FACULTY OF TECHNICAL SCIENCES

STUDENT ATTENDANCE MANAGEMENT SYSTEM
CASE STUDY: CENTRAL UNIVERSITY MILE-91

BY
IBRAHIM ALUSINE KAMARA
MBALU SESAY
ABDUL KPAKRA FABA

A DISSERTATION SUBMITTED IN PARTIAL
FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF BACHELOR OF SCIENCE (HONS)
IN BUSINESS INFORMATION
TECHNOLOGY

JANUARY 2023

PERMISSION TO CARRY OUT PROJECT

As project supervisor, I, David Sapunka Fornah, hereby grant permission to Mbalu Sesay, Ibrahim Alusine Kamara, and Abdul Kpakra Faba to undertake their proposed project "Student Attendance Management System" as part of fulfilling their Bachelor of Science in Business Information Technology degree within Central University's Faculty of Technical Sciences.

I have reviewed the comprehensive project proposal outlining the key objectives, methodology, and intended outcomes. I approve of their planned approach and feasibility, and will provide supervision to enable successful, ethical completion within the designated timeframe.

My oversight will guide the students in addressing obstacles arising during the project's execution. I will ensure adherence to all relevant university policies and regulations throughout the research and development process.

Having affirmed the capabilities and commitment of the project team, I am confident in their potential to effectively apply themselves and produce valuable results. I look forward to supporting this undertaking and witnessing the positive impacts of their innovative efforts.

Supervisor's Name:

David Sapunka Fornah

Date: ………………………...

DECLARATION

We hereby declare that except for the work of other researchers which has been duly referenced, this submission is the original result of our own research undertaken under the supervision of Mr. David Sapunka Fornah. This submission has not been presented in whole or in part for the award of any other degree or diploma at any other university or institution of higher learning.

Department: Business Information Technology

Name	IDs	Signature
Ibrahim Alusine Kamara	1920227	……………………………..
Mbalu Sesay	1920233	……………………………..
Abdul Kpakra Faba	1920012	……………………………..

This project has been submitted for examination with my approval as the university.

Name	Signature
David Sapunka Fornah	……………………………………

ACKNOWLEDGEMENT

First, we express heartfelt gratitude towards God Almighty for blessing us with the strength, wisdom, and favor required to successfully complete our studies. It is by His grace that we have accomplished this milestone. We offer our sincere praises and thanks to Him.

We deeply appreciate the invaluable guidance from our supervisor, Mr. David Sapunka Fornah, whose patience, insights, and encouragement empowered us through every phase of this research endeavor. It has been an absolute privilege to learn under his direction.

We also sincerely thank all faculty members of Central University's Department of Technical Sciences who have profoundly shaped our academic journey by imparting invaluable knowledge throughout the years.

Additionally, we are incredibly grateful for the unwavering support of our friends who continuously motivated us. We extend our appreciation to all individuals who contributed towards the fulfillment of this project in ways big and small.

As lead researchers, we take full accountability for any oversights or errors present in this work. We pray God grants His favors upon everyone who facilitated our academic accomplishments.

DEDICATION

We would like to extend our heartfelt dedication in this dissertation to GOD Almighty, our parents, and the Central University Administration, unwavering pillars of support who have consistently stood by our side, irrespective of the challenges faced. Your unwavering love and assistance have served as the bedrock upon which this journey was built.

Our sincere appreciation goes to our mentors and educators, whose guidance and nurturing have fostered our personal and intellectual growth. You have kindled the flames of curiosity within us and steered us towards the pursuit of knowledge.

To our cherished friends and dear ones, who have been a constant source of encouragement and strength throughout the ups and downs. Your empathy, motivation, and shared laughter have provided solace during the demanding phases of this endeavor.

We also pay tribute to the scholars and researchers whose groundbreaking contributions have both inspired and enriched our own exploration. Your work has broadened our horizons, unveiled fresh perspectives, and reshaped our worldview.

Our gratitude extends to the countless hours spent in the relentless pursuit of understanding, encompassing every triumph and setback that has molded our development. These experiences have collectively contributed to our present selves.

Lastly, we dedicate this dissertation to the limitless potential of the future, hoping that our work may, in some modest manner, contribute to the advancement of knowledge and the betterment of society.

This dissertation is dedicated to each one of you. Without your presence, faith, and guidance, this expedition would not have been feasible. We extend our heartfelt thanks to you all for being our guiding lights.

ABSTRACT

Learning institutions are swiftly implementing ICT into their operations to better the teaching and learning process and raise efficiency, accuracy, and overall performance. Class attendance is crucial within academic institutions to encourage punctuality, involvement, and improved accomplishment among students. As such, automation of attendance recording at educational institutions is gaining popularity internationally using mobile and biometric technology. With significant proliferation of smartphones, mobile platforms appear as an efficient option to implement attendance solutions especially among student populations. Equally, biometrics employ unique physiological markers to authenticate identify in a frictionless, easy, and secure way.

This pioneering research project centers on developing and mobilizing such an integrated attendance application at Central University in Mile 91, Sierra Leone—leveraging mobile devices and facial recognition capabilities to transform prevailing paper-based attendance records plagued by limitations like inconsistency and tedious document handling. The automated self-service method intends to build an improved learning environment based on class involvement linked to institutional goals around progress and results. This program will expose avenues for localized sustainable adoption while influencing countrywide academic planning as relates to strategic progress of the education sector into the digital age.

The study utilizes a numerous set of techniques such as the; qualitative ways to acquire insights, which was utilized to guide the system design requirements process utilizing the agile software development method. Resulting to the formation of a conceptual framework in the development and implementation of a mobile-based attendance management application system. Besides, it adds to knowledge in the area, this research was also able to fulfill the system software specification which fits the study demand.

This study endeavor is confined to the creation and implementation of a student attendance mobile-based application system that intends to handle student attendance procedures solely. In that respect, the research argues for the improvement of the system in feature by integrating student mark-sheet processing and any other relevant operations into the teaching and learning process.

Keywords—Mobile attendance, facial recognition, identification, attendance management.

TABLE OF CONTENTS

CHAPTER ONE

Introduction

CHAPTER TWO

Literature Review

CHAPTER THREE

Methodology

CHAPTER FOUR

System Analysis and Design

CHAPTER FIVE

System Implementation and Testing

CHAPTER SIX

Conclusion, Lessons Learnt

and Recommendations for Future Enhancement

List of Figures

DEFINITION OF TERMS

SAMS (Student Attendance Management System)

CUSL (Central University Sierra Leone)

Biometric: Refers to the measurement and analysis of unique physical or behavioral characteristics of an individual, such as fingerprints, iris patterns, or voice, for purposes of identification or authentication.

Face Recognition: Is a technology that identifies and verifies individuals by analyzing their facial features and comparing them to a database of known faces, often used for security and access control.

Superbase: Is a term that does not have a widely recognized standard definition in technology as of my last knowledge update in September 2021. It could refer to a database or data management system with advanced features, but without specific details, it's challenging to provide a precise definition.

PostgreSQL: Is a powerful, open-source relational database management system (RDBMS) known for its reliability and extensibility. It is often used for data storage and retrieval in various applications and supports advanced features like transactions and complex queries.

FlutterFlow: Is a visual development platform for creating mobile applications using Google's Flutter framework. It allows developers to design, prototype, and build mobile apps using a visual interface, simplifying the app development process.

MobileFaceNet: Is a compact and efficient deep learning model specifically designed for face recognition tasks on mobile devices, known for its ability to perform real-time facial recognition with limited computational resources while maintaining high accuracy.

CHAPTER ONE

Introduction

1.1 Chapter Summary

This chapter provides a comprehensive introduction to the research project, highlighting the crucial link between lecture attendance and student academic performance. It defines the project's objectives, research questions, underscoring the significance of enhancing efficiency, accuracy, and accountability in attendance management. Finally, it outlines the scope and limitations of the proposed system, highlighting its potential delimitations related to geographical scope of the research and technology.

1.2 Background

In the era of digital transformation, educational institutions are increasingly seeking innovative ways to improve their administrative processes and enhance the overall learning experience for staff and students using information and communication technology which is used as a crucial tool to leverage numerous advantages in keeping academic institution up and running well. The digital transformation of education through ICT is crucial for 21st-century needs. It enhances accessibility, personalizes learning, fosters engagement, nurtures critical thinking, and readies students for the future workforce (UNESCO IITE & Microsoft, 2016).

Central University Sierra Leone located in Tonkolili District was established in 2014. It is a prominent higher education institution known for its commitment to academic excellence and innovation. With a diverse student body and a wide range of programs. The university continually seeks ways to enhance its operational efficiency and provide a modern technologically driven learning environment. Like many other universities in the world, Central University Sierra Leone face with similar challenges of effectively and efficiently managing student attendance records. Since its inception, the university embarks on the traditional attendance-taking methods, that utilizes manual roll calls which are time-consuming and susceptible to inaccuracies, making it imperative for the institution to adopt a more advanced approach.

In that respect, this research work is geared towards developing and deploying a Face Recognition Mobile-based Student Attendance Management System that will leverage the challenges faced by the institution enabling administration to capture, process and produce student records in an effective and efficient manner using a robust attendance information system application.

1.3 Problem Statement

The staff and management of Central University are challenged in tracking student class attendance. Hence, this results to difficulties in properly managing student statistics and assessment. At present, the existing manual attendance marking system is plagued by issues of being time-consuming, error-prone, and liable to fraud. The traditional paper-based registers employed for manual attendance tracking are inefficient, error-prone, and lack the ability to verify students' identities, leading to potential instances of proxy attendance and compromising the reliability of attendance data.

Several automated attendance systems (such as Microsoft Excel/desktop-based approaches, Google Classroom Attendance,) have been utilized to capture and process student attendance. However, these strategies are not adequate in addressing the issue because, it takes much of human involvement by requiring both the lecturer and student which consume a considerable amount of a lecture period.

In response to these challenges, the researchers sought to develop and deploy a mobile attendance system incorporating facial recognition technology. This system aims to offer a more precise, convenient, and cost-effective method for tracking attendance.

1.4 Aim and Objectives

The aim of the proposed project is to develop and implement a Student Attendance System with Facial Recognition Feature in Universities. The system aims to revolutionize the attendance tracking process by leveraging facial recognition technology and mobile devices, thereby enhancing efficiency, accuracy, and security in attendance management.

1.5 Objectives

I. To design and develop a user-friendly mobile application that can track student attendance using facial recognition functionality.

