THE UNIVERSITY OF BUEA
P.O Box 63,
Buea, South West Region
CAMEROON
Tel: (237) 3332 21 34/3332 26 90
Fax: (237) 3332 22 72

REPUBLIC OF CAMEROON

Peace-Work-Fatherland

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

CEF440: INTERNET PROGRAMMING (J2EE) AND MOBILE PROGRAMMING

DESIGN AND IMPLEMENTATION OF A MOBILE-BASED ARCHIVAL AND RETRIEVAL OF MISSING OBJECTS APPLICATION USING IMAGE MATCHING

TASK TWO: **REQUIREMENTS GATHERING**

Course Supervisor:
Dr. NKEMENI VALERY
University Of Buea

GROUP 1 MEMBERS

	NAME	MATRICULE
1	MEWOABI NGUEFACK DORE	FE21A239
2	ALEANU NTIMAEH ENOW	FE21A134
3	OJONG-ENYANG OYERE	FE21A297
4	FONGANG KELUAM PAUL DIEUDONNE	FE21A193
5	TANWIE BRUNO ADEY	FE21A316

Table of Contents

1. INTRODUCTION

In an era characterized by rapid technological advancement and an ever-increasing reliance on mobile applications, the need for innovative solutions to everyday challenges has never been more pressing. One such challenge is the distressing experience of losing personal belongings, be it a cherished item of sentimental value or a practical necessity crucial for daily routines. The emotional toll and inconvenience caused by such losses are undeniable, often compounded by the arduous process of attempting to retrieve or replace the missing objects.

Recognizing the urgency of addressing this issue, the proposed project sets out to develop a groundbreaking mobile application aimed at revolutionizing the archival and retrieval of missing objects through the power of image matching technology. This application will not only streamline the process of reporting lost items but also enhance the likelihood of successful recovery, thereby alleviating the stress and frustration associated with such incidents.

1.1. Requirement Gathering Process:

To ensure the success and effectiveness of the envisioned application, a comprehensive requirement gathering process will be undertaken, encompassing a series of meticulously planned steps. These steps will serve as the foundational framework upon which the entire project will be built, guiding the development team towards the realization of the specified objectives. The following is an overview of the key stages in the requirement gathering process:

Stakeholder identification and analysis will precede **user requirements elicitation**, ensuring a comprehensive understanding of all involved parties. **Functional requirements** will then be delineated, followed by **non-functional requirements** to ensure optimal performance and user experience. **System requirements**, **data requirements**, and **quality attributes** will be defined to shape the application's technical foundation and operational parameters.

User interface requirements and **integration requirements** will subsequently be addressed, ensuring seamless interaction and compatibility with external systems. Finally, **risk assessment** will be conducted to identify and mitigate potential obstacles throughout the project lifecycle.

It's crucial to note the distinction between the requirements gathering phase and the subsequent requirements analysis phase. While the former focuses on identifying and documenting stakeholder needs and system specifications, the latter involves a deeper analysis and interpretation of gathered requirements to inform the design and development process effectively.

2. REQUIREMENTS GATHERING PROCESS: STEPS AND METHODOLOGY

2.1. Stakeholder Identification and Analysis

2.1.1. Who is a stakeholder?

A stakeholder is any individual, group or organization that has an interest in a particular project, initiative, or system. Stakeholders can influence or be influenced by the decisions and outcomes related to the project or system. They may have varying levels of influence, involvement, and stakes in the success or failure of the endeavor.

For a "mobile-based archival and retrieval of missing objects application using image matching," there are several stakeholders involved, each with distinct roles, responsibilities, and expectations.

2.1.2. Project Stakeholders

Some key stakeholders and their respective involvements are explained below;

i. End Users:

They are the primary users of the application.

Role:

- Reporting lost objects
- Searching for lost objects
- Report discovery of missing objects
- Search for information about lost items found
- Possibly providing feedback on their experience with the application

Responsibilities:

- Providing accurate information about lost objects
- Reporting found objects accurately
- complying with any procedures for returning items to their rightful owners.
- adhering to any guidelines or protocols outlined by the application

Expectations:

- Expect the application to provide a user-friendly experience
- Efficient search and retrieval functionalities
- Timely updates on the status of reported lost objects

ii. Development team:

Role:

- Responsible for designing, developing, and maintaining the mobile application
- Responsible for developing and maintaining image matching algorithms
- Responsible for developing the database

Responsibilities:

- Creating an intuitive (easy to use and understand) user interface
- Implementing robust image matching algorithms
- Ensuring data security and privacy
- Providing ongoing technical support and updates

Expectations:

