THE UNIVERSITY OF BUEA

P.O Box 63,

Buea,South West Region

Cameroon

Tel: (237) 674354327

Fax: (237) 3332 22 72

REPUBLIC OF CAMEROON

PEACE-WORK-FATHERLAND

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT :Computer Engineering

REQUIREMENT ANALYSIS OF A MOBILE-BASED ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS USING IMAGE MATCHING

PRESENTED BY

NAME	MATRICULE
EKUNDIME GLEAN MAKOGE	FE21A433
DJOUMESSI LEKANE WENDY FORTUNE	FE21A173
INDAH RISCOBELLE MBAH	FE21A204
KEDJU PRECIOUS NGWE LEKUNZE	FE21A211
NGUEDIA JEATSA JOYCE GRACE	FE21A263

Course Supervisor   **Dr NKAMENI VALERY**

Academic year: 2023/2024

Table of Contents

1. INTRODUCTION AND DESCRIPTION

1.1 Definition of mobile-based archival and retrieval of missing objects(documents) application.

After thorough analysis of the requirements and user needs for our mobile app designed for archival and retrieval of missing objects, it has become evident that a significant portion of our users primarily seek assistance in archiving and retrieving important documents such as ID cards, passports, and licenses. Considering the high demand and critical nature of document retrieval for our user base, we have made the strategic decision to focus our development efforts on enhancing the archival and retrieval functionalities specifically tailored to documents. By prioritizing this aspect of our app, we aim to streamline the process of storing and retrieving vital documents, thereby addressing a crucial pain point for our users and providing them with a more efficiently. This decision aligns with our commitment to meeting the specific needs of our users while ensuring the success and effectiveness of our mobile app in facilitating the retrieval of missing objects.

1.2 General overview on how the app will function

In our fast-paced world, losing track of personal documents is an all-too-common occurrence. These incidents can disrupt our daily lives and cause unnecessary stress. To address this issue, we introduce a revolutionary mobile-based archival and retrieval system designed to help users locate their missing documents quickly and efficiently through image matching technology.

Mobile-based archival and retrieval system leverages the power of image recognition and matching algorithms to provide users with a seamless experience in locating their lost items. The application utilizes the camera functionality of smartphones to capture images of documents that users wish to keep track of. These images are then securely stored in the app's database, along with relevant metadata such as timestamps and descriptions.

When a user realizes that they have misplaced a document, they can simply open the app and initiate a search by either describing the lost object or uploading a reference image. The application's advanced image matching technology quickly scans through the database to identify potential matches based on visual similarities.

Once a match is found, the user receives instant notifications, along with details about the location where the matching image was captured. This feature is particularly useful for retracing steps and pinpointing the last known whereabouts of the missing item. Additionally, users can opt to receive alerts when their lost objects are detected by other users who have the app installed, creating a network of assistance for locating misplaced belongings.

Privacy and security are paramount in All user data, including images and personal information, are encrypted and stored securely on servers with stringent access controls.

Users have full control over their data and can choose to delete or update information at any time.

With the APP the frustration and inconvenience of losing personal items are minimized, allowing users to regain peace of mind and focus on what matters most in their lives.

1.3 Objective and Goal

1.3.1 Objective

The primary objective of our archival and retrieval system for missing documents is to empower users with a secure and efficient platform to safeguard their important identification and personal paperwork. By offering a reliable tool for the archival and retrieval of missing documents, our aim is to alleviate the stress and inconvenience associated with lost or misplaced IDs, passports, licenses, and other essential documents. Through seamless organization, quick access, and dependable retrieval features, our solution strives to provide users with peace of mind, ensuring they can easily locate and restore their vital documents whenever needed.

1.3.2 Goal

Our goal is to develop a mobile application that simplifies the process of archiving, organizing, and retrieving essential documents, including IDs, passports, and licenses. By offering a secure and user-friendly platform.

2. STAKEHOLDERS ANALYSIS

Stakeholders are of two types, internal stakeholders and external stakeholders. Internal stakeholders have a direct relationship with the project and are invested in its success. They are typically involved in the day-to-day operations and decision-making processes.

2.1 Internal stakeholders
- **Project Team:**
  This includes developers, designers, user experience (UX) specialists,QA engineer and project managers.
  The project team is responsible for developing, maintaining and ensuring that the mobile app meet the need of the end users.

- o **End users:**

  - **Primary Users:**

    Individuals who have lost objects (documents).

  - **Secondary Users:**

    Public who might find lost objects.