II. To identify strategies and procedures in the development of a mobile-based student attendance information system architecture design, implementation, and deployment.

III. To examine data accusation strategies, compliance, and report generation

1.6 Significance of the project

Face recognition mobile-based "Student Attendance Management System" holds significant implications for educational institutions and beyond. By harnessing the power of facial recognition technology, this system will offer a modern and efficient solution to track student attendance. Its significance lies in its ability to streamline the attendance recording process, reducing administrative burdens on educators and institutions.

Moreover, it enhances accuracy, eliminating the potential for manual errors and proxy attendance, ultimately promoting a fair and transparent record-keeping system. This innovation not only optimizes time and resources but also provides valuable data for analyzing attendance patterns, which can inform interventions to improve student engagement and retention.

Furthermore, it aligns with the broader technological trends in education, fostering digital literacy and adapting to the changing landscape of learning. In sum, this project will represent a transformative step towards more efficient attendance management and data-driven decision-making in educational settings, offering benefits in terms of accountability, resource optimization, and improved student outcomes.

1.7 Scope and Limitations

The study focuses on biometric face recognition of attendance at Central University Sierra Leone, using MobileFaceNet model. However, it may not fully capture the diversity and unique challenges in other geographical areas or educational institutions. The development feedback for the application system testing is collected through interviews and observations, with participants including administrators, lecturers, and university students. The project-programming environment includes a web browser, operating system, native mobile application, and a database (Supabase uses PostgreSQL). Users need to register to use the system, and administrators perform administrative functions by checking and granting functionalities to users.

The study on facial recognition attendance may face challenges due to budget constraints, time limitations, and data availability and quality. High costs associated with creating and implementing the system may limit its scope and testing. Additionally, the study's efficacy may be compromised by the lack of comprehensive and diverse training datasets, potentially impacting the system's accuracy and generalizability.

1.8 Chapter Demarcation

Chapter 2: Literature Review: Examines existing literature on student attendance system, covering historical developments, global and regional practices, and accuracy security concerns. It effectively synthesizes information from various sources, highlighting the relevance of mobile-based face recognition attendance systems while acknowledging the challenges and limitations associated with their implementation. The literature review discusses the methodologies, technologies, and theoretical frameworks used in prior research, and it identifies research gaps that your project aims to address.

Chapter 3: Methodology: Outlines the systematic approach employed in this study to address student attendance challenges. It offers a unique and integrated methodology that combines both qualitative research methods and software development practices, utilizing the Agile Methodology to ensure flexibility and responsiveness during the project's lifecycle. Additionally, the chapter highlights the crucial role of the MobileFaceNet model in training the data for facial recognition, underlining the significance of this hybrid approach in creating an effective and adaptive student attendance solution.

Chapter 4: Evaluation and Findings: Delves into the practical implementation and assessment of the mobile-based face recognition attendance app developed in this project. It presents the results of rigorous testing and evaluation, showcasing the app's performance in real-world scenarios. This chapter meticulously analyzes the accuracy of face recognition, app usability, and system reliability, offering valuable insights into the app's strengths and limitations. The findings provide a comprehensive view of the app's effectiveness in automating attendance management and serve as a basis for drawing conclusions and making recommendations for future improvements.

Chapter 5: Discussion: Engages in a critical examination of the findings presented in the preceding chapters, synthesizing them to draw meaningful insights and implications. This chapter provides a platform for in-depth analysis, addressing the significance of the project's outcomes in the context of face recognition attendance apps. It explores the alignment of results with research objectives, identifies any unexpected outcomes or limitations, and discusses the practical implications and contributions of the app's development. Additionally, Chapter 5 offers a space for comparing the findings with existing literature and theories, enabling a broader understanding of the project's impact and potential avenues for future research and application.

Chapter 6: Conclusion and Recommendations: Provides a concise yet comprehensive synthesis of the research findings, emphasizing the critical need to address class attendance challenges within educational contexts. The conclusion segment adeptly recaps the primary research outcomes, underscoring the app's significance in improving attendance practices by streamlining processes and enhancing efficiency. In the recommendations section, actionable insights are presented, including the refinement of user interfaces, bolstering data security measures, and offering training to instructors and students for seamless integration. Furthermore, the chapter advocates for future research initiatives, such as exploring adaptive attendance policies and conducting longitudinal studies, to assess the app's long-term impact and advance the field of attendance management.

CHAPTER TWO

Literature Review

Chapter Summary

The literature review serves as a comprehensive overview of existing research, scholarly articles, and relevant literature that explores the topics of face recognition and student attendance management. This section establishes the theoretical framework and contextual background for the research, highlighting key concepts, theories, and empirical studies related to the student attendance management system and its associated functionalities. It discusses the significance of understanding the current state of research in order to identify gaps and contribute to the existing body of knowledge.

2.1 Introduction

According to Strickley (2020), student attendance is "an act of recording whether students are present or absent from school or class," which helps educational institutions keep tabs on how often their students participate in class activities. As such, it functions as a gauge for tracking the development of students, identifying problems, and ensuring that students receive the necessary teaching in compliance with rules pertaining to compulsory education.

There are several methods available for measuring attendance, ranging from manual paper-based logs to automated biometric checks, depending on the size, resources, and preferences of the school. At the start of class, roll calls are frequently used to record attendance and identify absentees. To verify student safety, several institutions additionally monitor attendance at the end of the school day.

Strickley (2020) put forward “the institutional significance of student attendance is supported by several arguments. First, it makes it possible to track students' academic progress and identify those who are failing or who are missing important material and need extra help. Second, persistent absences may be a sign of bullying, truancy, or dropout risks, which would allow school administrators to take preventative action. In the end, attendance at school signifies participation in required learning, to which children are legally entitled”. Therefore, keeping track of attendance functions as a diagnostic tool for identifying more general student issues as well as an indicator of academic achievement.

Sawhney et al. (2019) said “to verify the student attendance record, the personnel staff ought to have an appropriate system for approving and maintaining the attendance record consistently. By and large, there are two kinds of student attendance framework, i.e. Manual Attendance System (MAS) and Automated Attendance System (AAS). Practically in MAS, the staff may have trouble in both approving and keeping up every student's record in a classroom all the time. Strickley (2020) goes on saying “in a classroom with a high teacher-to-student ratio, it turns into an extremely dreary and tedious process to mark the attendance physically and cumulative attendance of each student”.

Consequently, we can execute a viable framework which will mark the attendance of students automatically via face recognition. AAS may decrease the managerial work of its staff. Especially for an attendance system which embraces Human Face Recognition (HFR), it normally includes the students' facial images captured when clock in within the classroom.

2.2 Framing Mobile Attendance Tracker Dynamics

Effective design and operation of a mobile-based attendance tracking solution needs addressing important connected components - the user-facing application interface, underlying organizational system architecture as well as data accrual and monitoring regulations. The research on maximizing efficiency, adoption and integrity of such systems suggests models including these essential variables and related dynamics for a comprehensive approach.

The following conceptual framework graphically summarizes the essential aspects needed in designing an integrated attendance management solution.

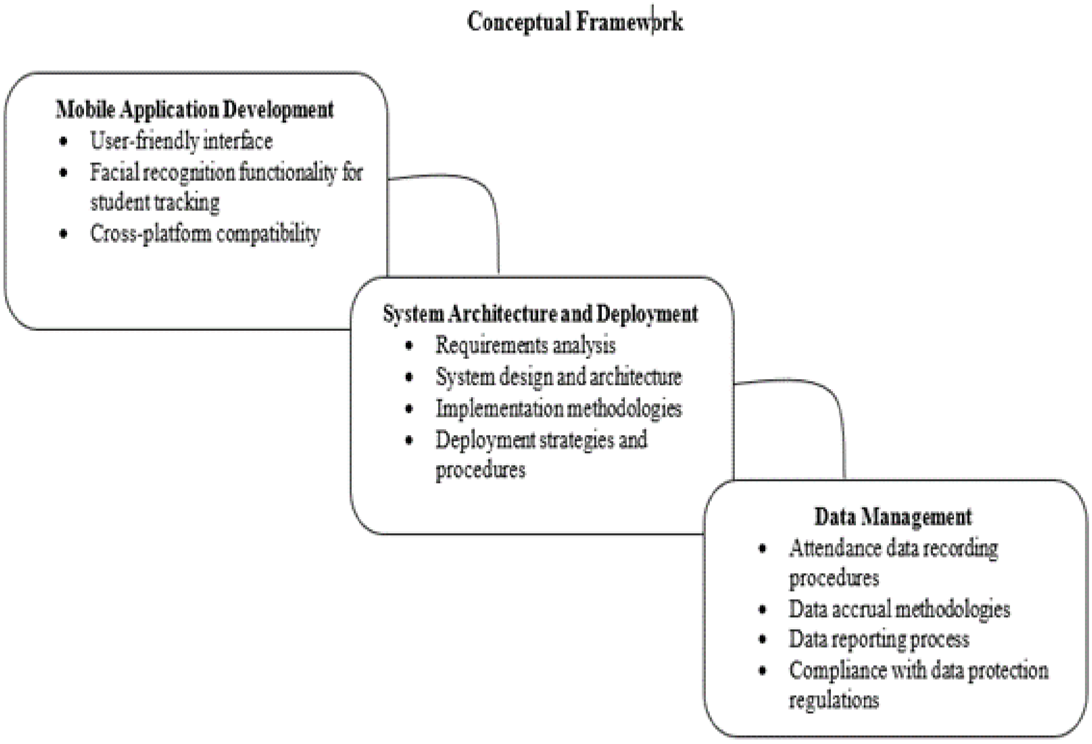

Conceptual Framework
Mobile Application Development
User-friendly interface
Facial recognition functionality for student tracking
Cross-platform compatibility
System Architecture and Deployment
Requirements analysis
System design and architecture
Implementation methodologies
Deployment strategies and procedures
Data Management
Attendance data recording procedures
Data accrual methodologies
Data reporting process
Compliance with data protection regulations

***Figure 2. 1** Displaying the three essential structures of developing a student Attendance Management System*
***Source:** Study activity (November 2023).*

As outlined out, major focal areas cover (1) User-Centric Application Design (2) Process-Focused System Architecture and (3) Compliance-Driven Data Management Procedures. Of key focus is promoting voluntary adoption among end-users - the students - through intuitive interfaces and capabilities meeting functional demands surrounding attendance processes. Structural design around integration with existing institutional structures and regulations is equally vital for a sustainable solution ecosystem. Rigorous rules preserving data integrity and stakeholder rights generate greater trust and acceptance.