- Regular updates and improvements based on user feedback and technological advancements

iii. Law Enforcements Agencies:

Role:

- Utilize the application to access information about reported lost objects
- Assist in investigations related to missing items

Responsibilities:

- Validating reported lost objects
- Using the application as a tool for crime prevention and investigation

Expectations:

- Expect the application to facilitate the exchange of relevant information
- Expect the application to facilitate the exchange of relevant information
- Contribute to the overall effectiveness of law enforcement efforts

iv. Businesses and Organizations (Lost and Found Services):

Role:

- Businesses or organizations that manage lost and found services, such as transportation companies, event venues, hotels, and shopping centers Use the application to facilitate their own processes

Responsibilities:

- Integrating the application into their existing lost and found processes
- Promoting its use among customers and employees
- Collaborating with developers to enhance its functionality.

Expectations:

- Expect the application to improve the efficiency of lost and found operations
- Reduce costs associated with managing lost items
- Enhance customer satisfaction and loyalty.

2.2. User Requirements Elicitation

After stakeholders are identified and analyzed, the requirements elicitation phase begins. This phase involves gathering information from stakeholders regarding their requirements, preferences, and constraints related to the project. It is the process of gathering information about the needs and expectations of stakeholders by conducting interviews, questionnaires, task analysis, Delphi techniques, prototyping, etc.

For this project, the requirements Elicitation phase was carried out as follows

2.2.1. Brainstorming:

A couple of meetings were held among ourselves (the team) to discuss about how we can come up with sufficient information about needs and expectations of stakeholders of this system and also to give individual opinion about what we thought was needed. These meeting gave a common idea of how to move on to the actual elicitation phase which we did through conduction of interviews to the end users

2.2.2. Interviews:

Interviews were conducted on some classmates, family members, friends who can be considered end users of the project application who gave their objective opinion about their needs which will be stated and described below. These interviews were the heart of our requirements gathering process and helped us outline the functional and non-functional requirements of our systems. In such interviews we asked the following questions to our participants

- Participants were asked to recount experiences of losing valuable items and describe the emotions and steps taken to retrieve them.
- The features they deem essential in an application intended for locating missing items.

- Participants were also prompted to discuss the frequency and types of items typically lost or misplaced.
- The challenges they face when attempting to locate missing items, including specific obstacles or frustrations encountered during the process.
- They were also queried about their current methods of organizing belongings and any tools or techniques utilized for this purpose.
- Input was solicited on the information participants deemed pertinent when reporting a lost item through a mobile application, such as descriptions, photos, locations, or timestamps.
- Perspectives were gathered on the importance of privacy and data security when using a mobile application for reporting lost items, along with expectations regarding measures to safeguard personal information.
- Participants were invited to share scenarios or circumstances in which they envisioned the application being particularly beneficial or valuable.
- Expectations were explored concerning the accuracy and reliability of the application in matching lost items with found objects, as well as criteria for defining its success.
- Suggestions were welcomed for additional features or functionalities perceived to enhance the usefulness or usability of the application.

2.2.3. Research:

Independent research was conducted by members on several websites and documentaries for similar projects and the most prioritized objectives or requirements taken into consideration

2.2.4. Task analysis:

Task analysis provided valuable insights into user behaviors, needs, and preferences which are essential for capturing accurate and comprehensive requirements. Task analysis is a systematic process that helps in understanding the steps and activities involved in completing a specific task or achieving a particular goal. This helped to appropriately deduce our system requirements specifications or functional, non-functional requirements, constraints and others after conducting the requirements elicitation process

2.3. Requirements Specification

Requirement's specification involves documenting the identified requirements in a clear, consistent, and unambiguous manner to create a comprehensive document understandable by both the development team and stakeholders. It aims to provide formal software requirement models, encompassing functional, non-functional requirements, constraints, and acceptance criteria. Usually, in the requirements gathering phase, functionalities are simply listed based on elicitation and stakeholder analysis and actual categorization into functional or non-functional requirements is done in the requirements analysis phase but for the sake of organization, we will do the categorization in this phase.

After the requirements Elicitation phase, we were able to come up with the following requirements specifications for our application

2.3.1. Functional Requirements

The functional requirement of our system defines what the system should do, specifying the behaviors, actions and capabilities of our image recognition software. It describes specific features, functions and interactions that users expect from the app. Below are some of the functional requirements based on user needs and their importance to the functioning of the application in order to achieve the project objectives.

1. Image recognition and Real time image processing

This is the core functionality of the image processing software as it the primary reason why users access the system. The platform implements an image recognition algorithm to analyze uploaded images and identify potential matches from other images which have been stored in the system's database. This is done through the integration of an image recognition and image processing API.

