

FACULTY OF ENGENEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGENEERING

## CEF440: INTERNET PROGRAMMING AND MOBILE PROGRAMMING

**TASK 2: REQUIREMENT GATHERING REPORT**

|  |  |
| --- | --- |
| **NAMES** | **MATRICULES** |
| NSEM CONFIDENT NJOCK | FE21A285 |
| NZEMTEJUH SYLVANUS | FE21A296 |
| NKWENKAM JENNIFER | FE21A279 |
| VTALAH DE GERAUD | FE21A329 |
| FOLETIA AJONGAKUE SYNCLAIR | FE21A190 |

**MOBILE APP REQUIRMENTS FOR MOBILE-BASED ARCHIVAL AND RETRIEVAL OF MISSING OBJECT**

1. **PURPOSE**

The purpose of this report is to document the requirements and implementation of a mobile based application that will enable that will make users to archive and retrieve missing object using image matching techniques

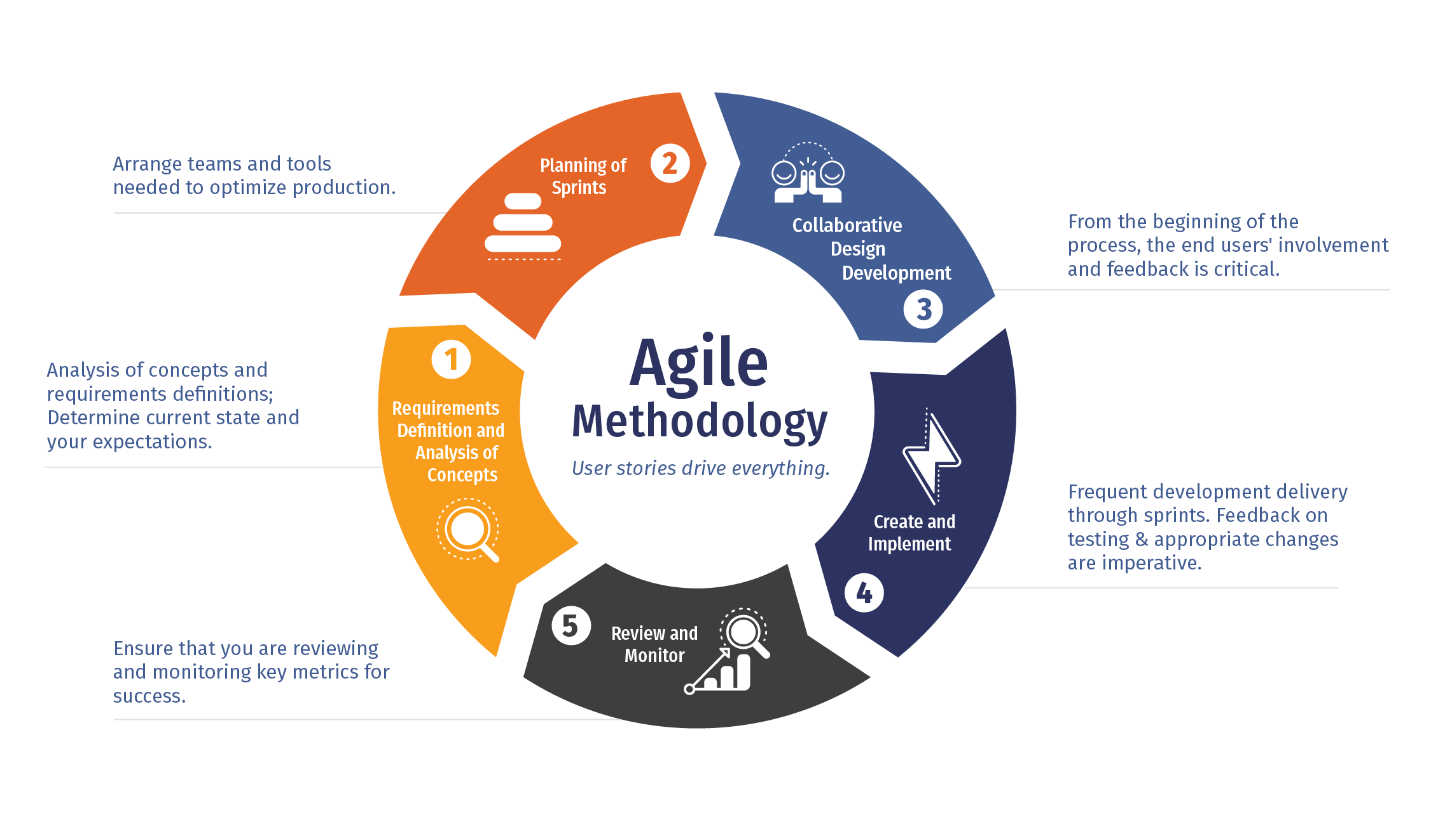
1. **SCOPE**

This document covers the following aspect of the application.