The paradigm presumes synergistic links between excellence in human-centered design thinking, system-level architectural alignment and data governance rules accordingly that ultimately generate optimal usage and effect.

An examination of literature across three areas - user experience principles, technological design approaches and data compliance standards - will reveal essential dynamics and concerns in designing an efficient mobile attendance tracker for academic organizations. Patterns verifying or strengthening such a multi-factor view will guide a complete path ahead.

2.3 Design And Develop a User-Friendly Mobile Application That Can Track Student Attendance Using Facial Recognition Functionality.

Education establishments are shifting from manual or card-based methods to mobile applications in the digital age to improve attendance tracking. With the widespread use of cellphones and innovations like face recognition, attendance monitoring has never been easier, more precise, or safer. However, creating such apps is a complex task in contemporary academic environments as it necessitates striking a balance between privacy concerns and user-friendliness.

(Azhaguraj et al., 2022) The paper presents an automated attendance management system for educational settings, leveraging facial recognition technology to streamline the process. Utilizing OpenCV for real-time face detection and the LBPH algorithm for recognition, the system captures images of individuals in the classroom and matches them against a pre-built dataset to mark attendance. This innovative approach aims to replace traditional, time-consuming attendance methods, incorporating an automated notification system that alerts parents when students are marked absent. The system is built on a general-purpose programming language and is designed to function without human intervention, illustrating the integration of machine learning algorithms in practical applications.

2.3.1 User-Friendly Interface

A driving force behind adoption for any new system are design factors that create accessibility, convenience, and purpose for end-users. The application interface constitutes the critical front-facing component interfacing directly with students - the primary stakeholders - in attendance tracking.

Literature regularly underlines the significance of human-centered design in encouraging voluntary uptake of innovations (Yang et al., 2018). Simplicity, intuitiveness, and ease-of-use promote general satisfaction and continuing usage intention (Goel et al., 2017). Key components include minimum data inputs, process openness, and comprehensive dashboard displays enhance self-serviceability. Personalization features meet individual tastes and demands. Gamification fosters engagement through goal-setting and micro-incentives.

Such user-friendly design features involve the application's learnability, memorability, efficiency, and subjective appeal during attendance self-marking. This further impact prolonged usage habit over time. Ergonomic principles require evaluating target user demographics, tasks, and settings (Gasmi et al., 2017). Customizations consequently provide flexibility across contexts. Overall, literature recommends user-centricity - by offering clear processes, ease, choice, control, and motivation - as the forefront issue while creating attendance management interfaces and interactions. Adoption outcomes rely on UX optimality.

2.3.2 Facial Recognition Functionality

This subsection delves deeper into the concept of face recognition in the context of student attendance management. Over the years, facial recognition technologies have seen significant advancements, with applications spanning various sectors, from security to health and education. The incorporation of facial recognition in educational tools, especially for attendance monitoring, ensures both speed and accuracy.

Li et al. (2020). Face recognition technology is a biometric technology, which is based on the identification of facial features of a person. People collect the face images, and the recognition equipment automatically processes the images.

(Hashem Al-Ghaili, 2019) Facial recognition technology has swiftly evolved from a sci-fi concept to an everyday reality, now integral to devices and systems worldwide—from smartphones using it for unlocking screens to governments employing it for identification purposes. It intricately analyzes a multitude of facial features to distinguish identities, tackling challenges like aging and viewpoint variations through sophisticated techniques, such as 3D modeling. With its accuracy reaching levels where it can differentiate even identical twins, facial recognition is being adopted by critical sectors like banking for security and by law enforcement for database creation. As this technology becomes increasingly embedded in our daily activities, understanding its workings and implications is as crucial as the technology's ability to learn about us.

(Wagh et al., 2015) discussed various face recognition techniques like Principle Component Analysis (PCA), Eigenface, Support Vector Machines (SVM) and Neural Networks and compared them based on their success rate. The authors also wrote about system architecture, step by-step methodology and supported it with its algorithm. They have also provided a mathematical model using mathematical concepts and language.

Xiao et al. (2018) presents an optimization of the MobileFaceNet face recognition network for deployment in embedded environments, addressing the challenges posed by deep convolutional neural networks in such settings. The optimization involves reducing the number of layers in MobileFaceNet to decrease model size, substituting PReLU with the h-

ReLU6 activation function, and incorporating an efficient channel attention module to enhance feature channel significance. As a result, the optimized MobileFaceNet's size is reduced to 3.4 MB from 4.9 MB, achieving mean Average Precision (mAP) scores of 98.52%, 97.54%, and 91.33% on LFW, VGGFace2, and a custom database, respectively, with a recognition time of approximately 85 ms/photo. These improvements demonstrate a successful balance between reducing model complexity and maintaining high performance in face recognition tasks.

2.3.3 Natively Cross-Platform Compatibility

Literature advocates cross-platform approaches for mobile development allowing writing codebase once and deploying across multiple platforms (Zohud & Zein, 2021). React Native, Flutter, Xamarin and other frameworks now facilitate this model through asynchronous JavaScript and Dart, C#, respectively.

These bridge web development languages into native experiences on target operating systems leveraging performance of the platform SDKs and interface elements. The approach contrasts with hybrid web apps that run through browser engines. Cross-compilation hence replaces device-specific programming benefiting feature parity.

Here is a paragraph elaborating the advantages of using Flutter for cross-platform mobile app development:

Flutter stands out as an optimal cross-platform framework given its architecture enabling compilation of Dart code into platform-specialized native machine code rather than interpreted JavaScript bridges. This ensures consistently high performance aligned with native experience expectations across iOS and Android devices (Wikipedia contributors, 2024). Flutter's rich widget set facilitates rapid prototyping while "hot reload" features accelerate iterative development cycles allowing tweaking visuals without full recompilation. By leveraging specifics of underlying platform SDKs, Flutter allows over 90% business logic reuse with interface customization flexibility where needed.

The framework thereby balances productivity gains from code sharing with ability to tweak for UI/UX parity when required. The functional declarative composition of visual components also makes Flutter apt for complex informational apps with dynamic data. In essence, Flutter's native compilation, dev velocity and adaptive UI stack makes it a strategic choice for mobile developers aiming for reusable yet customizable logic across mobile platforms. Literature affirms businesses gain significantly by the framework's unique ability to balance consistency, personalization, and speed.

As the student attendance application uses Dart, the framework's capacity to render native-styled UIs provides flexibility for platform-specific customizations while reusing business logic across iOS and Android. Target devices access platform-specialized UI modules for optimized performance retaining over 90% code reuse (Pinto & Coutinho, 2018). Impact analysis suggests architectural modifications rather than overhaul for new versions. React Native's declarative nature also eases maintainability and extensibility.

2.4 Identify Strategies and Procedures in The Development of A Mobile-Based Student Attendance Information System Architecture Design, Implementation, And Deployment.

The usage of mobile and biometric technology is transforming student attendance tracking in academic institutions. However, adopting such systems takes careful preparation and a multidimensional strategy. This study evaluates research on techniques and procedural recommendations for designing interoperable, scalable, and user-centric mobile attendance trackers. It highlights the necessity for modular system design, agile development approaches, and change management strategies. The study underlines the significance of holistic design, including technology, processes, and people for integrated platforms that can endure beyond first deployment.

2.4.1 System Architecture and Deployment

Beyond the outward-facing application interface, an equally crucial feature incorporates structural support systems providing mainstream accessibility, interaction with existing databases, and policy alignment for a unified attendance solution.

Technical literature recognizes that successful adoption of innovations relies substantially on organizational-level planning and methods for assimilation within business ecosystems (Lano & Tehrani, 2023). Attendance tracker deployment consequently needs designing backend interfaces to institutional databases identifying students, courses and schedules.

Seamless data transfers between these sources and the tracker provide reliable identity verification and contextual mapping for accurate attendance records. Further, access protocols, storage structures and functional regulations need simplification for compliance and optimal use by stakeholders.

Literature recommends modular architectures providing integration with external academic systems through APIs as well as scalability to accommodate escalating use demands (Alatalo et al., 2007). Cloud-based attendance data storage offers security precautions and facilitates bespoke analytics. Finally, role-based access models prevent unwanted data breaches while providing transparency.

The framework above incorporates these essential concerns - open designs, implementation techniques, deployment strategies and database integrations creating the holistic attendance solution ecosystem students engage with for self-marking attendance. Ensuring structural soundness and policy congruence drives its sustained mainstreaming.

According to Patel and Patel (2023), mobile architecture is made up of a mix of structural elements and their unique set interface and the framework behavior of all structural elements. In simple words, mobile app architecture is the collection of principles that must be followed to design a totally working application. Although, according to industry norms, you might change the techniques and patterns applied to personalize as per the vendor's specifications.

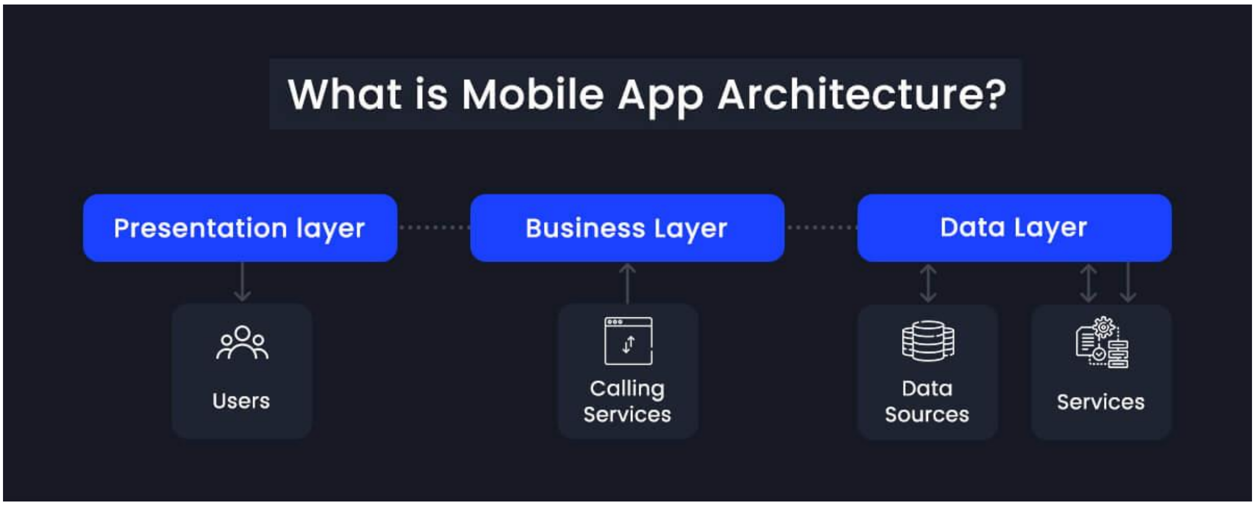


***Figure 2. 2** Definition of Mobile Application Architecture*
***Retrieved from**: https://aglowiditsolutions.com/blog/design-app-architecture*

Mobile apps commonly employ a three-tiered logical architecture spanning presentation, business logic and data storage layers that separate user interface aspects from functional components and information repositories - the topmost UI presentation layer handles visual components like menus and text that users interface with to tap or swipe, the middle business logic layer retrieves inputs to perform computational operations like calculating order values or validating credentials to drive app behavior, while the bottom data layer manages communication with database sources that supply essential information; the independence between the tiers processing tasks focused on display, logical procedures and persistence respectively allows easier configuration, troubleshooting and upgrades targeting each facet independently without entanglement enabling modular mobile application development management catering to role specialization across client-end visuals, server-side logics and backend data provision functionalities.