2. User Authentication and Security

This feature enables users to be able to create accounts and log in securely into the system directly from their mobile phones. This feature also enables users to submit visual information for the application to analyze ensuring security in the user private data and access to only authorized users in the system. It is implemented through the use of passwords and biometric which should match those previously given by the user in the database.

3. Database Integration

The application is connected to a database which contains information about missing items, user information, descriptions and other relevant details. The database is updated every time to ensure that it is accurate and ensures correct outputs. An inaccurate database leads to mismatch and wrong outputs.

4. User Interface Design

An intuitive and user-friendly interface is designed which makes navigation through the application easy, uploading images smooth and searching for missing items quick. Also, providing clear instructions on how to use the app hence attracts more users who have lost items to look up to the application for help in locating those items.

5. Notification System

Keeping users informed about potential matches for their missing items increases engagement and encourages active participation in the search process. The app will have a notification system to alert users when a potential match for their missing item is found. Notifications can be sent via push notifications or email, depending on user preferences.

6. Feedback Loop

The application allows users to provide feedback which helps improve the app's accuracy over time, enhancing user satisfaction and the overall effectiveness of the platform. Users will have the ability to provide feedback on potential matches suggested by the application. This feedback helps improve the accuracy of the system over time by providing valuable data for training the image recognition algorithms.

7. Location-based Services

While important for some users, this feature may not be as critical as others, depending on the nature of the missing items and the user's search preferences. Utilizing GPS technology to enable location-based searching for missing items. Users can filter search results based on their current location or specify a particular area of interest, improving the relevance of search results.

8. Image Capture

Although convenient, users may be willing to upload images from their device's gallery if real-time image capture is not immediately available. Integration with the device's camera functionality to allows users capture images directly within the app. This feature enhances user convenience by enabling them to quickly upload images of missing items without leaving the application.

9. Offline Mode

Providing offline functionality for users to access certain features of the application, such as viewing previously searched items or accessing saved data, even when they are not connected to the internet. This enhances usability and ensures uninterrupted access to essential features.

By prioritizing features in this manner, we focus on delivering the most impactful functionalities that align with the project's objectives and user needs. These functionalities shall later on be analyzed in the analysis phase to filter those that are essential in the building of our MVP.

2.3.2. Non-Functional Requirements

Nonfunctional requirements specify how the system should perform, giving the quality attributes, constraints and characteristics that governs the system behavior. They are also referred to as quality attributes and are critical in ensuring that the software meets the expectations and needs of the users. Below are some of the non-functional requirements our system will implement.

1) Performance

The system should be able to process image recognition requests quickly and efficiently, providing near-real-time results to users. Response times for image processing and database queries should be optimized to minimize user wait time. While giving the correct and accurate output of the image processing.

2) Scalability

The application should be designed to handle a potentially large volume of users and image uploads. It should be able to scale horizontally to take in an increasing number of users without sacrificing its performance or its efficiency in the output.

3) Reliability

The system should be highly reliable, with minimal downtime or service interruptions. This requires robust error handling, fault tolerance, and backup mechanisms to ensure continuous operation. This means when the system is going through a fault or update, the users should not be able to suspect a thing.

4) Availability

The application should be available to users whenever they need it, with high availability and uptime. This may involve implementing redundant servers, load balancing, and failover mechanisms to mitigate the impact of hardware failures or maintenance.

5) Privacy

Users' privacy should be respected, and personal data should be handled in accordance with relevant privacy regulations. This involves obtaining user consent for data collection and processing, as well as implementing privacy-enhancing features such as anonymization and data minimization.

6) Security

Data security is critical to protect user information and prevent unauthorized access to sensitive data. The system should implement encryption, secure authentication protocols, and access control mechanisms to safeguard user data and prevent data breaches.

7) Compatibility

The application should be compatible with a wide range of mobile devices and operating systems to maximize accessibility for users. It should be tested on different platforms to ensure consistent performance and user experience across devices.

8) Usability

The user interface should be intuitive and easy to navigate, with clear instructions and feedback to guide users through the process of uploading images and interpreting search results. Accessibility features should also be implemented to accommodate users with disabilities.

9) Maintainability

The system should be designed with maintainability in mind, with clean and well-documented code that is easy to understand and modify. This facilitates future updates and enhancements to the application without introducing unnecessary complexity or risk.

10) Regulatory Compliance

The system should comply with relevant laws and regulations governing data protection, privacy, and security. This includes adherence to industry standards and best practices for handling sensitive information and ensuring transparency in data processing practices.