3. ANALYSIS OF COLLECTED REQUIREMENTS

The data collected is being analysed based on the goal of the project.

3.1. Requirement Analysis using the prioritisation process.

The analysation process was done using the MoSCoW method as seen below.

Must-Have (MUST):

- ➢ **Core functionalities:**
  - o Ability to report a missing object (document) with details and photos.
  - o Search for potentially matching objects based on text, image, category, and location.
  - o Contact the reporting party of a potential match.(ability to connect with finder)
  - o Secure user authentication and data storage (object archiving).
  - o Performance: Fast loading times for core functionalities.

Should-Have (SHOULD):

- ➢ **Additional features**:

  - o User accounts for managing reports and retrieval history.
  - o Push notifications for updates on reported objects.
  - o Backup Functionality
  - o Social media integration
  - o Reward system.
  - o Multi-lingual support
  - o Location tracking
  - o Feedback features

Could-Have (COULD):

- Offline functionality for basic reporting (limited data entry).
- Advanced search filters (e.g., date range, specific keywords).

Won't-Have (WON'T):

- Potential integration with existing cloud storage services for document management.

4. SYSTEM REQUIREMENT SPECIFICATIONS

4.1. Functional Requirements

- **User Registration and Authentication:**
  o Users can create accounts
  o login securely.
  o A robust authentication process for security of user account and data.
  o Modify the profile informations.

- **Object Archiving:**
  o Users can capture and upload images of their belongings through the mobile app which they want to keep track of.
  o Users should have the option to tag items with keywords or labels for easier organization and retrieval.

- **Image Matching and Recognition:**
  o The application should utilize image matching and recognition technology to identify and match uploaded images with archived items.
  o It should be capable of recognizing specific features or patterns to accurately match missing objects with archived images.

- **Search and Retrieval:**

- o Users can search for missing objects by browsing through their archived items or using search filters.
- o Users can search the database by image using the uploaded photo
- o Texted based search should be possible allowing users to search by keywords related to the item description.

- **Location Tracking:**
- o The application should support location tracking to record the last known whereabouts of archived items.
- o Users can view the location history of items and receive notifications if an item's location changes significantly.

- **Reporting Lost Items:**
- o Users should have the ability to report items as lost or stolen within the app.
- o The app should prompt users to provide relevant details about the missing item, such as its description, location, and any identifying features.

- **Alerts and Notifications:**
- o The application should send alerts and notifications to users when a missing object is found or when there are updates on the search status.
- o Users should be able to receive notifications through the app.

- **Integration with External Systems:**
- o Integration with social media platforms could enable users to share information about missing items with a broader audience.

- **User Feedback and Ratings:**
- o Users should be able to provide feedback on the search and retrieval process, including the accuracy of matches and the effectiveness of notifications.
- o Users should be able to rate the app and provide suggestions for improvement.

- **Data Management and Privacy:**
- o Users can have control over their data, including the ability to delete archived items or adjust privacy settings.

- **Contact Information**
- o users can contact the finder of a potential match (secure messaging system or masked contact details)

4.2. Non-functional requirements

4.2.1 Non-Functional Requirements related to Design

Performance:

- The system should have low latency in processing image queries and returning results to users.
- It should be capable of handling a large number of concurrent users without significant degradation in performance.
- The image matching algorithm should be optimized for speed and efficiency to provide real-time results.

Scalability:

- The system should be scalable to accommodate an increasing number of users and a growing database of archived images.
- It should be able to scale horizontally by adding more servers or cloud resources to handle increased load.
- The app should support adding metadata such as item name, description, category, and any other relevant information.

Reliability:

- The system should have high availability, with minimal downtime for maintenance or updates.
- It should be resilient to failures, with mechanisms in place for automatic failover and recovery in case of server or network issues.
- Data integrity and consistency should be maintained, ensuring that archived images are not lost or corrupted.

Security:

- The system should employ strong encryption mechanisms to protect user data, including images and personal information, both in transit and at rest.
- Access controls should be implemented to restrict unauthorized access to sensitive data and functionalities.
- User authentication and authorization mechanisms should be in place to ensure that only authorized users can access the system and perform specific actions.
- The application should adhere to data privacy regulations and ensure the secure storage and handling of user data.

Usability:

- The user interface should be intuitive and easy to navigate, catering to users of all levels of technical expertise.
- The system should support multiple languages and provide accessibility features to accommodate users with disabilities.
- Feedback mechanisms should be implemented to keep users informed about the status of their actions, such as uploading images or searching for missing objects.