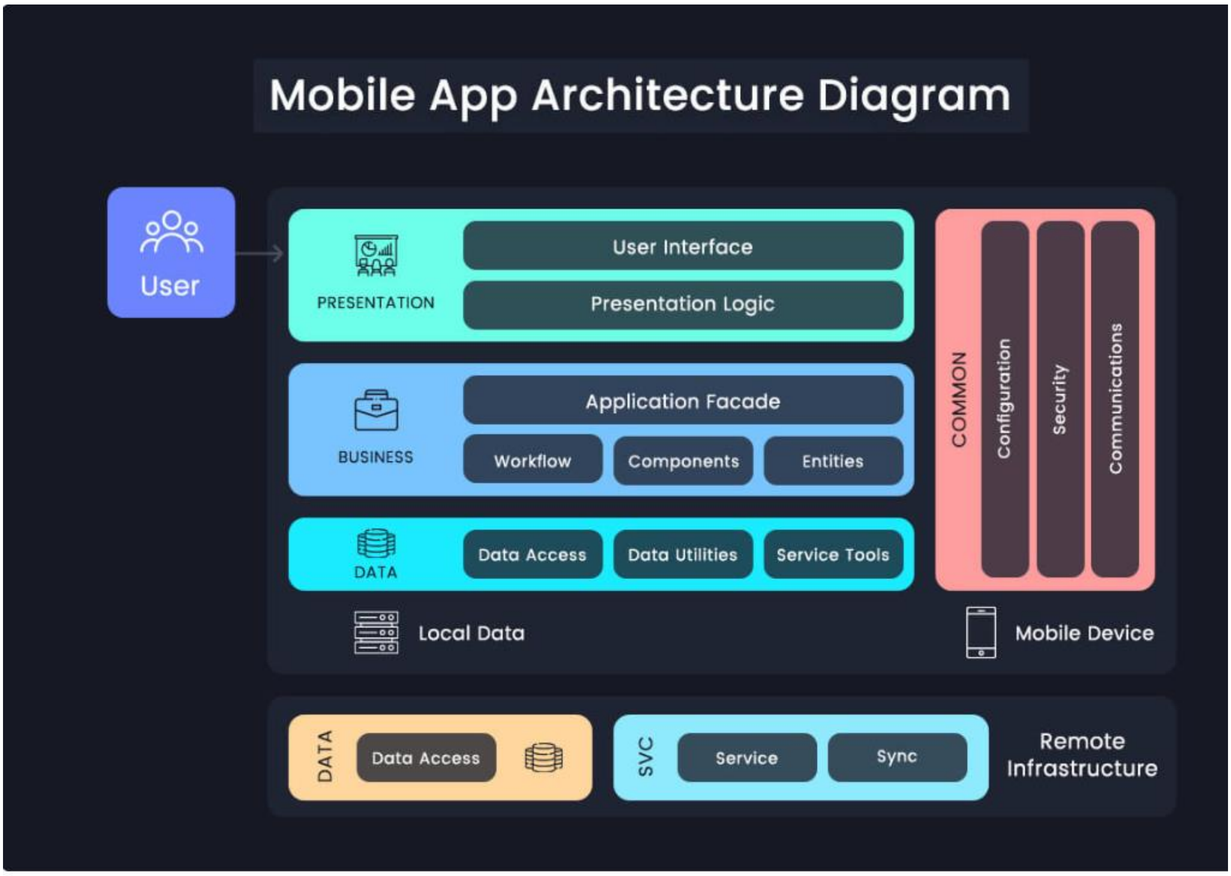


***Figure 2. 3** Mobile Application Architecture Diagram*

Retrieved from: *https://aglowiditsolutions.com/blog/design-app-architecture/*

When architecture is independent of external resources, then mobile app architecture is correct in the true sense. Furthermore, because of the independence, the efficiency of the mobile app will remain consistent irrespective of the mobile OS platform.

Mobile application architecture encompasses the structural layers, data flows and logical processes underpinning how apps function - spanning presentation aspects visible to users, business logic components executing solutions and data persistence models; while no uniform standards exist across platforms like iOS and Android, core considerations include target device types, network connectivity needs, UI/UX information flows catering to use cases and push notification expectations - additionally app architecture decisions depend on pre-set budgets, development timelines and requirements spanning features, platforms, compliance factors and testing needs; robust architecture ensures apps balance user experience with technical realizations as well as scale sustainably over the long term so upfront deliberations over multifaceted functional and non-functional parameters is key to devising integrated and adaptable mobile app solutions that perform reliably across customer lifecycles delivering value and managing costs Vaidya (2023) put forward.

2.4.1 Requirement Analysis

Requirements specification marks a pivotal initial stage guiding architecture and design choices during mobile application development. Beyond enumerating desired functionality, holistic analysis covers contextual dimensions surrounding prospective app utility, delivery, usage, and evolution (Vaidya, 2023b)

Thorough understanding of target user groups, their needs and environments allows appropriate scoping aligned to workflows, behaviors and device ecosystem aspects. Common techniques like user personas, journey maps and use case modeling facilitate exploring needs and priorities. Furthermore, institutional deployment scenarios impose technical constraints around integration, compliance, and performance requirements. Mapping cross-platform compatibility guides technological decision making.

With mobile platforms rapidly evolving, analysis also entails assessing emerging feature expectations, portability necessities and future extensibility needs. Defining test coverage and progress indicators provides quality goals supporting evaluation. Maintainability demands likewise require factoring ease of updating, optimizing and supporting apps over time.

In essence, comprehensive requirements gathering around user, functional, technical and managerial dimensions make for systematic mobile app development and sustainable deployment. It expands solution vision beyond features enumerated to encompass holistic fitness criteria that well-designed architecture aims to satisfy across individual apps and their embedding environment. The analysis hence sets the direction for downstream development activities.

2.4.2 Implementation Methodologies

Wagh et al. (2015) addressed numerous implementation strategies for attendance systems undergoing mobilization in complicated real-world scenarios. The authors blend planning with adaptability for solidifying visions based on institutional input rather than strict sequential methods.

The technique interleaves important tasks including requirements gathering, prototype iterations and interactive interactions allowing progressive absorption adapted to academic institutions. Short iterative release cycles encourage engagement, uncovering adoption constraints and anchoring innovations around community meaning building.

Continuous planning permits charting system roadmaps including policy evolutions influencing compliance demands. The techniques differ with isolated technology installations by incorporating attendance solutions within larger socio-technical transitions accounting for user, organizational and environmental adjustments throughout time. This approach views change facilitated by participatory problem inquiry. The literature supports techniques recognizing the interplay between balancing technological functions and human factors to overcome situational restrictions in practice for sustained institutionalization.

In essence, successful attendance system deployment relies on adaptive strategies that are continually calibrated through co-created studies of the academic environment and community.

2.4.3 Deployment Strategies and Procedures

According to information provided by Global App Testing (Spa Worldwide Limited), mobile app deployment is the process of making your app accessible to either an internal or external audience. In the case of internal deployment, the app is placed in a production environment for modification and assessment. For external deployment, it involves releasing the app to end-users.

In the case of native apps, which are directly installed on mobile devices, end-users can download them through dedicated app stores accessed from their devices. However, submitting a native app to these stores can be a complex process, requiring a thorough understanding of the submission procedures. For long-term success in app deployment, it's crucial to consider both customer appeal and basic functionality. Engaging in the deployment process is only meaningful if there is a demand for the app, emphasizing the importance of creating something that users want to utilize

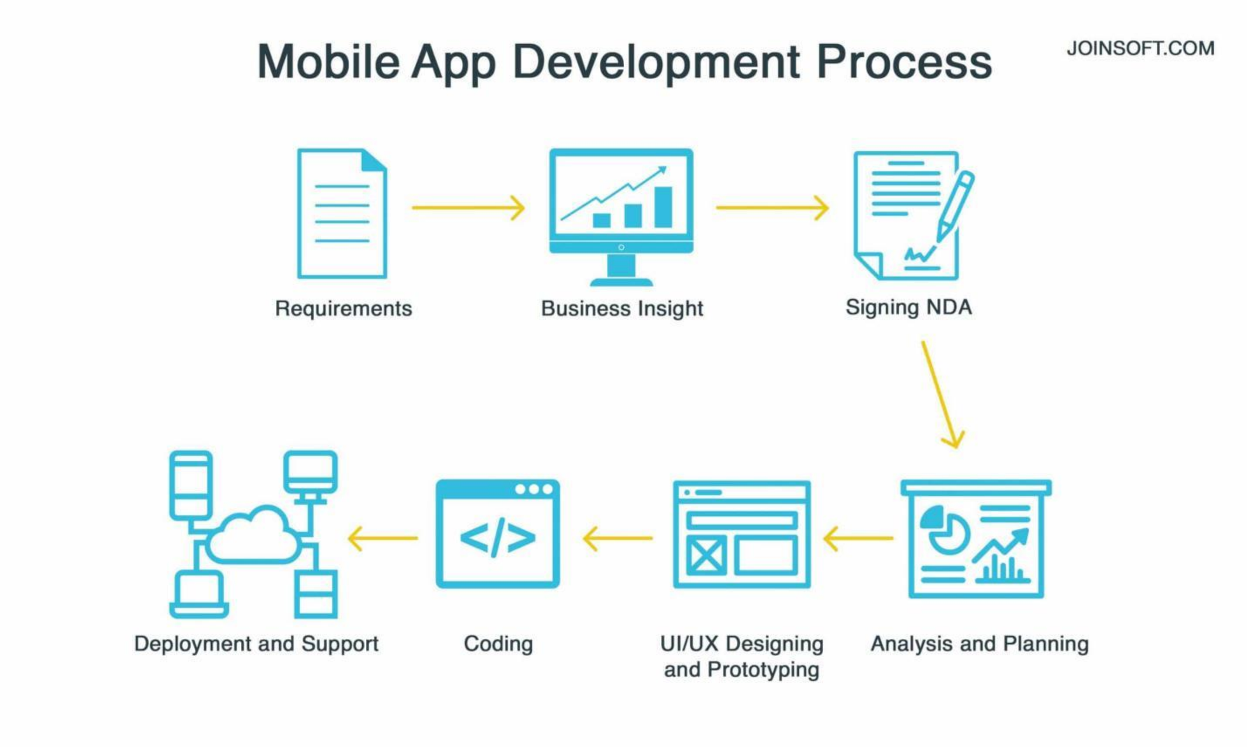


Figure 2. 4 Mobile Deployment Process
Source: *https://joinsoft.com/blog/mobile-app-development-process/*