By addressing these non-functional requirements, the image recognition system can deliver a reliable, secure, and user-friendly experience that meets the needs of both users and stakeholders.

2.3.3. Constraints and Limitations

Several limitations may affect the design and implementation of the project:

1. Accuracy of Image Recognition

The effectiveness of image recognition algorithms can be limited by factors such as image quality, lighting conditions, and occlusions. Achieving high accuracy may require sophisticated algorithms and extensive training data.

2. Availability of Data

The success of the application depends on the availability of a comprehensive database of missing items. Limited data availability or incomplete information about missing items may reduce the effectiveness of the search feature.

3. Resource Constraints

The processing power and memory limitations of mobile devices may impose constraints on the complexity of image recognition algorithms and the amount of data that can be processed locally. This may require offloading some processing tasks to remote servers, introducing latency and bandwidth constraints.

4. Connectivity Issues

Users in areas with poor network connectivity may experience difficulties uploading images or receiving real-time updates. Offline functionality and asynchronous data synchronization can help mitigate the impact of connectivity issues.

5. Privacy Concerns

Users may have concerns about uploading personal images and data to the application, especially if they are related to sensitive or private information. Implementing robust privacy controls and transparent data handling practices is essential to address these concerns.

6. Regulatory Compliance

The project may be subject to various regulatory requirements, such as data protection laws and regulations governing the use of image recognition technology. Ensuring compliance with these regulations may require additional resources and expertise.

2.4. System Requirements

System requirements define the hardware and software components necessary for the successful operation of the software system. These requirements outline compatibility specifications, operating system requirements, and dependencies on external services or libraries. They ensure that the software can be deployed and accessed across various devices and platforms effectively.

2.4.1. Hardware requirements

- **Supported Mobile Devices:** The application should be compatible with popular mobile devices such as smartphones and tablets running on both iOS and Android platforms.
- **Network Connectivity:** The application should support both Wi-Fi and cellular network connections to ensure accessibility in various environments.

2.4.2. Software requirements

- **Operating Systems:** The application should be compatible with the latest versions of iOS and Android, ensuring broad device coverage.
- **Third-Party Dependencies:** Any third-party libraries or frameworks required for image processing

and database management should be identified and integrated into the development environment.

- **Compatibility**: The application should be designed to seamlessly integrate with other systems or services, such as cloud storage for data backup and retrieval.

2.4.3. Environmental Considerations

- **Lighting Conditions**: The application should be designed to handle different lighting conditions to ensure accurate image capture and matching.
- **Network Bandwidth**: Real-time image matching functionality should be optimized to minimize the impact on network bandwidth, especially in areas with limited connectivity.

2.5. Data Requirements

Data requirements outline the types of data to be collected, stored, and processed by the software system. These requirements specify data formats, structures, and storage mechanisms, ensuring efficient handling and management of data. Considerations are made regarding data privacy and compliance with relevant regulations to safeguard users' data throughout the system's lifecycle.

2.5.1. Types of Data

- **Images of Lost Items:** The application will collect and store images of lost items captured by users using their mobile devices.
- **Metadata**: Each image entry in the database should include metadata such as timestamp, location, description, and any additional user-provided information.
- **User Authentication Data:** User authentication data, including usernames, passwords, and authentication tokens, should be securely stored to ensure the security of user accounts.

2.5.2. Data Structures and Schema

- **Centralized Database**: The application should utilize a centralized database to store and manage archived objects efficiently.
- **Relational Schema**: The database schema should include tables for storing images, metadata,

user authentication data, and any other relevant information, following standard relational database design principles.

2.5.3. Data Privacy and Security

- **Compliance**: The application should comply with relevant data protection regulations, such as GDPR or CCPA, to ensure the privacy and security of user data.
- **Encryption**: Sensitive data, including user authentication credentials and personal information, should be encrypted both at rest and in transit to prevent unauthorized access.
- **Access Controls**: Role-based access controls should be implemented to restrict access to sensitive data and functionality based on user roles and permissions.