Compatibility:

- The system should be compatible with a wide range of mobile devices, operating systems, and screen sizes to ensure broad accessibility.
- It should integrate seamlessly with other mobile applications and services, such as cloud storage platforms for backing up archived images.

Maintainability:

- The system should be modular and well-documented, allowing for easy maintenance and future enhancements.
- Codebase should adhere to coding standards and best practices to facilitate collaboration among developers and ensure code quality.

4.2.2. Non-Functional Requirements related to Architecture.

Scalability:

- **Database Scalability:**

  The backend database needs to handle a growing number of users and images efficiently. Consider using a distributed database or cloud storage solution to accommodate increasing data volume.

- **Image Processing Scalability:**

  The image processing pipeline, including feature extraction and matching algorithms, should be scalable to handle a high volume of image uploads without compromising performance. Explore techniques like distributed processing or leveraging cloud-based image recognition services.

Performance:

- **API Response Times:**

The APIs responsible for communication between the mobile app and the backend server should have low latency to ensure a responsive user experience. Optimize code and consider caching mechanisms for frequently accessed data.

- **Image Delivery:**

The system should efficiently deliver images (both lost and found) to mobile devices without significant loading times. Utilize Content Delivery Networks (CDNs) to geographically distribute image storage and improve delivery speed.

Availability:

- **Server Redundancy:**

 Implement redundant servers to ensure uninterrupted service in case of server failure. This ensures the app remains accessible for reporting and finding lost items.

- **Disaster Recovery:**

Establish a disaster recovery plan to minimize downtime in case of unforeseen circumstances like natural disasters or cyberattacks.

Security:

- **Data Encryption:**

 Enforce data encryption at rest and in transit to protect sensitive user information, including descriptions, pictures, and potentially location data.

- **Access Control:**

Implement robust access control mechanisms to restrict unauthorized access to user data and system functionalities within the architecture.

Maintainability:

- **Modular Architecture:**

Design the architecture with modular components (mobile app frontend, backend server, database) to facilitate easier maintenance, upgrades, and independent scaling of individual components.

- **Monitoring and Logging:**

Implement comprehensive monitoring and logging systems to track system performance, identify potential issues, and troubleshoot problems efficiently.

4.2.3 Non-Functional Requirements related to Technology

Frontend

- **Framework:**

  React Native: A popular choice for building cross-platform mobile apps using JavaScript and React components. It offers native-like performance and a rich ecosystem of libraries.

- **Libraries:**
  - **State Management library (Redux):** Manages application state efficiently, especially for complex data flows.
  - **Navigation libraries (React Navigation):** Handle navigation between different screens within the app.
  - **Image processing libraries (Glide):** Optimize image loading and display within the app.

Backend:

- **Languages**

  Node.js with Express: A popular and versatile choice for building server-side APIs in JavaScript. It provides a familiar environment for React Native developers and offers a vast ecosystem of libraries

- **Image Processing Libraries**

  TensorFlow.js: A JavaScript port of TensorFlow, enabling you to build and run image recognition models directly in your Node.js backend

- **Cloud Infrastructure**

Cloud Platform (AWS): It is a platform that offers services like cloud storage, cloud functions, databases, and authentication services.

Database:

- MongoDB: A flexible NoSQL database suitable for storing diverse data structures and high performance use cases.

Authentication:

- Mongodb Authentication: A convenient option for user authentication and authorization within your mobile app.
- Custom Authentication: Implement your own authentication system if you have specific requirements.

Push Notifications:

- Mongodb Messaging: Send real-time notifications to users about updates on reported objects.
- Cloud Functions: Build custom logic for handling push notification triggers within your backend

4.2.4. Non-Functional Requirements related to Business

Monetization Strategy:

- **Freemium Model:**

  Offer basic features for free and charge for premium features like priority matching or location-based search functionalities.

- **Subscription Model:**

  Provide different subscription tiers with varying levels of access, such as unlimited lost item reports or extended matching durations.

- **Partnerships:**

Collaborate with businesses or organizations frequented by your target audience. They could sponsor the app or offer incentives to users who find lost items on their premises.

Marketing and User Acquisition:

- **App Store Optimization (ASO):**

  Optimize the app listing with relevant keywords to improve discoverability in app stores.

- **Social Media Marketing:**

  Utilize social media platforms to promote the app and its benefits to potential users.

- **Partnerships:**

  Collaborate with relevant organizations or influencers to reach a wider audience.