According to Kolodiy (2023), modern standards for software development and user experience necessitate developers to update their projects continuously. Deployment and integration have become continuous processes – a contemporary application is delivered every day. Therefore, having a robust deployment strategy is now more critical than ever.

Today, systems are more flexible and scalable. They generally leverage Cloud infrastructure, serverless method, or AWS microservices to give a more adaptable capability. There are several development teams participating in the process, and such variety might make deployment challenging. In this post, you'll find out how to deploy an application utilizing the most feasible application deployment methodologies.

He further elaborated that a deployment strategy refers to the process of releasing an application into production. It involves a series of steps that ensure the application is correctly configured and ready to be used by end-users. The deployment process is an essential part of any software development lifecycle, as it determines how quickly and efficiently a new application or feature can be launched.

According to Kolodiy (2023), modern standards for software development and user experience necessitate developers to update their projects continuously. Deployment and

integration have become continuous processes – a contemporary application is delivered every day. This is why having a robust deployment strategy is now more critical than ever.

Today, systems are more flexible and scalable. They generally leverage Cloud infrastructure, serverless method, or AWS microservices to give a more adaptable capability. There are several development teams participating in the process, and such variety might make deployment challenging. In this post, you will find out how to deploy an application utilizing the most feasible application deployment methodologies.

Kolodiy (2023) further discussed, Deployment strategy refers to the process of releasing an application into production. It involves a series of steps that ensure the application is correctly configured and ready to be used by end-users. The deployment process is an essential part of any software development lifecycle, as it determines how quickly and efficiently a new application or feature can be launched.

Deployment approaches balance tradeoffs between continuity and agility including recreate strategies that sequentially phase out legacy versions for new updates enabling clean transitions while rolling techniques simultaneously run multiple editions widening test coverage through segmented canary launches - such multifaceted deployment pathways combine change management safeguards and efficiency gains whether replicating environments through blue/green differentials, validating incremental via canaries or shadow launching editions in tandem; in essence, literature prescribes holistic deployment cycles allowing teams to choreograph and assess the interplay of old and new modules with flexibility for rollback while expediting iteration through compartmentalized testing for modular upgrades that limit ripple effects, thereby enabling both uninterrupted services and accelerated innovation.

2.5 Examine Data Accusation Strategies, Compliance, And Report Generation.

Student attendance data represents fundamental functional input for academic choices and downstream operations surrounding grading, advancement, resource allocation etc. However, efficient data accrual needs stringency across instrument dependability, inspection and transparency dimensions sustaining ethical integrity.

This needs proactive design - spanning attendance collection mechanisms, storage infrastructure specifications and access control models - that emphasize responsible data regulations protecting integrity and rights within solution ecosystems. Equally, derived attendance analytics require deliberate scoping suited to usage contexts balancing openness, privacy and value based on role entitlements.

In essence, scholarship increasingly recognizes data processes as pivotal mediators between attendance tracking functionalities and actual institutional impact; their design merits deliberate considerations around permissioned access, auditable trails and accountability mechanisms that foster stakeholder trust in the underlying reporting. By taking an integrated compliance approach, solution architects may consciously construct data pipelines elevating deployment results above limiting technical configurations alone.

This segment accordingly examines structural constraints and planning imperatives around designing ethical and useful attendance data apparatus - spanning capture mechanisms, security controls and analytical reporting capabilities - that collectively uphold credibility, safeguard rights, and reveal value from such systems.

According to Rasmussen (2022), "A data acquisition strategy lays out a clear plan for gathering new data in a way that meets your organization's strategic needs. It's a roadmap that guides staff in how to evaluate, acquire, store, and leverage that new information efficiently and effectively. A well-designed strategy should provide clarity on the specific roles and responsibilities of each person involved throughout the process".

An effective data acquisition strategy should address the following:

- **Needs Assessment:** Identifying the information required to satisfy the objectives of your project or use case, as well as your wider organizational goals.
  - **Auditing Existing Data:** Auditing your existing data to prevent duplicate purchases and find areas that require improvement.
  - **Data Evaluation:** Evaluating the costs, benefits, and quality of different data sources on the market and determining whether they meet your objectives.
  - **Data Privacy:** Planning how to protect personally identifiable information (PII) and address any data privacy issues throughout the data evaluation and acquisition process.
  - **Storage and Security:** Determining the location where raw and processed data will be stored (locally within your organization or in the cloud in the appropriate region) to ensure the security and accessibility of your data.

- **ETL and Data Integration:** Evaluating the steps required to engineer raw data into useful information, as well as any tools needed in this process.

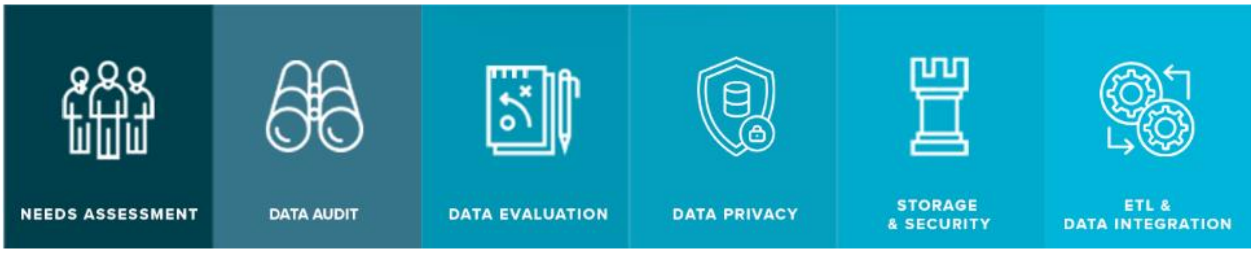


***Figure 2. 6** Components of an effective data acquisition strategy*

***Retrieved from:** https://urbanlogiq.com/how-to-write-an-effective-data-acquisition-strategy/*

2.5.1 Attendance data recording procedures

Attendance data recording processes in the context of a mobile app with biometric face recognition comprise the systematic process of acquiring and maintaining attendance information using facial recognition technology on mobile devices. The approach normally requires the installation and use of a specific mobile app equipped with biometric capabilities. Students utilize the mobile app to record their attendance by allowing the device's camera to collect and evaluate their face characteristics. Facial recognition technology processes the acquired image, establishing a unique biometric template connected to each student. The time and date of attendance are recorded, producing a historical history of attendance events.

Real-time updates allow fast tracking of attendance status, and the connection of the mobile app with academic systems facilitates smooth record-keeping and accessibility for administrators and instructors. Security measures, including encryption and safe data storage, are necessary to comply with privacy requirements and preserve the biometric data of individuals.

The mobile app may integrate user-friendly interfaces to increase the simplicity of attendance recording for students. Additionally, verification and validation checks are done to guarantee the authenticity of attendance data, validating the identification of pupils during the face recognition process. Exception handling techniques address specific cases, and automatic reporting systems provide thorough attendance records, offering useful insights for stakeholders. Robust audit trails track any alterations made to attendance data, guaranteeing openness and accountability inside the mobile app.

2.5.2 Data accrual methodologies

Literature discusses the growing role of learning analytics, driving decision making and exploring research possibilities based on student attendance data accrued from modern systems (Papamitsiou & Economides, 2022). This necessitates formulating rigorous accrual techniques upholding evidentiary quality for such secondary usage environments while preserving ethical bounds.

Federated learning methodologies facilitate consolidating activity patterns across apps by exchanging insights rather than raw attendance data (Ramachandran et al. 2022). This allows institutional modeling uncovering correlations while avoiding direct access to personal student data upholding privacy commitments. Access controlled cloud storage offers additional operational security provisions.

Data accrual itself is staged through tiered abstraction modeling isolating personal identifiers for transparency. Encryption and immutable audits preempt tampering risks. Overall, the multi-pronged strategies balance utility and compliance getting maximal value from attendance systems while future proofing for edge scenarios involving third-party analytics integration or migration. They emphasize ethical attendance data curation right from accrual mechanisms.

In essence, forward-thinking data consolidation protocols entail proactive risk mitigation planning through tiered governance frameworks that accommodate both present and emergent usage regimes - guiding attendance architecture design around trust and participation.

