REPUBLIQUE DU CAMEROON
PAIX-Travail-Patrie
MINISTRE DE L'ENSEIGNEMENT
SUPERIEUR

FACULTE D'INGINERIE
ET TECHGNOLOGIE

REPUBLIC OF CAMEROON
Peace-Work-Fatherland
MINISTER OF HIGHER EDUCATION

FACULTY OF ENGINEERING
AND TECHNOLOGY

********************* UNIVERSITY OF BUEA ************************

COURSE TITLE: INTERNET PROGRAMMING AND MOBILE PROGRAMMING

PROJECT: DESIGN AND IMPLEMENTATION OF A BIOMETRIC STUDENT'S ATTENDANCE MOBILE APPLICATION

TASK THREE : REQUIREMENT GATHERING

PRESENTED BY:

NAME	REGISTRATION NUMBER
IHIMBRU ZADOLF ONGUM	FE21A203
CHE KASSINA KUM	FE21A158
NFOUA EUGENE MGBA	FE21A257
FONJI DANIEL KUKUH	FE21A194
EPIE MUKEH SANDRA	FE21A185

COURSE FACILITATOR: **MAY 2024**

Dr. VALERY NKEMENI

TABLE OF CONTENTS

ABSTRACT

This report outlines a comprehensive requirements analysis process for an attendance tracking system, focusing on key stakeholders' needs and system functionalities. It delineates steps for identifying stakeholders, capturing requirements, interpreting and recording them, modeling the requirements, and addressing both functional and non-functional aspects. The process emphasizes clarity, feasibility, and alignment with business needs, ensuring the development of a reliable and scalable solution. Through detailed analysis and prioritization, potential conflicts are resolved, and feasibility is assessed across hardware, software, and system requirements.

1. INTRODUCTION

Requirements analysis is a process used to determine the needs and expectations of a new product. It involves frequent communication with the stakeholders and end-users of the product to define expectations, resolve conflicts, and document all the key requirements.

One of the greatest challenges faced by any organization is to share the vision of the final product with the customers. Hence, a business requirements analysis involves a team effort of all the key stakeholders, software developers, end-users, and customer managers to achieve a shared understanding of what the product should do. This is always done in the early phase of any project to ensure that the final product conforms to all the requirements.

The development of an effective attendance tracking system requires a systematic approach to gather, interpret, and prioritize requirements. This report presents a structured process to ensure that stakeholders' needs are adequately addressed, leading to the creation of a robust solution. By identifying key stakeholders, including administrators, teachers, and students, and understanding their roles and responsibilities, the report sets the stage for capturing relevant requirements. Subsequent steps involve categorizing requirements, resolving conflicts, and assessing feasibility to guide the development process effectively.

1.1 Overview

The requirements analysis process encompasses multiple stages, each aimed at refining stakeholders' inputs into actionable development guidelines. Initially, key stakeholders and end-users are identified, followed by the capture and interpretation of their requirements. Through prioritization, conflict resolution, and feasibility analysis, the report ensures that the final requirements align with business objectives and technical constraints. Additionally, the modeling of requirements, both functional and non-functional, provides stakeholders with a tangible representation of the envisioned system, facilitating feedback and validation.

1.2 Objective

The primary objective of this report is to guide the development of an attendance tracking system by systematically analyzing stakeholders' requirements. By following the outlined process, stakeholders can collaboratively define clear, detailed, and feasible requirements, laying the foundation for a successful implementation. The report aims to foster transparency, collaboration, and accountability throughout the development lifecycle, ultimately delivering a solution that meets stakeholders' expectations and enhances organizational efficiency.

Requirements Analysis Process

1. Identify Key Stakeholders and End-Users

The first step of the requirements analysis process is to identify key stakeholders who are the main sponsors of the project. They will have the final say on what should be included in the scope of the project.

Next, identify the end-users of the product. Since the product is intended to satisfy their needs, their inputs are equally important.

We have three main stakeholders in our system, and below are the stakeholders, their roles and responsibility.