* Technical requirements
* User requirements
* System requirement
  + Functional requirements
  + Non-functional requirements
* Requirements based on the priorities
* User stories

1. **METHODOLOGY**

Agile methodology, with its emphasis on flexibility, collaboration, customer-centricity, incremental development, and continuous improvement, would be best suited for developing a Mobile-Based Archival and Retrieval of Missing Objects Application using Image Matching. By adopting Agile practices, tools, and principles, the development team can effectively navigate challenges, capitalize on opportunities, and deliver a successful, valuable, and high-quality solution that meets users' needs, expectations, and requirements throughout its lifecycle.



1. **Technical requirements and their description**
2. **User Registration and Authentication:**

* Allow users to create accounts and log in securely.
* Implement authentication mechanisms to verify users' identities and protect their information.

1. **Image Upload and Storage:**

* Enable users to capture and upload images of missing objects.
* Store uploaded images securely in the database with associated metadata (e.g., object type, location, date).

1. **Image Matching and Retrieval:**

* Implement an image matching algorithm to compare uploaded images with the database.
* Retrieve and display matching results, including similar or potentially matching objects based on the image analysis.

1. **Search and Filter Functionality:**

* Provide users with search and filter options to easily locate missing objects by object type, location, date, or other relevant criteria.
* Display search results in a clear and organized manner, allowing users to view details and images of matching objects.

1. **User Profile and Management:**

* Allow users to manage their profiles, update information, and track their uploaded images and search history.
* Provide features to edit, delete, or archive uploaded images as needed.

1. **Notification and Alerts:**

* Implement notification and alert mechanisms to inform users about potential matches or updates related to their uploaded images.
* Enable users to receive notifications via email, SMS, or within the app.

1. **Reporting and Feedback:**

* Enable users to report false positives, provide feedback on search results, or request assistance.
* Implement a system to handle user reports, feedback, and inquiries efficiently.

1. **Privacy and Security:**

* Implement privacy controls and security measures to protect users' personal information and uploaded images.
* Comply with data protection regulations and standards to ensure confidentiality, integrity, and availability of data.

1. **User Guidance and Support:**

* Provide users with guidance, instructions, and tips on using the app effectively.
* Offer customer support and assistance through FAQs, help articles, or direct contact options.

1. **Speed Up Robust Feature(SURF) Matching Algorithm**

* extract the feature vectors from a query image of a lost item and compare it to those of the existing found items images in our database.

1. **REQUIREMENTS BASE ON TYPES**
2. USER REQUIRMENTS

* Registration and authentication
* Login
* Login
* Upload image of missing object (incorporates searching database for potential matches)
  + User can edit the uploaded image
  + Delete the image
* Upload image of found object (incorporates searching database for potential matches)
  + User can edit the uploaded image
  + Delete the image
* Search and filter to locate missing objects
* Display the result of the search
* Notification and alerts sent when object is found.
* Reporting and feedback

1. SYSTEM REQUIRMENTS

Here we are going to categories the system requirements into to categories which are

* **Functional requirement**
  + Registration
  + Login
  + Change password
  + Upload lost object image(includes running the surf algorithm on the image)
  + Upload found object image(includes running the surf algorithm on the image)
  + Search and filter
  + Display results
  + Notification alerts\
* **Non-functional requirement**
* Performance:
* Response Time:Ensure quick and efficient image upload, matching, and retrieval processes.
* scalability: Handle increasing numbers of users, images, and data without compromising performance.
* Resource Utilization: Optimize memory, CPU, and bandwidth usage to maximize efficiency and minimize resource consumption.