2.5.4. Data Retention Policies

- **Retention Period**: The application should define data retention policies specifying how long archived objects and associated data will be stored in the database.
- **Backup and Recovery**: Regular data backups should be performed to prevent data loss in case of system failures, with robust recovery mechanisms in place to restore data in the event of a

2.6. Quality Attributes

During this phase, the project team focuses on identifying, prioritizing, and specifying the quality attributes that are essential for meeting user expectations and achieving project objectives. By defining clear quality attributes, the development team can make informed decisions regarding design, implementation, and testing, thereby ensuring that the application meets the highest standards of quality and user satisfaction. One of the main terms we have under this topic is “software Quality Assurance”

2.6.1. Software Quality Assurance

Software quality assurance (SQA) is the part of quality management that includes a planned set of organizational actions. The purpose of these actions is to improve the software development process, introducing standards of quality for preventing errors and bugs in the product hence has to be planned prior to the design and implementation phases of the project

1. Plan the Testing and QA Processes:

Before starting the development process, planning should be done on how to ensure its quality through testing and quality assurance processes. A test policy should be created to outline our commitment to building a reliable and user-friendly app. The policy should define the types of items users can report as lost, how they should describe them, and how match verification shall be done. Additionally, a quality management plan shall be developed to establish our quality standards and a test strategy to outline our approach to testing. By planning ahead, a solid foundation can be set for the app's development and quality assurance.

2. Employ Test-Oriented Software Development Management:

When development begins, test-driven development (TDD) practices shall be adopted. Implying that before writing any code, tests shall be conducted on previous functionality to ensure the expected behavior of each feature. For instance, before implementing the feature for users to report lost items, tests shall be written tests to ensure that the reported items are correctly stored in the app's database. Pair programming shall also be encouraged, where two or more developers collaborate closely to write high-quality code. Embracing these practices, will help build a robust and reliable app from the ground up.

3. Use a Shift-Left Approach to Start Testing Early and Often:

Implementation of a shift-left approach to testing shall be done, meaning testing shall be done early in the development process. Even before the app is fully developed, small tests shall be conducted to validate basic functionalities such as user authentication and item reporting. By testing early and often, issues can be identified and fixed sooner, reducing the risk of encountering major problems later in the development process.

4. Conduct Formal Technical Reviews:

As progress is done with the development of the app, formal and technical reviews shall be conducted to evaluate its functionality and quality. These reviews will involve all team members to jointly examine the app's features and code to identify any issues or areas for improvement.

5. Ensure Suitable Work Environment for QA Team:

A suitable work environment shall be established for the team to ensure that testing activities are carried out effectively. Roles shall be defined clearly with responsibilities for each team member. Each team member will have specific tasks and goals to contribute to the app's quality assurance efforts. Fostering a collaborative and supportive work environment, we'll empower the team to deliver a high-quality app that meets the needs of our users.

2.7. User Interface and User Experience Requirements

User experience design (UX design) begins at the requirements stage and proceeds through all the stages of development, including the testing and post-release stages. The process of UX design includes research, prototyping, usability testing, and the actual designing part, during which lots of documentation and deliverables are produced. Several stages are involved in the planning of the UI/UX requirements

2.7.1. Research Stage:

In this stage, practical researched in carried out and information is gotten from individuals which can then be compiled into different formats that give a clearer view of the problem and the measures that can be taken to solve them. Such formats involve

i. User Personas:

User Personas are created and documented during the research stage. The information gathered during user interviews and surveys is compiled into functional user persona documents. User personas represent the key characteristics of real users, focusing on behavior, thought patterns, and motivation

Some created personas as shown below:

a) Neighbor Sarah:

A busy professional who frequently travels and often misplaces items in airports.

Behavior: Frequently travels for work, often rushes through airports.

Motivation: Needs a quick and efficient way to report lost items due to her busy schedule.

Thought Patterns: Values convenience and reliability in mobile applications.

b) Mark a quarter Friend :

A college student who often forgets his belongings in public places like libraries or cafes.

Behavior: Spends long hours studying in libraries or cafes.

Motivation: Wants a solution to help him locate misplaced belongings in public places.

Thought Patterns: Prefers simple and straightforward apps that are easy to use.

ii. User Scenarios:

A user scenario is a document that describes the steps a user persona will take to accomplish a specific task. User scenarios focus on what a user will do, rather than outlining the thought process. The set of scenarios can be either visual or narrative, and describe the existing scenarios or future functionality

- Create user scenarios outlining the steps users like Sarah and Mark would take to report and retrieve lost items.

Example scenario of a user scenario:

Scenario: Sarah loses her laptop at the airport.

Steps:

- Sarah opens the app and logs in.
- She selects "Report Lost Item" and provides details such as item type, description, and location last seen.
- Sarah uploads a photo of her lost laptop and submits the report.
- The app confirms receipt of the report and notifies Sarah that potential matches will be identified

iii. Scenario Map:

Scenario maps are used to compile the existing user scenarios into a single document. Scenario maps show all possible scenarios available at a given moment for every single function, as well as intersecting scenario steps

- Ensure the scenario map covers various user actions, such as reporting lost items, searching for items, and managing user profiles.

iv. User Story Map:

A user story map is formed from the backlog of the product. This type of document helps to arrange the user stories into future functions or parts of the application. A user story map can be a scheme or a table of user stories grouped in a particular order to denote the required functions for a certain sprint

Example user story map:

- Sprint 1: Implement user authentication and basic reporting functionality.