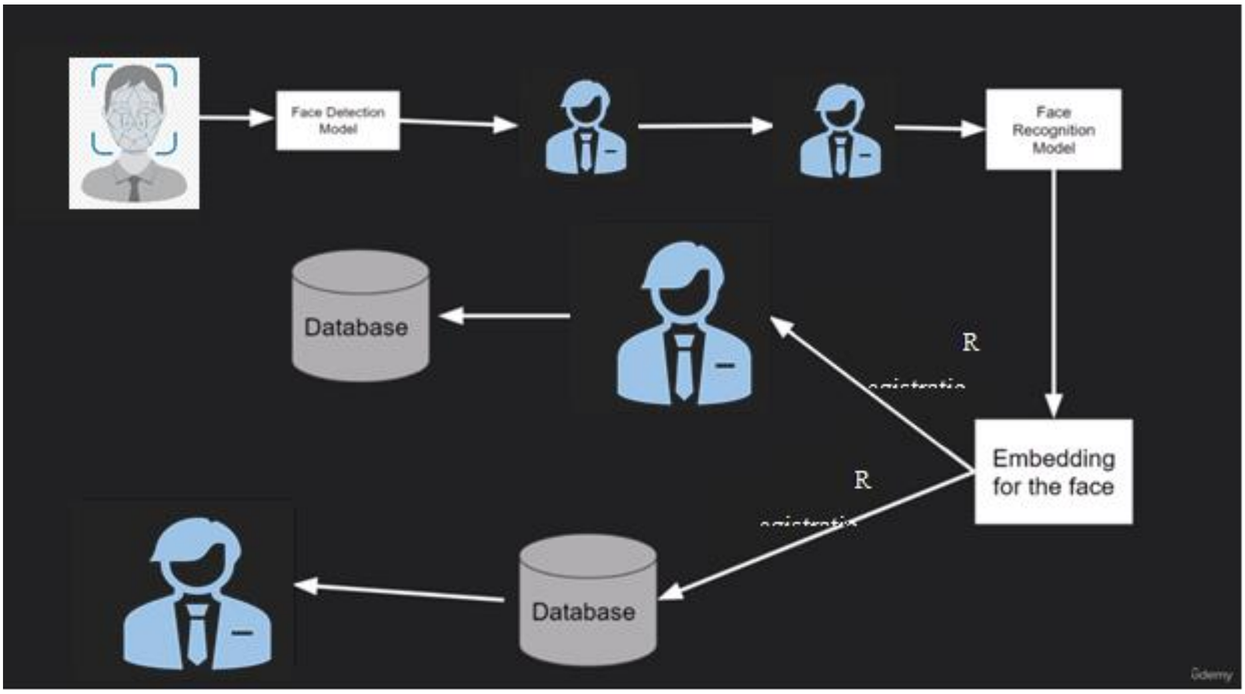


***Figure 2.** 7   Data Accrual Methodologies*

***Source:** Study material December (2023)*

In the data accrual process for facial recognition, the initial phase includes feeding a picture into a face detection model to recognize and locate faces within the image. This stage applies a machine learning model particularly built for face detection, creating bounding boxes around the discovered faces.

Subsequently, depending on the model's output, the recognized faces are separated by cropping the image around each bounding box. This acts as a strategy to focus on the face areas and isolate them from the surrounding background.

The following phase includes submitting these clipped faces to a face recognition algorithm to build face embeddings. Each cropped face picture undergoes processing by a face recognition model, which extracts a mathematical representation known as a face embedding. This embedding contains distinctive face traits, collecting details such as eye distance, nose shape, and jawline layout. For recognition purposes, the produced face embedding of the unknown face is compared with the embeddings of registered faces. The recognition technique comprises recognizing the recorded face with the least distance from the embedding of the unknown face. This comparison is important for selecting the most similar face in the database and subsequently recognizing the individual.

2.5.3 Data reporting process

Established and precise data recording techniques are in place, wherein a mobile application adeptly captures individual student attendance data during lessons through facial recognition. The application systematically associates and records biometric facial scan data, linking it with unique student IDs and essential course/class details for each attendance occurrence.

These detailed raw attendance transaction records from mobile devices undergo consistent aggregation using accrual procedures, facilitating the transfer of data batches to centralized databases and servers. The seamless flow of attendance data from various mobile sources to integrated back-end repositories is facilitated through appropriate connections and APIs.

With a substantial volume of recorded student attendance data amassed in the databases, stringent compliance measures are implemented to safeguard personal data. Subsequently, downstream reporting systems come into play, generating aggregated attendance metrics at class, teacher, and administration levels for analysis. Metrics such as attendance percentages, absence rates, and involvement frequencies are systematically tracked over time.

The user-friendly front-end of the mobile application ensures effortless attendance data collection at scale, while well-organized back-end data pipelines guarantee the reliability of reporting mechanisms and dashboards. These systematic processes transform low-level transaction data into meaningful insights, facilitating the effective utilization of the mobile-based system architecture for student attendance tracking in educational settings.

2.5.4 Compliance with data protection regulations

According to Babalola (2023), data governance refers to the comprehensive administration of (personal and non-personal) data to promote business goals. Data protection, on the other hand, principally controls the handling of personal data for the preservation of users' privacy and other basic rights and freedoms.

However, more data also implies more weaknesses and a wider surface area for attackers. According to IBM's Cost of a Data Breach, the global average cost of a data breach in 2023 was USD 4.45 million—a 15 percent rise over three years.

Data compliance helps reduce these dangers and keep client data safe. It creates a set of controls—or data compliance standards—that companies and people must follow while managing data. The goal of these compliance standards is to provide safeguards that preserve data privacy and prevent data misuse. Data compliance may also assist businesses and individuals build rules and processes to manage data better responsibly.

The General Data Protection Regulation (GDPR), as highlighted by IBM in their analysis of the Cost of a Data Breach in 2023, is a comprehensive data privacy framework implemented by the European Union (EU) to safeguard the personal information of its citizens. Specifically targeting personally identifiable information (PII), GDPR imposes stringent compliance requirements on data providers. It establishes a mandate for organizations, both within and outside Europe, to practice transparency in their data collection practices, thereby empowering individuals with greater control over their PII.

A distinctive feature of GDPR lies in its unwavering approach to non-compliance. The regulation imposes substantial fines on entities that fail to adhere to its privacy regulations and data compliance standards. These fines can potentially amount to up to 4 percent of an organization's annual global turnover or €20 million, depending on whichever is greater.

Given the severe consequences of non-compliance, GDPR has compelled businesses globally to reassess and refine their data collection and handling practices. This has underscored the paramount importance of implementing robust data security measures and maintaining strict adherence to compliance standards in the contemporary data-driven landscape.

In conclusion, this literature study has offered a complete analysis across three crucial areas for designing an efficient mobile-based student attendance management system - user-centric design concepts, technological architecture approaches, and data governance procedures. Research reveals that excellence across these interconnected features is vital for encouraging voluntary adoption, seamless integration and continued compliant usage of such tracking technologies. Specifically, intuitive mobile interfaces catering to student needs stimulate engagement, while modular system architecture facilitates integration with academic databases and policies maintaining continuous relevance. Similarly, ethical data accrual processes and access constraints sustain integrity for decision-making. While individual components maintain value, study increasingly recognizes the synergistic ties between human-centered, process-focused and compliance-driven design for attendance tracker viability. By addressing user experience, architecture, and data regulations concurrently through a multi-perspective method, producers can create holistic attendance solutions purposed for academic growth.

CHAPTER THREE

Methodology

3.1 Introduction

This chapter delineates the methodical process undertaken to develop the mobile-based face recognition attendance system for Central University. It outlines the developmental life cycle, techniques, tools, and frameworks leveraged at each phase to create an optimized and user-centric attendance solution.

3.2 Study Research Methodology

The definition of research methodology is a critical aspect of any scholarly inquiry, providing the framework through which knowledge is generated and validated. McCombes (2023) offers a comprehensive understanding, describing research methodology as the systematic approach employed to gather, analyze, and interpret data in a structured and rigorous manner. This definition emphasizes the importance of a well-defined process, highlighting the need for clarity in research design, data collection, and analysis.

Furthermore, Pandey (2015) delves into the philosophical underpinnings, asserting that research methodology encapsulates the researcher's worldview, guiding their choices of methods and tools. This multidimensional perspective aligns with the evolving nature of research, acknowledging the diverse paradigms that influence methodological decisions. However, critiques also arise, with some scholars arguing that a singular, universally accepted definition remains elusive due to the dynamic nature of research practices (Kadasah et al., 2022).

Despite such debates, the consensus centers on recognizing research methodology as the backbone of scientific inquiry, shaping the trajectory and reliability of research outcomes. The methodology integrates qualitative techniques like interviews and quantitative methods like system evaluation to collect well-rounded feedback. Additionally, it adopts an agile approach to enable flexibility and adaptations throughout the iterative development process.

3.3 Fact Finding Approaches

The researcher established a concentrated plan to acquire information using qualitative methodologies to achieve deep knowledge on the problem being examined. Procedures were picked based on guaranteeing rigor, dependability, and validity of the data acquired, given the nature of the study. Those strategies consisted of:

3.3.1 Interviews

Interviews with students and professors are a great qualitative tool for investigating viewpoints on Central University's student attendance system. Utilizing one-on-one interviews promotes the use of open-ended questions, offering a complete knowledge of experiences and attitudes about the system. When interviewing students, the focus might be on collecting insights into their impressions of the system's utility, user-friendliness, accessibility problems, impact on attendance patterns, efficacy, and recommendations for changes. Similarly, faculty interviews give the chance to obtain thorough input on the integration of the system into their courses, its effect on teaching techniques, observed usage trends among students, and ideas for improvement. Additionally, interviews with IT workers linked with the attendance system might disclose technical issues, plans for improvements, and strategies for enhancing system performance. The open-ended nature of these interviews permits the collection of personalized narratives, anecdotes, worries, and responses, contributing to a deeper qualitative knowledge of Central University's student attendance management system.

3.3.2 Observation

Direct observation of the application of the attendance management system at Central University appears as a valuable qualitative tool for acquiring insights into student and staff relationships. This technique may entail actively engaging in classes, quietly watching how students check in via the system, detecting any encountered concerns, and determining trends in system usage. The researcher can attentively monitor if the attendance process effortlessly integrates into class procedures or leads to disturbances.

Furthermore, observations of user interfaces and faculty/staff interactions with the system, such as data entry and report access, might disclose firsthand usability concerns. Thorough notes on system interaction, comments made, assistance needed from personnel, and techniques taken during obstacles give a complete knowledge for prospective modifications. The observational technique enables for the real-time viewing of user activities, delivering a more authentic viewpoint than depending exclusively on self-reported accounts. The qualitative data gathered through observational approaches promise an immersive knowledge of the adoption, acceptability, and issues related with the operation of the student attendance system at Central University.