Tools:

- Performance monitoring tools (e.g., New Relic, Dynatrace)

- Load testing tools (e.g., JMeter, LoadRunner)

* Usability:
* User Interface: Design an intuitive, user-friendly, and accessible mobile interface.
* User Experience: Provide clear instructions, guidance, and feedback to users during image upload, search, and retrieval processes.
* Accessibility:\*\* Ensure the application is accessible to users with disabilities, supporting features like screen readers, voice commands, and text-to-speech.

Tools:

- User experience (UX) design tools (e.g., Adobe XD, Sketch)

- Accessibility testing tools (e.g., AXE, WAVE)

* Reliability:
* Availability:\*\* Ensure high availability and uptime of the application to support users' needs and demands.
* Fault Tolerance:\*\* Implement mechanisms to handle and recover from failures, errors, or issues without data loss or service interruption.
* Backup and Recovery:\*\* Regularly backup data, images, and configurations, and implement effective recovery strategies in case of data loss or system failures.

Tools:

- Monitoring and alerting tools (e.g., Nagios, Prometheus)

- Backup and recovery solutions (e.g., Acronis, Veeam)

* Security:
* Data Protection:\*\* Implement encryption, data masking, and secure transmission protocols to protect users' personal information, images, and data.
* Authentication and Authorization:\*\* Ensure secure user authentication, authorization, and access control mechanisms to protect against unauthorized access, misuse, or malicious activities.
* Compliance:\*\* Comply with relevant privacy laws, regulations, and standards (e.g., GDPR, CCPA) to ensure confidentiality, integrity, and security of users' data and information.

Tools:

- Security scanning and vulnerability assessment tools (e.g., OWASP ZAP, Nessus)

- Identity and access management solutions (e.g., Okta, Auth0)

* Compatibility:
* Platform Support:\*\* Ensure compatibility and support for various mobile platforms, devices, and operating systems.
* Integration:\*\* Enable seamless integration with other systems, services, or platforms to enhance functionality, interoperability, and user experience.

Tools:

- Cross-platform development frameworks (e.g., React Native, Flutter)

- API management and integration platforms (e.g., MuleSoft, Apigee)

* Maintainability:\*\*
* Code Quality:\*\* Maintain high code quality, readability, and maintainability through clean coding practices, code reviews, and refactoring.
* Documentation:\*\* Provide comprehensive documentation, guidelines, and resources to support development, deployment, and maintenance activities.
* Updates and Upgrades:\*\* Ensure ease of updates, upgrades, and modifications to adapt to new technologies, features, or requirements over time.

Tools:

- Version control systems (e.g., Git, SVN)

- Continuous integration and continuous delivery (CI/CD) tools (e.g., Jenkins, GitLab CI/CD)

These non-functional requirements and tools form the foundation for developing a robust, secure, and scalable Mobile-Based Archival and Retrieval of Missing Objects Application using Image Matching. By addressing these requirements and leveraging appropriate tools and technologies, developers can ensure the application meets users' needs, performs effectively, and maintains high standards of quality, reliability, and security throughout its lifecycle.

1. REQUIREMENTS BASED ON PRIORITIES

* MUST HAVE
  + Upload lost image
  + Upload found image
  + Search and filter
  + Display results
  + Registration
  + Login
  + Track uploads
* SHOULD HAVE
  + Change password
  + Privacy and security
  + Notification alerts
* COULD HAVE
  + Customer support
  + Reporting and feedback
* WON’T HAVE
  + Payment when object is found so it can be sent the owner of the object.

1. USER STORIE AS PER THE APP REQUIRMENT
2. A business logo displays when users load the application.
3. The application opens the home page with a cardview presentation of some objects, a plus button and a search bar with a filter button beside it.
4. The user clicks on a plus icon button at the bottom of the screen to post a lost or found image for which the algorithm is going run on.
5. The user clicks on the lost button to open a new page were the user can input the image details such as the image category, select a location, date lost, name of item, description, a button to upload the images of missing object and a button to publish
6. User click publish and the system search and compare the images with the database for potential matches and displays found if it finds the object or not found.
7. The user click on found button to open a new page were the user can input the image details such as the image category, select a location, date lost, name of item, description, a button to upload the images of missing object and a button to publish and step (v) is repeated
8. If the object is found, notify the lost item reporter
9. User claims the item from the authority.

References.

* Mohammed Ghazal, Fasila Haneefa, Samr Ali, Yasmina AlKhalil, and Eman Rashed, requirement of an archival and retrieval of missing object app', 2016