- **Sprint 2**: Integrate image matching algorithms and real-time notifications.

The above research formats help plan the application flow and have a better visioning of the application, the kind of flow and feel the users expect hence providing better user experience and Interface.

v. UX Style Guide:

The UX style guide is a document that includes the design patterns for the future product. It also describes all possible UI elements and content types used, defining the rules of how they should be arranged and work with each other. But, unlike a UI style guide, UX designers don't describe the actual look of the interface.

2.7.2. Prototyping and Design Stage

During the stage of prototyping and designing, a UX designer often works with the deliverables and updates documentation on pair with other team members, including the product owner, UI designers, and engineers**.**

i. Site Maps:

- Create a visual scheme representing the connection between all pages and functions of the application.

- Ensure the site map accurately reflects the structure of the app, including reporting, searching, and user profile management.

ii. User Flow Schemes:

- Map the steps users take through the application, depicting the logic of user movement.

Example user flow:

User logs in > Reports lost item > Receives notifications about potential matches > Confirms match > Retrieves lost item.

iii. Wireframes:

- Develop wireframes as blueprints for the UI, showing the arrangement of elements and their behavior.

- Focus on functionality rather than visual design, ensuring ease of use and intuitive navigation.

2.7.3. Testing and Post-Release Stage:

i. Usability Testing Reports:

- Conduct usability testing to gather feedback on the application's ease of use and functionality.

- Document test results and incorporate feedback to improve the user experience.

Throughout the UX design process, collaboration with product owners, UI designers, and engineers is crucial to ensure alignment with project goals and technical feasibility. Continuously updating and refining documentation ensures that the user experience remains a priority from requirements gathering to post-release iterations.

2.8. Integration Requirements

Integration requirements play a vital role in ensuring the seamless functionality of the Mobile-Based Archival and Retrieval of Missing Objects Application using Image Matching. This step focuses on defining the necessary interfaces, protocols, and data exchange mechanisms to facilitate smooth communication between different components of the application and external systems or services. By addressing integration requirements, we can ensure that the application seamlessly integrates with external databases, image recognition services, and other relevant systems, enhancing its effectiveness in helping users locate missing items. The following aspects were outlined with respect to integration requirements of our application

i. Interface Specification:

This involves defining interfaces between the application's frontend, backend, and external services such as image recognition APIs. Specification of communication protocols and data formats for transmitting image data, search queries, and response messages as well as Ensuring compatibility between different interfaces to enable seamless interaction and data exchange.

ii. External Database Integration:

External databases or repositories for storing and accessing archived objects and associated metadata will have to be Identified. Requirements for querying and retrieving data from external databases also have to be determined, for ensuring efficient and secure access. Mechanisms for synchronizing data between the application's database and external repositories would to have to be discussed ensure data consistency.

iii. Image Recognition Service Integration:

This will involve finding appropriate image recognition services or libraries to perform image matching and similarity scoring. Requirements for sending image data to the recognition service, processing results, and handling matches or mismatches have to be defined accurately. Handling mechanisms to address communication failures or inaccuracies in image recognition results also have to outlined

iv. Data Exchange Mechanisms:

This will include the Specification of mechanisms for exchanging data between the application and external systems, such as RESTful APIs or web services, the definition of data transformation and mapping requirements to ensure compatibility between different data formats and structures. It also

involves the listing of secure communication protocols and authentication mechanisms to implement for the protection of sensitive data during transmission.

v. Compatibility Testing:

This involves discussing the methods and tools to be used for the conduction of compatibility tests to ensure that the application integrates seamlessly with various mobile devices, operating systems, and network environments. It also includes Validation of the compatibility of interfaces, protocols, and data exchange mechanisms across different platforms and configurations as well as Identification and addressing of any compatibility issues or interoperability challenges encountered during testing.

vi. Performance Optimization:

This involves how to optimize the performance of integration processes to minimize latency and response times. It also focuses on the need and how to Implement caching mechanisms to store frequently accessed data and reduce the need for repeated queries as well as Monitoring system performance metrics and optimize resource utilization to ensure optimal responsiveness and scalability.

vii. Documentation and Maintenance:

The Integration requirements will have to be documented as per the requirements management phase, including interface specifications, external service dependencies, and data exchange protocols, providing comprehensive documentation and guidelines for developers and system administrators to configure, deploy, and maintain integration components. Updates will have to be done on the documentation regularly to reflect changes in integration requirements or external dependencies.