Administrators

Role

Administrators oversee data management, reporting, and system administration related to attendance tracking.

Responsibilities

- Data management: Administrators are responsible for maintaining accurate attendance records and ensuring data integrity.

- Reporting: They may need to generate attendance reports for various stakeholders, including faculty, parents, and accreditation bodies.

- System administration: Administrators manage the technical aspects of the attendance system, including configuration, updates, and user permissions

Teachers

Role

Teachers are the primary users responsible for taking attendance during classes. They play a pivotal role in ensuring the accurate and timely recording of student attendance

Responsibilities

- Taking attendance: Teachers are tasked with the responsibility of recording student attendance at the beginning of each class session.

- Monitoring attendance patterns: They need to track students' attendance trends over time to identify patterns of absenteeism or tardiness.

- Reporting: Teachers may be required to generate attendance reports for administrative purposes or to provide feedback to students

Students

Role

Students are the end-users who interact with the attendance system to register their presence in class.

Responsibilities

- Registering attendance: Students need to actively participate in the attendance tracking process by verifying their presence using biometric authentication.

- Compliance: It is the responsibility of students to comply with attendance policies established by their educational institution.

2. Capture Requirements

Here we ask each of the stakeholders and end-users their requirements for the new product. The requirements are then categorized for interpretation and elicitation. After which the requirements are then classified in terms of priority base on feasibility and other specifications

3. Interpret and Record Requirements

The goal here is to ensure that the requirements are clearly worded, sufficiently detailed, and related to business needs.

3.1 Define Requirements Precisely

Ensure that the requirements are clearly worded, sufficiently detailed, and related to business needs.

3.2 Prioritize Requirements

Prioritize requirements and list them out based on which ones are the “most critical” and which ones are just “nice-to-have”.

3.3 Carry Out an Impact Analysis

Carry out an impact analysis to make sure that you fully understand the consequences of the requirements.

3.4 Resolve Conflicts

Arrange a meeting with key stakeholders and resolve conflicting requirements. You can also perform a scenario analysis to explore how the requirements would work for different possible scenarios.

3.5 Analyze Feasibility

Perform a detailed analysis of the product based on the requirements gathered to determine its reliability and to identify any major problems. Once all the requirements are analyzed, create a detailed written document and circulate it among the key stakeholders, end-users and development teams.

4. Modeling the Requirements

This stage involves creating requirement models that ultimately allow stakeholders to imagine the product in the making. Various functions, data tables, external elements, and their relation to each other are represented in graphical forms. A graphical viewing of these things assists in finding flaws in the requirements. It allows the developers to see if there are any inconsistencies, missing, wrong, or unnecessary elements added to the system. Such requirement models can be divided into the following categories.

Requirement Eliciting

Functional Requirement

1. Registration

Definition: Users (both students and lecturers) should be able to register their biometric data and personal information securely within the system.

Impact Analysis: Without registration functionality, users cannot be uniquely identified within the system, rendering attendance tracking impossible.

Resolve Conflicts: Implement stringent security measures to safeguard sensitive biometric data and personal information during the registration process.

Feasibility: registration functionalities are standard features in biometric systems and can be implemented securely.

Implementation: Develop a user registration module where both students and lecturers can securely input their biometric data and personal information. Utilize encryption techniques to safeguard sensitive data during transmission and storage. Implement user authentication mechanisms to ensure only authorized users can access the registration system.

2. Biometric Data Capture

Definition: The system should accurately capture and store biometric data from students and lecturers to uniquely identify individuals for attendance tracking. This includes capturing biometric modalities, fingerprints in this case.

Impact Analysis: Accurate biometric data capture is crucial for ensuring the reliability and integrity of attendance records. Failure to capture biometric data accurately can lead to identification errors, compromising the accuracy of attendance tracking and potentially impacting administrative decisions.

Resolve Conflicts: Ensure that the chosen biometric modality is suitable for the intended environment and user demographics. Address any potential conflicts regarding privacy concerns or user acceptance through transparent communication and adherence to regulatory standards.