3.4 System Development Methodology

The agile methodology serves as the foundation for developing the attendance system, facilitating incremental improvements through swift cycles of planning, development, testing, and integration. Central tenets of agile, including continuous collaboration with stakeholders, a focus on delivering working software, and adaptability to change, are integral to the development process. In particular, the Scrum framework is employed to structure the development into fixed-length sprints. Each sprint encompasses various phases such as requirement analysis, design modifications, coding, testing, and integration, culminating in a comprehensive review. This structured approach plays a crucial role in effectively managing the complexity of the project, fostering transparency, and ensuring frequent feedback loops. By adhering to Scrum principles, the development team can respond promptly to evolving requirements, enhance communication between team members and stakeholders, and maintain a consistent pace of delivery

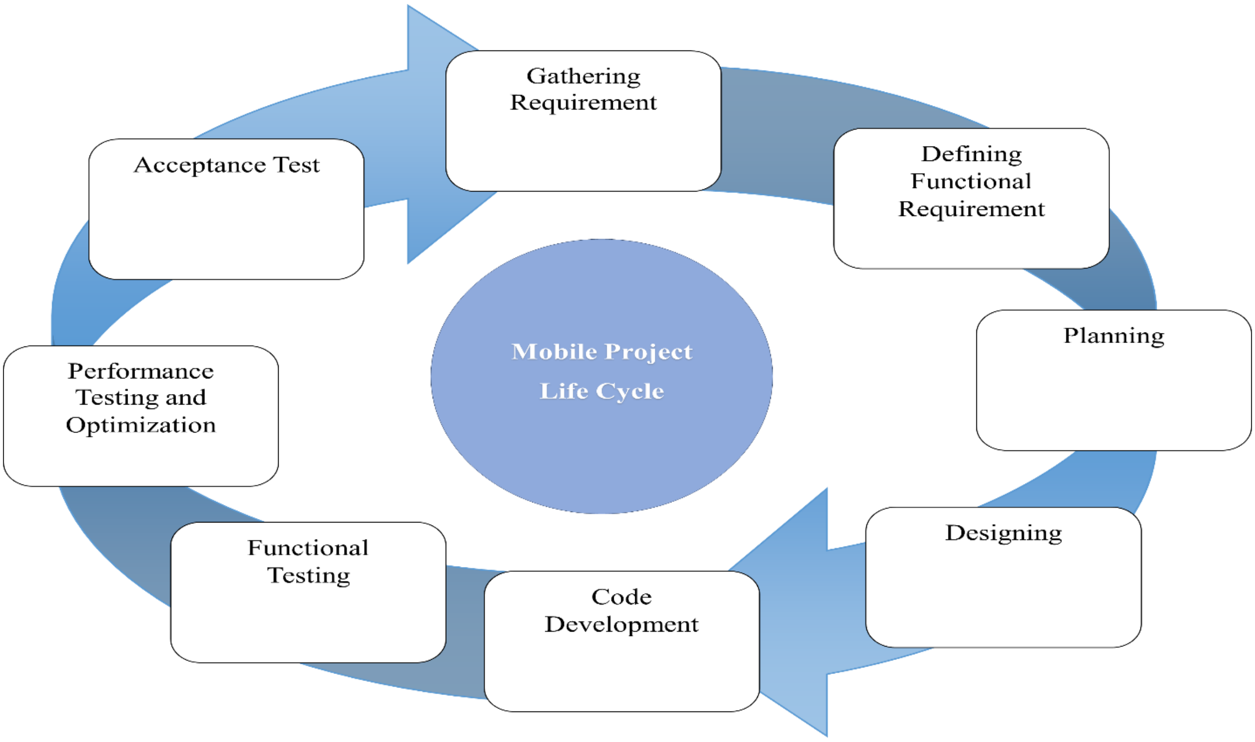

Gathering Requirement
Defining Functional Requirement
Planning
Designing
Code Development
Functional Testing
Performance Testing and Optimization
Acceptance Test
Mobile Project Life Cycle

***Figure 3. 1** Mobile Application Development Life Cycle*

***Source:** Study material December (2023)*

3.4.1 Requirements Gathering

Comprehensive requirements are gathered early on through structured interviews with administrators, instructors, and students at Central University. This elicits user needs, priorities, and expectations for the system. Both functional specifications like login, facial recognition, notifications, etc. and non-functional aspects like usability, security, and scalability are covered.

Additionally, documents outlining existing attendance policies and pain points are reviewed to further understand the landscape and challenges. The aggregated findings shape the foundations of system design and development.

3.4.2 System Architecture and Design

A cloud-based architecture leveraging Supabase for the backend database and Flutter for cross-platform mobile development is adopted. Supabase provides secured database storage and management, while Flutter enables iOS and Android compatibility.

System design is an iterative process, refined continuously based on user feedback on prototypes. However, core elements include role-based access control, graphical display of attendance metrics, and real-time data sync across devices.

3.4.3 Development and Implementation

The MobileFaceNet deep neural network forms the crux of facial recognition capabilities, trained on collected face datasets. For mobile optimization, the network is compressed by reducing parameters while retaining accuracy.

The front-end mobile interface is built with Flutter, incorporating native features like camera access. Supabase facilitates crucial capabilities like user authentication and data persistence.

Multiple builds are tested internally throughout development and by a small external user group before full-scale deployment. This reveals bugs and areas of improvement to address.

3.4.4 Training and Support

Comprehensive training guides covering login, interface navigation, taking attendance, and generating reports are provided. In-person orientation sessions prepare administrators and instructors before the system rollout.

A multifaceted support system through email, chatbots, and contact forms aids the transition by promptly resolving user difficulties. Post-launch surveys also elicit areas where more training or tweaks may be beneficial.

3.5 Software Requirement Specification (SRS)

The SRS document plays a crucial role in building the foundation for the design and implementation of mobile attendance solutions by explicitly recording detailed functional demands and priorities based on a full understanding of academic contexts.

As research demonstrates, rigorous definition related to institutional objectives, pain spots, workflows and restrictions is a prerequisite to creating durable tailored tech interventions (Ali & Lai, 2015). Accordingly, the SRS codifies essential attendance use cases covering teacher monitoring demands, student interfaces, integrations with timetables/class schedules etc. led by accurate comprehension of contextual procedures.

Equally, it specifies non-functional requirements regarding performance, security, compliance etc. molded strategically by education sector rules, mobile platform standards and campus level policies. Itemization of user types and access concerns facilitates role-based customization. Prioritized lists allow correlating features to academic aims.

Overall, meticulously enumerating the issue and solution scope within institutional frames of reference gives clarity for later constructing strong modular designs, platform capabilities mapping and deployment planning - converting demands to technological replies.

The SRS paper consequently promises verifiable, generalizable yet personalized mobilization routes for dramatic change.

In essence, a thoroughly built SRS works as an anchoring blueprint combining organizational functional objectives and technical potential at the confluence of projected advancement. For sustainable attendance solutions, detailed specification related to academic aims and practices is important.

Please let me know if you need any explanation or have additional requirements for examining the value of SRS documents for directing mobile attendance system development.

3.6 Purpose of the System

The mobile attendance management system aims to transform the prevailing manual attendance tracking processes in educational institutions through automation and enhanced efficiency - by equipping faculty and administrators with integrated visibility into attendance statistics and patterns, the purpose encompasses timely identification of emerging student issues enabling intervention; tailored notifications facilitate following up absenteeism helping uphold academic policies and performance correlation analysis provides data-driven inputs for resource planning.

Fundamentally, by mobilizing a convenient yet systematic attendance monitoring mechanism, the system purposes to make class participation central toward driving institutional objectives around student progress - the envisioned solution thereby elevates attendance data processes from operational hygiene toward impact generation via actionable insights for all stakeholders. The fundamental promise rests in improving educational aims themselves by making attendance central among vital indications monitored by the institution through technological augmentation.

3.7 Scope of the Developed System

The scope of the developed mobile-based student attendance management system encompasses comprehensive functionalities, including accurate attendance tracking through biometric facial recognition, secure user authentication for authorized access, real-time monitoring capabilities, efficient data storage and management, generation of detailed reports and analytics, notification and alert mechanisms, user-friendly interfaces for both mobile applications and backend systems, compliance with privacy regulations, scalability for future expansion, seamless integration with existing academic systems, user feedback mechanisms for continuous improvement, and robust security measures to safeguard sensitive data and prevent unauthorized access. This comprehensive scope outlines the system's boundaries and features, ensuring it satisfies the institution's objectives for effective attendance management while following regulatory and security standards.

3.8 System Architecture

The system architecture of the mobile-based student attendance management system is designed to promote efficient and secure operation. At its heart, the design incorporates a client-server approach, where the mobile application operates as the client, while the backend server maintains data storage, processing, and communication. The client side features a user interface for students, instructors, and administrators, allowing straightforward interactions for attendance tracking and system access. The backend server, equipped with a database, oversees the storage of attendance records, user profiles, and system configurations. The integration of biometric face recognition technology comprises a separate module for gathering and processing facial data. The system utilizes a secure communication protocol to send data between the client and server, preserving the security and integrity of critical information. Additionally, the design enables for scalability, enabling the system to meet rising user populations and shifting institutional demands. Continuous monitoring, regular backups, and frequent system upgrades contribute to the dependability and resilience of the entire system design.