By addressing these integration requirements tailored to the Mobile-Based Archival and Retrieval of Missing Objects Application, we can ensure seamless communication with external systems and services, enhancing the application's functionality and usability for users in locating their missing items effectively. These Integration requirements shall be processed in the analysis phase to come up with the best tools and methods to ensure proper integration and functioning of our application

2.9. Risk Assessment

Risk assessment is a critical step in the development of our system. This process involves identifying, analyzing, and mitigating potential risks that could affect the project's success, user satisfaction, or data security. By conducting a thorough risk assessment, we can proactively address challenges and uncertainties, minimizing their impact on the application's development and deployment. Under risk assessment gathering, the following points were gathered for analysis and discussion:

i. Identification of Risks:

This involve Identifying potential risks that could arise throughout the project lifecycle, including technical, operational, and external factors. The risks to be considered involve Consider risks related to technology dependencies, data privacy regulations, user adoption, and market competition. Stakeholders, development team members, and domain experts all have to be involved in brainstorming sessions during the requirements analysis phase to identify a comprehensive list of risks. As well as how they can be solved or mitigated

For our system, the following risked could be identified:

a) Technical Risks:

- Dependency on external image recognition algorithms or APIs may lead to service disruptions or changes in functionality.
- Compatibility issues with different mobile devices, operating systems, or network environments could impact the application's performance and usability.
- Inadequate scalability may result in performance degradation or system failures during periods of high user activity.

b) Data Security Risks:

- Vulnerabilities in the application's data storage or transmission mechanisms could lead to unauthorized access or data breaches.
- Insufficient encryption or authentication mechanisms may compromise the confidentiality and integrity of user data.
- Inadequate measures for protecting personal information could result in privacy violations and regulatory compliance issues.

c) User Adoption Risks:

- Poor user interface design or lack of user-friendly features may lead to low adoption rates and user dissatisfaction.
- Inadequate marketing or promotion strategies may result in limited awareness and uptake of the application among potential users.
- Negative perceptions or resistance to change from users accustomed to traditional methods of retrieving lost items may hinder adoption and usage.

d) Operational Risks:

- Insufficient server capacity or inadequate infrastructure may result in system downtime or performance degradation.
- Lack of robust backup and recovery mechanisms could lead to data loss or corruption in the event of system failures or disasters.

- Inadequate support or maintenance processes may result in delays in addressing user issues or software bugs.

e) Regulatory Risks:

- Failure to comply with data protection regulations, such as GDPR or HIPAA, could result in legal liabilities and financial penalties.
- Changes in regulatory requirements or legal frameworks may necessitate updates or modifications to the application, impacting project timelines and budgets.
- Inadequate documentation or transparency regarding data handling practices may lead to compliance failures and reputational damage.

ii. Risk Analysis:

Here, Assessing the likelihood and potential impact of each identified risk on the project objectives, schedule, and budget is of primary concern Risks need to be prioritized based on their severity and probability of occurrence, focusing on those with the highest impact or likelihood and the root causes and contributing factors of each risk have to be analyzed to develop effective mitigation strategies.

iii. Risk Mitigation:

This step strives at developing risk mitigation strategies and action plans to address identified risks proactively, Implementation of preventive measures to minimize the likelihood of occurrence or mitigate the impact of high-risk events as well as Allocation of resources, budget, and contingency plans to address potential risks and uncertainties effectively.

iv. Monitoring and Control:

Here mechanisms for monitoring and controlling risks throughout the project lifecycle are Established. Regular review and update the risk register is performed to reflect changes in risk likelihood, impact, or mitigation status. Discussions on how Risk response strategies can be promptly as risks evolve or new risks emerge during project execution are done in this step.

v. Communication and Reporting:

This involves how risk assessment findings and mitigation plans are communicated to stakeholders, project sponsors, and team members. As well as how regular updates on the status of identified risks, mitigation efforts, and risk response activities are being provided. It will also involve how transparency and collaboration can be fostered to ensure that all stakeholders are informed and engaged in risk management efforts.