Feasibility: Modern biometric technology offers feasible solutions for data capture, with various modalities available to suit different preferences and requirements. However, feasibility considerations should include factors such as hardware compatibility, cost-effectiveness, and regulatory compliance.

Implementation: Integrate biometric devices (such as fingerprint scanners) into the system for accurate capture and storage of biometric data. Use APIs or SDKs provided by the device manufacturers to interface with the hardware. Implement data validation techniques to ensure the integrity and accuracy of captured biometric data.

3. Report Generation:

Definition: The system should automatically compile and generate comprehensive attendance reports or lists after the conclusion of each session or class.

Impact Analysis: Manual compilation of attendance lists is time-consuming and prone to errors, whereas automated generation streamlines the process and ensures accuracy and timeliness.

Resolve Conflicts: Ensure compatibility and seamless integration with existing administrative systems to facilitate the transfer of attendance data for further processing or analysis.

Feasibility: Modern biometric systems offer features for automated report generation based on recorded attendance data.

Implementation: Develop automated reporting functionality that compiles and generates attendance reports at the conclusion of each session or class. Utilize scheduling mechanisms to trigger report generation processes at predefined intervals. Implement data visualization tools to present attendance data in a clear and understandable format.

4. Attendance Activation

Definition: Lecturers must be able to initiate attendance tracking for each session by accessing the system through a secure login, selecting the appropriate class or session, and activating the attendance feature.

Impact Analysis: Without this feature, attendance tracking cannot commence, potentially leading to inaccuracies or omissions in attendance records, affecting overall reliability and integrity.

Resolve Conflicts: Ensure that the activation process is intuitive and efficient, with clear instructions provided to lecturers to minimize confusion or errors.

Feasibility: Modern biometric attendance systems offer user-friendly interfaces and straightforward activation procedures**.**

Implementation: Design a secure login system for lecturers to access the system and initiate attendance tracking for each session. Implement role-based access controls to ensure only authorized personnel can activate attendance tracking. Develop intuitive user interfaces with clear instructions for lecturers to navigate and activate the attendance feature.

5. Lecturers should be able to register students in special cases.

Define Requirements Precisely: Lecturers should have the ability to manually add or register students to the attendance roster in exceptional circumstances, such as late enrollment or transfer students.

Impact Analysis: Failure to provide this functionality may result in inaccuracies or omissions in attendance records for students who cannot register themselves through the standard process.

Resolve Conflicts: Implement appropriate authorization and authentication mechanisms to ensure that only authorized personnel can perform manual student registrations, mitigating the risk of misuse or abuse.

Feasibility: Feasible; manual student registration functionalities can be integrated into the system with proper access controls and audit trails.

Implementation: Provide lecturers with the capability to manually add or register students in exceptional cases, such as late enrollment or transfer students. Develop administrative interfaces with appropriate access controls to facilitate manual student registration. Implement validation checks to ensure data accuracy and integrity during the registration process.

6. The system should respond to errors promptly

Define Requirements Precisely: The system should provide immediate feedback and error messages in response to any encountered errors or anomalies, guiding users on corrective actions or next steps.

Impact Analysis: Delays or inadequacies in error response mechanisms may lead to user dissatisfaction, reduced productivity, and compromised data accuracy and integrity.

Resolve Conflicts: Implement comprehensive error handling mechanisms and user-friendly error messages across all system interfaces and interactions to ensure clarity and effectiveness.

Feasibility: error handling functionalities are standard features in software development and can be implemented effectively with proper design and testing.

Implementation: Implement robust error handling mechanisms throughout the system to provide immediate feedback and error messages in response to encountered errors or anomalies. Develop error logging functionality to record and track system errors for debugging purposes. Utilize asynchronous processing techniques to minimize disruptions and ensure prompt response to errors.