Data Management and Analytics:

- **Data Privacy Compliance:**

  Ensure compliance with all relevant data privacy regulations like GDPR and CCPA to protect user information.

- **Data Analytics:**

  Develop a strategy to collect and analyze user data. This can provide insights into user behavior, identify areas for improvement, and inform future marketing efforts.

Scalability and Sustainability:

- **User Base Growth:**

  The system should be scalable to accommodate a growing number of users and data volume without compromising performance.

- **Long-Term Sustainability:**

  Develop a business model that ensures the app's long-term financial viability. This involve a combination of the monetization strategies mentioned earlier.

Community Building:

- **User Reviews and Feedback:**

  Implement mechanisms to collect user feedback and reviews. This allows you to identify areas for improvement and enhance the overall user experience.

- **Community Features:**

  Consider features that encourage user interaction and foster a sense of community. This could include success stories of reunited objects or a forum for users to share tips on preventing loss.

5. STANDARD

There aren't universally enforced standards for mobile-based archival and retrieval applications using image matching, there are established best practices and relevant guidelines to consider during development. Here's a breakdown of some key areas:

Image Processing and Matching:

- **Image Preprocessing:**

  Standardize image pre-processing techniques to ensure consistency and improve matching accuracy. This might involve resizing, color normalization, or noise reduction.

- **Feature Extraction:**

  Consider leveraging established feature extraction algorithms like SIFT (Scale-Invariant Feature Transform) or SURF (Speeded Up Robust Features) for their robustness and efficiency.

- **Matching Algorithms:**

  Explore well-established matching algorithms like nearest neighbor search or FLANN (Fast Library for Approximate Nearest Neighbors) to efficiently find similar images within the database.

Data Security and Privacy:

- **Data Encryption:**

Adhere to industry standards like AES (Advanced Encryption Standard) for encrypting user data, including descriptions and image files, both at rest and in transit.

- **Data Privacy Regulations:** Ensure compliance with relevant data privacy regulations like GDPR (General Data Protection Regulation) and CCPA (California Consumer Privacy Act) governing data collection, storage, and user consent.

Usability and Accessibility:

- **WCAG Guidelines:**

  Follow the Web Content Accessibility Guidelines (WCAG) to ensure your app is accessible to users with disabilities. This includes features like screen reader compatibility, adjustable text size, and clear labeling.

- **Intuitive Interface:**

  Strive for a user-friendly and intuitive interface with clear instructions and well-defined functionalities. Users should be able to report lost objects, upload pictures, and browse potential matches effortlessly.

Performance and Scalability:

- **Image Optimization:**

  Optimize image formats and sizes to ensure fast loading times without compromising image quality.

- **Cloud Infrastructure:**

  Consider leveraging cloud-based platforms like AWS, Google Cloud Platform, or Azure to benefit from their scalability and reliability. This allows the app to handle a growing user base and data volume efficiently.

Testing and Maintenance:

- **Thorough Testing:**

  Implement a comprehensive testing strategy that covers functionalities, image matching accuracy, performance on various devices, and security vulnerabilities.

- **Regular Updates:**

  Maintain the app with regular updates to address bugs, improve features, and incorporate new technologies as they emerge.

Industry Best Practices:

Review Existing Apps:

6. CONCLUSION

This technology offers several advantages over traditional lost and found methods. Image recognition eliminates the need for lengthy written descriptions, leading to faster and more accurate matching. The mobile platform allows anyone to report or find lost objects, creating a wider search network and increasing the chances of successful retrieval.

However, for this technology to reach its full potential, careful consideration must be given to both functional and non-functional requirements. From user-friendly interfaces to robust image matching algorithms and secure data storage, every aspect needs to be meticulously designed and implemented.

7. REFERENCES

https://apus.libguides.com/apusuahome/blog/Preserving-Your-Digital-Files

https://www.researchgate.net/publication/338371560_Object_Retrieval_in_Microsc

https://cloud.google.com/blog/topics/solutions-how-tos/best-practices-for-architecting-google-cloud-workloads

https://webstandards.ca.gov/usability/

https://www.business2community.com/mobile-apps/market-small-business-mobile-app-01452888

https://themanifest.com/

https://www.cpomagazine.com/data-privacy/

http://go.appannie.com/HowtoTrackYourOwnAppsMetrics.html

https://community.make.com/t/make-community-mobile-app/15821

https://ieeexplore.ieee.org/document/4587710

https://oag.ca.gov/privacy/ccpa

https://gdpr.eu/

https://aws.amazon.com/
https://cloud.google.com/