3. IMPORTANCES AND BENEFITS OF REQUIREMENTS ENGINEERING

3.1. Why Requirement Gathering is important?

Requirement gathering holds immense importance in software development for several critical reasons:

i. Clarity of Project Objectives:

Requirement gathering sets the stage by defining and clarifying the objectives of the software project. It ensures that all stakeholders, including clients, users, and development teams, have a shared understanding of what needs to be achieved.

ii. Customer Satisfaction:

Understanding and meeting customer needs is paramount for customer satisfaction. Requirement gathering allows developers to comprehend the expectations of end-users and clients, leading to the creation of a product that aligns with their desires and requirements.

iii. Scope Definition:

Clearly defined requirements help in establishing the scope of the project. This delineation is crucial for managing expectations, avoiding scope creep (uncontrolled changes to project scope), and ensuring that the project stays on track.

iv. Reduced Misunderstandings:

Ambiguities and misunderstandings are common sources of project failures. Requirement gathering facilitates clear communication between stakeholders, reducing the risk of misinterpretations and ensuring that everyone involved is on the same page.

v. Risk Mitigation:

Identifying and addressing potential issues at the requirements stage helps mitigate risks early in the development process. This proactive approach minimizes the chances of costly errors, rework, and delays later in the project life cycle.

3.2. Benefits of Requirements Gathering:

The benefits of effective requirements gathering in software development include:

i. Cost Reduction:

One of the primary benefits of effective requirements gathering is cost reduction. When requirements are well-defined and thoroughly understood at the beginning of a project, it minimizes the likelihood of costly changes and rework later in the development process.

ii. Customer Satisfaction:

Clear and accurate requirements gathering directly contributes to customer satisfaction. When the end product aligns closely with the expectations and needs of the stakeholders, it enhances user experience and meets customer demands. This satisfaction is not only vital for the success of the current project but also contributes to positive relationships between the development team and clients, fostering trust and potential future collaborations.

iii. Improved Communication:

Requirements gathering serves as a communication bridge between various stakeholders involved in a project, including developers, clients, users, and project managers. Miscommunication is a common source of project failures and delays. By clearly documenting and understanding requirements, the development team ensures that everyone involved has a shared vision of the project objectives, functionalities, and constraints.

iv. Efficient Resource Utilization:

Thorough requirements gathering enables the efficient allocation and utilization of resources. Resources, including time, manpower, and technology, are finite and valuable. When requirements are well-defined, project teams can allocate resources more accurately, avoiding unnecessary expenditures or overcommitting resources to certain aspects of the project.

v. Enhanced Quality:

Well-documented requirements serve as the foundation for quality assurance throughout the development process. When the project team has a clear understanding of what needs to be achieved, they can establish quality standards and criteria from the outset. This clarity enables the implementation of effective testing strategies, ensuring that each aspect of the system is thoroughly evaluated against the specified requirements.

vi. Risk Management:

Requirements gathering is a crucial component of effective risk management. By identifying potential risks early in the project, stakeholders can proactively address ambiguities, conflicting requirements, and other challenges that could pose a threat to the project's success.

vii. Accurate Planning:

Accurate project planning is dependent on a clear understanding of project requirements. When requirements are well-documented, project managers can create realistic schedules, milestones, and deliverables. This accurate planning is crucial for setting expectations, managing stakeholder timelines, and ensuring that the project progresses according to the established timeline.

4. CONCLUSION

The requirements gathering phase not only serves as the bedrock of the Mobile-Based Archival and Retrieval of Missing Objects Application using Image Matching but also sets the stage for its subsequent development phases. Through meticulous stakeholder identification and analysis, we have gained profound insights into user needs and expectations. By employing interviews and careful analysis, we unearthed critical user challenges and identified key features necessary for an effective application.

Furthermore, the delineation of both functional and non-functional requirements ensures that the application meets stringent standards of reliability, usability, and security. These gathered requirements will serve as the raw material for the requirements analysis phase. In this upcoming phase, these insights will undergo careful scrutiny and refinement to inform the application's design and development. Through detailed analysis, we will derive exact solutions, tools, and procedures necessary to address the identified user needs and challenges effectively.

In essence, the requirements gathering phase not only lays the groundwork for the project's success but also provides the critical input for subsequent phases. By leveraging the insights gained during requirements gathering and subjecting them to rigorous analysis in the upcoming phase, we will ensure that the Mobile-Based Archival and Retrieval of Missing Objects Application emerges as a robust, user-centric solution that meets the highest standards of quality and effectiveness.

REFERENCES

1. https://www.altexsoft.com/blog/software-testing-qa-best-practices/ date: 4/21/2024

2. “Requirements Engineering in Software Engineering” geeksforgeeks, https://www.geeksforgeeks.org/software-engineering-requirements-engineering-process/ date: 4/21/2024

3. https://blog.logrocket.com/product-management/what-is-requirements-gathering-template-examples/ date: 4/21/2024

4. https://www.seguetech.com/key-phases-software-development/ date: 4/21/2024

5. https://www.geeksforgeeks.org/requirements-gathering-introduction-processes-benefits-and-tools/ date: 4/21/2024