System Requirement

Hardware Requirements

1. **Reliable and Scalable Database Infrastructure**

Definition: This requirement entails having a database system that can reliably store and manage attendance data. It should also be scalable to accommodate the growth of data over time without compromising performance.

Impact Analysis: Insufficient reliability could lead to data corruption or loss, affecting the accuracy and integrity of attendance records.

Resolve Conflicts: To ensure reliability, robust database management systems (DBMS) with features like data replication, backup, and recovery should be employed.

Feasibility: Reliable and scalable database solutions are widely available in the market, ranging from open-source options like MySQL to enterprise-grade solutions like Oracle Database or Microsoft SQL Server.

Implementation: Deploy a reliable and scalable database management system (DBMS) capable of efficiently storing and managing attendance data. Configure database replication and backup mechanisms to ensure data redundancy and disaster recovery capabilities. Monitor database performance and scalability metrics to proactively address any scalability issues.

2. **Biometric Authentication Hardware**

Definition: This requirement involves the installation of biometric devices (fingerprint scanner) capable of accurately capturing biometric data from students for attendance registration.

Impact Analysis: Inaccurate or unreliable biometric authentication hardware may lead to errors in attendance tracking, compromising the system's accuracy and effectiveness.

Resolve Conflicts: Select biometric devices from reputable manufacturers with a proven track record of accuracy and security.
Implement encryption and other security measures to protect biometric data stored or transmitted by the hardware.

Feasibility:
Various biometric authentication hardware options are available on the market, ranging from standalone fingerprint scanners to integrated facial recognition systems. These devices are generally feasible for implementation, but careful consideration should be given to factors such as accuracy, security, and compatibility with the system.

Implementation: Procure and install biometric devices (e.g., fingerprint scanners) from reputable manufacturers with proven accuracy and security features. Integrate biometric hardware with the attendance tracking system using standardized protocols or APIs. Implement encryption techniques to protect biometric data both at rest and in transit.

3. Compatibility with Various Devices

Definition: The system should be compatible with a range of devices including tablets and smartphones (in our case) to ensure accessibility for users with different devices and preferences.

Impact Analysis: Incompatibility with certain devices may limit user access and adoption, reducing the system's effectiveness and user satisfaction.
Compatibility issues could result in additional development efforts and costs to ensure the system works seamlessly across different platforms.

Resolve Conflicts: Adopt cross-platform development frameworks or standards (e.g., responsive web design, native app development) to ensure compatibility with various devices.

Feasibility:
Ensuring compatibility with various devices is feasible with modern development technologies and practices.
Cross-platform development tools and frameworks make it easier to develop applications that work across multiple devices and platforms.

Implementation: Develop the attendance tracking system using cross-platform development frameworks (e.g., React Native, Flutter) to ensure compatibility with various devices, including tablets and smartphones. Utilize responsive web design principles to create user interfaces that adapt seamlessly to different screen sizes and resolutions.

4. **Robust Security Measures**

Definition: The system should implement robust security measures, including encryption protocols, secure storage solutions, and access controls, to protect sensitive biometric data from unauthorized access or misuse.

Impact Analysis: Security breaches could lead to unauthorized access to biometric data, compromising student privacy and confidentiality.
Non-compliance with security regulations (e.g., GDPR, HIPAA) could result in legal consequences and reputational damage for the institution.

Resolve Conflicts: Implement encryption algorithms (e.g., AES) to encrypt biometric data both at rest and in transit.
Use secure storage solutions (e.g., encrypted databases, secure file systems) to safeguard biometric data stored on servers or devices.

Feasibility: Robust security measures are feasible to implement using a combination of encryption technologies, access controls, and secure development practices.
While implementing security measures may require additional resources and expertise, the benefits of protecting sensitive biometric data outweigh the costs.

Implementation: Implement robust security measures, including encryption protocols, access controls, and secure authentication mechanisms, to protect sensitive biometric data from unauthorized access or misuse. Conduct regular security audits and vulnerability assessments to identify and address potential security vulnerabilities. Stay updated with industry best practices and regulatory requirements for data security.

