goal or functionality.
- **Entity-Relationship Diagram (ERD)**: A visual representation of the entities (objects) in a database and the relationships between them, used to design and understand the structure of the database.
- **Mockup**: A static representation or blueprint of the user interface design, typically used to visualize layout, navigation, and content placement.
- **Wireframe**: A basic visual guide that represents the skeletal framework of a website or application, outlining the arrangement of elements without final design details.

REFERENCE

https://www.lambdatest.com/learning-hub/requirement-analysis

https://www.geeksforgeeks.org/software-requirement-specification-srs-format/