Software requirements

1. **Compatibility with Android and iOS Operating Systems**

Definition: The application should be developed to run seamlessly on both Android and iOS operating systems, ensuring that users with different mobile devices can access the system.

Impact Analysis: Failure to support both Android and iOS could exclude a significant portion of potential users who use devices running these operating systems, reducing the system's overall effectiveness and adoption. Supporting multiple platforms may increase development complexity and maintenance efforts but is essential for reaching a wider audience.

Resolve Conflicts: Utilize cross-platform development frameworks like React Native or Flutter to build a single codebase that can be deployed to both Android and iOS platforms. Conduct thorough testing on both Android and iOS devices to ensure compatibility and address any platform-specific issues.

Feasibility: Developing applications for both Android and iOS platforms is feasible using cross-platform development frameworks, which streamline the development process and reduce the need for separate codebases.

Implementation: Develop native or cross-platform mobile applications for Android and iOS operating systems using development frameworks like React Native or Flutter. Conduct extensive testing on both Android and iOS devices to ensure compatibility and performance optimization for each platform.

2. Support for a Specific DBMS

Definition: The application will support MySQL or PostgreSQL for storing user data, attendance records, and system configurations.

Impact Analysis: The selected DBMS should be capable of efficiently storing and retrieving large volumes of data, ensuring fast and reliable access to attendance records. Compatibility with the chosen DBMS may influence system architecture, development decisions, and integration with other components.

Resolve Conflicts: Evaluate different DBMS options based on factors such as performance, scalability, reliability, and cost. Consider factors like ease of integration with other technologies, support for specific data types, and compatibility with the chosen development framework.

Feasibility: Choosing a specific DBMS is feasible, as there are various options available in the market catering to different needs and requirements.

Implementation: Select and configure MySQL or PostgreSQL as the database management system (DBMS) for storing user data, attendance records, and system configurations. Utilize database migration tools to migrate existing data to the selected DBMS. Optimize database queries and indexes for improved performance and scalability.

3. Offline Functionality Options

Definition: The application should include offline functionality options to ensure that users can access and use the system even in environments with limited or no internet connectivity.

Impact Analysis: Without offline functionality, users may be unable to access the system or record attendance in environments with poor or no internet connectivity, leading to disruptions in attendance tracking and data loss. Offline functionality may require additional development effort and complexity to synchronize data between the client and server when connectivity is restored.

Resolve Conflicts: Implement features like local data storage, caching, and synchronization mechanisms to enable offline functionality while ensuring data integrity and consistency. Design user interfaces and workflows to provide clear feedback and guidance to users when operating in offline mode.

Feasibility: Implementing offline functionality is feasible using modern web and mobile development technologies, which offer features like service workers, local storage, and background synchronization. While offline functionality adds complexity to the system, the benefits in terms of usability and accessibility outweigh the challenges.

Implementation: Implement offline functionality options using techniques such as local data storage, caching, and synchronization mechanisms. Develop offline-first strategies to prioritize local data access and minimize reliance on network connectivity. Utilize background synchronization processes to reconcile data changes between the client and server when connectivity is restored.

NON-Functional Requirement

1. SECURITY

Define Requirements Precisely: Security requirements include secure storage and encryption of biometric data, strong authentication mechanisms, compliance with data protection regulations, and regular security audits.

Impact Analysis: implementing strong security measures may impact system performance due to additional processing requirements for encryption and authentication.

Resolve Conflicts: while ensuring fast response times for attendance registration is essential for performance, it should not compromise security measures such as encryption and authentication.

Feasibility: While some requirements may pose technical challenges, advancements in biometric technology and software development frameworks can help address them.

Implementation: Implement end-to-end encryption for biometric data transmission and storage. Utilize strong authentication mechanisms, such as multi-factor authentication, to prevent unauthorized access. Adhere to data protection regulations (e.g., GDPR, HIPAA) and industry standards for biometric data security.

2. PERFORMACE

Define Requirements Precisely: Performance requirements focus on fast response times for attendance registration and efficient processing of biometric data.

Impact Analysis:

Resolve Conflicts: while ensuring fast response times for attendance registration is essential for performance, it should not compromise security measures such as encryption and authentication.

Feasibility: The feasibility of implementing each requirement depends on factors such as available technology, expertise, budget, and timeline

Implementation: Optimize system performance by implementing efficient data processing algorithms and database query optimization techniques. Utilize caching mechanisms to reduce latency and improve response times. Conduct load testing to identify performance bottlenecks and scale infrastructure resources as needed.

3. USABILITY

Define Requirements Precisely: Usability requirements cover intuitive user interface design, accessibility features for users with disabilities, compatibility across devices, and continuous improvement through user feedback.

Impact Analysis: Usability enhancements, such as accessibility features, may require additional development effort but can significantly improve user experience and inclusivity.

Resolve Conflicts: Identify any conflicts or contradictions between requirements and address them accordingly.

Feasibility: Feasibility analysis should also consider the availability of relevant technologies, expertise, and potential risks or challenges that may arise during implementation.

Implementation: Design intuitive user interfaces with clear navigation and visual cues to guide users through the attendance tracking process. Implement accessibility features, such as screen reader support and keyboard navigation, to accommodate users with disabilities. Gather user feedback through usability testing and incorporate iterative improvements based on user preferences.

4. SCALABILITY

Define Requirements Precisely: The scalability requirement emphasizes the ability to handle increased users and data without performance degradation, requiring both vertical and horizontal scaling capabilities.

Impact Analysis: Scalability features, such as horizontal scaling, may require architectural adjustments and additional infrastructure resources, impacting system design and cost.

Resolve Conflicts: There are no significant conflicts apparent among the requirements. However, trade-offs may need to be made in resource allocation to balance competing priorities, such as scalability versus cost or security versus performance.

Feasibility: ensuring compatibility with a wide range of devices may require additional development effort and testing to achieve.

Implementation: Design the system architecture with scalability in mind, utilizing distributed computing techniques and horizontal scaling strategies. Implement auto-scaling policies to dynamically adjust resource allocation based on demand. Monitor system performance metrics and scale infrastructure resources proactively to handle increased user loads.

5. TRANSPARENCY

Define Requirements Precisely: Transparency requirements entail clear explanations of data handling processes, user access to attendance records, and proactive communication of system maintenance.

Impact Analysis: Transparency and usability requirements contribute to user trust and satisfaction, making them high-priority as well.

Resolve Conflicts: Each requirement should be carefully evaluated to ensure they complement each other and collectively contribute to the overall effectiveness of the system.

Feasibility: Continuous monitoring and evaluation of feasibility throughout the development process are crucial to ensure successful implementation of all requirements.

Implementation: Provide users with clear explanations of data handling processes and privacy policies through accessible documentation and user guides. Implement audit trails and logging mechanisms to track user actions and system events transparently. Foster open communication channels for users to inquire about system maintenance and data security practices.

Conclusion

This report has outlined a structured and systematic approach to requirements analysis for the development of an attendance tracking system. By identifying key stakeholders, capturing their requirements, and modeling them into functional and non-functional specifications, the report provides a clear roadmap for system development. Through prioritization, conflict resolution, and feasibility analysis, potential challenges are addressed, ensuring the alignment of the final solution with business objectives and technical constraints.

Furthermore, the emphasis on transparency, usability, scalability, and security underscores the importance of delivering a reliable and user-friendly system that can adapt to changing needs and environments. By engaging stakeholders throughout the requirements analysis process and incorporating their feedback, the report promotes collaboration and ownership, leading to the development of a solution that meets stakeholders' expectations.

References

https://www.visual-paradigm.com/guide/requirements-gathering/requirement-analysis-techniques/

https://www.smartsheet.com/content/requirements-analysis

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

https://www.simplilearn.com/what-is-requirement-analysis-article