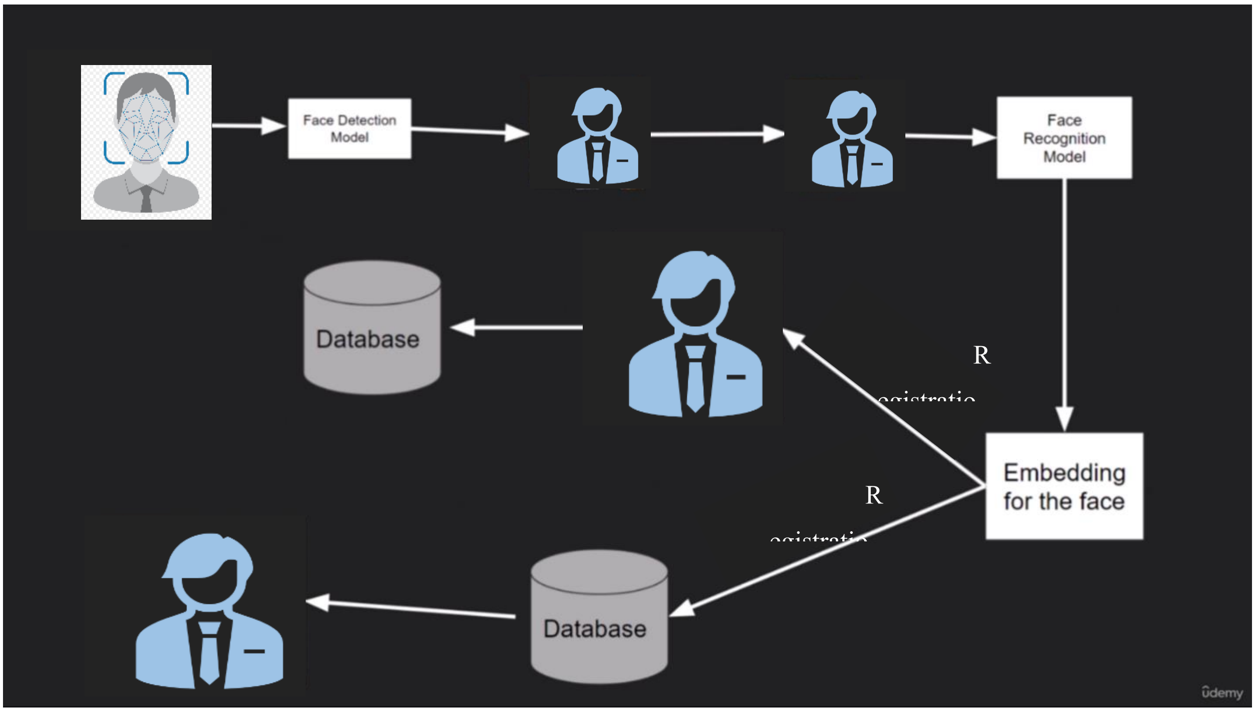


Figure 3. 2 *System Architecture*

Source: *YouTube.* *https://www.youtube.com/watch?v=3Uv0zvE_gNM*

3.9 System Modules

The system targets two core academic users - Teachers and Students. Teachers require analytics on class and student trends to identify issues. Students need easy self-marking workflows via facial recognition. Personalized experiences for each role enhance system stickiness fit for the institutional context. Strategic user focus expands solution relevance beyond technology to attendance processes. Embracing user roles anchors innovation for campus ecosystems.

3.9.1 Tutors:

- Teachers would utilize the system to take attendance during classes and review participation analytics.
- Via the mobile app or web dashboard, they can initiate the facial recognition attendance check-in process.
- Once students have checked-in, teachers can view attendance reports summarizing absentees.
- Dashboards would also provide analytics on overall class and individual student attendance patterns.

- Helps teachers identify struggling students based on low attendance to provide extra support.
- Simple workflows facilitate easy teacher onboarding and system adoption.

3.9.2 Students:

- Students' primary interaction is via the mobile app to check-in their attendance.
- When launching the app in class, students use the face scanner for biometric confirmation.
- No manual data input is needed, reducing friction.

3.9.3 Admin:

1. User Management
    - Add, update, and cancel user accounts (students, teachers, admins).
    - Manage user profiles and permissions/access levels.
2. Attendance Settings
    - Configure attendance parameters like late arrival/early departure thresholds.
    - Customize attendance status categories (present, absent, late etc.).
    - Set up automated notifications and alerts.
3. Academic Settings
    - Build academic calendar and schedules.
    - Map school terms, semesters, sessions.
    - Insert public holidays.
4. Class Management
    - Add/modify list of courses and classes.
    - Assign courses to periods and designate teachers - Class room scheduling.
5. Attendance Analytics
    - School-wide, departmental, and class-wise reports.
    - Student attendance history and percentages.
    - Export and print reports.

6. System Administration
    - Software upgrades, troubleshooting.
    - Usage monitoring and analytics.
    - Manage data backups.

3.10 Hardware Infrastructure:

- Mobile devices, such as smartphones or tablets, are designated for use by students and teachers.
- Robust backup and storage systems to ensure data integrity and accessibility.
- Essential network equipment, including routers, switches, and Wi-Fi access points, to establish and maintain seamless connectivity within the attendance management system.

3.11 Software Architecture:

- Mobile app for iOS and Android platforms.

3.12 Software Features

The mobile-based student attendance management system boasts a user-friendly interface, supporting secure authentication for students, teachers, and administrators. It enables real-time attendance tracking through various methods and offers a notification system for alerts and reminders. The app integrates seamlessly with academic calendars, providing comprehensive attendance views. Robust reporting and analytics tools help identify patterns, and the system ensures differentiated access levels, data security, and offline functionality. With automated record-keeping, scalability, customization options, and a feedback mechanism, the system promotes ease of use and adaptability.

Additionally, it serves as a communication platform and facilitates integration with other school systems, contributing to its overall effectiveness in managing student attendance efficiently.

CHAPTER FOUR

System Analysis and Design

Chapter Summary

This chapter delineates the hands-on development, deployment, and performance benchmarking of the Facial Recognition Attendance Tracker mobile application built as part of this research project. Guided by current attendance management protocols, the multi-platform application leverages advanced face APIs to automate student attendance capture via mobile devices.

It is also giving a full study of the system with an emphasis on its objectives. These comprise system development feasibility, functional, economic, and technological feasibility. The chapter further presents a full discussion of the system's functionality and the strategies taken to attain its aims. Emphasis is given on different factors such as system processes, design technologies, data flow, user case activities, relationships, entity relationships, user functionality, and the functionalities of system objects. Additionally, the discussion involves the explanation of the system's database design schemas, throwing light on the links between tables and detailing the features of the system's output.

4.1 System Development Feasibility Analysis

By employing this methodology, the researcher effectively collected and interpreted information, pinpointing problems through a systematic process known as "System Feasibility Studies." The activities involved in this process included modifying and further refining the approach to decomposing the system into its components. Through these activities, the researcher aimed to enhance the understanding of the system's feasibility, focusing on system development feasibility, functional feasibility, economic feasibility, and technical feasibility. This approach facilitated a comprehensive examination of the system's components, aiding in the identification of potential challenges and the formulation of informed solutions.

4.1.1 Functional Feasibility

The proposed system is a mobile application for student attendance management utilizing facial recognition functionality. The mobile app is developed natively for Android and iOS platforms, downloadable from their respective app stores. Students and teachers require smartphones with an integrated camera along with a stable internet connection for the facial recognition attendance workflows. Initial user registration via the app captures the facial biometric data to create student profiles. For attendance, students simply launch the app to scan their face for automated presence marking. The mobile app connects with cloud-hosted servers for processing and storing attendance data.

Therefore, the solution does not mandate specialized infrastructure or dedicated hardware procurement for deployments. By leveraging the existing computing platform of consumer mobile devices, it minimizes adoption barriers regarding technical requirements. The availability of facial recognition APIs on mobile operating systems further eases integration. With a user-friendly mobile app at its core connected to backed services on cloud infrastructure, the system is functionally feasible for rapid real-world implementation. Minor enhancements may augment reliability, but the current technology base fulfils core requirements regarding practical application.

4.1.2 Economic Feasibility

The proposed attendance management system does not need purchasing specialist hardware or software technologies. The mobile application employs built-in cameras and biometric APIs accessible on common consumer devices running Android or iOS. Cloud platforms used for hosting backend functions are offered on a pay-as-you-go operational expense basis rather than capital outlays.

From the development standpoint, open-source libraries for facial recognition and standards-based programming languages save license costs. The modular design of mobile app development also contained initial development time and complexity estimations made during project scoping stages. Agile sprints ensured timely delivery by the engineering team averting cost overruns.

Therefore, by capitalizing on the ubiquitous computing platform of mobile devices and cloud infrastructure for attendance automation workflows, the solution enhances the economic feasibility. Scalable pricing models match costs to value delivery while contemporary development processes provide predictability of initial and ongoing spending. The technology enables attendance tracking without extravagant expenditures connected with legacy biometric systems, expanding access for academic institutions confronting financial limits. The economic feasibility assessment verifies cautious deployment of technological enablers to improve attendance management inexpensively.

4.1.3 Technical feasibility

The suggested method leverages facial recognition technology for automated attendance marking on mobile devices utilizing built-in cameras and biometric scanning capabilities. Recent improvements in mobile computer vision algorithms can now accurately match facial traits to individual identities in real-time. The prevalence of smartphone usage further provides the requisite technology for performing AI-based facial recognition consistently.

On the software infrastructure side, powerful API libraries and pre-trained machine learning models are readily accessible for integration into new mobile applications. Mature cloud platforms offer computer capability for operationalizing the face analysis tasks. The consistency of programming languages across mobile and online development also fosters synergies.

Therefore, with dependable facial recognition libraries, expanding device capabilities, flexible cloud infrastructure and interoperable platforms, the technical building blocks are firmly in place. While enhancements can augment performance, technical limits do not hinder fundamental functional implementation or scale. The solution realistically automates attendance by applying modern technologies linked to use case needs without facing prohibitive feasibility constraints from tests or prototypes. Face recognition and cloud-hosted processing exhibit readiness increasing application scope.

4.2 System Design

For the "Face Recognition Mobile-Based Student Attendance Management System," the suggested System Requirements Specification (SRS) document is not the main focus throughout the System Design phase; instead, an implementation-ready format takes center stage. In order to bridge the gap between the issue domain and the current system, this transformative stage attempts to precisely define how the entire system will function. Sub-activities are created from the complex activities listed in the SRS document to make them easier to handle. With a strategic coordination of these efforts, the researcher accomplishes the main objective of creating the suggested software system. The System Design phase is essential for guaranteeing a smooth and effective architecture, optimizing workflows, and setting the foundation for an effective rollout of the facial recognition mobile-based student attendance management system.

4.2.1 Data Design Diagrams

This data design diagram provides a conceptual blueprint for the organizational framework of the face recognition-based student attendance management system. Its primary purpose is to articulate and visualize the planned structure of student-related information, attendance records, and other relevant data within the database. By presenting an organized representation, this diagram serves as a crucial planning tool, enabling a comprehensive understanding of how data elements will be structured and interconnected. It plays a pivotal role in guiding the effective implementation of the system's database architecture.

The design modeling approach being utilized is the Unified Modeling Language (UML), whose simple and straightforward concepts make it well-suited for depicting various diagrams of a project. This design modeling technique can more effectively convey conceptual and logical diagrams in an easy-to-understand way. Below are some key aspects of the proposed project architecture as represented in a logical view:

- Activity diagram
- Use case diagrams.
- Data Flow diagrams,
- Entity Relationship Diagram
- Sequence Diagrams

4.2.1.1 Activity Diagrams

The activity diagrams below depict the set of use case workflows, processes, structural activities, and actions that exist in the system. It models what flows or activities occur, in what order/sequence, based on various conditions and Control Flows.

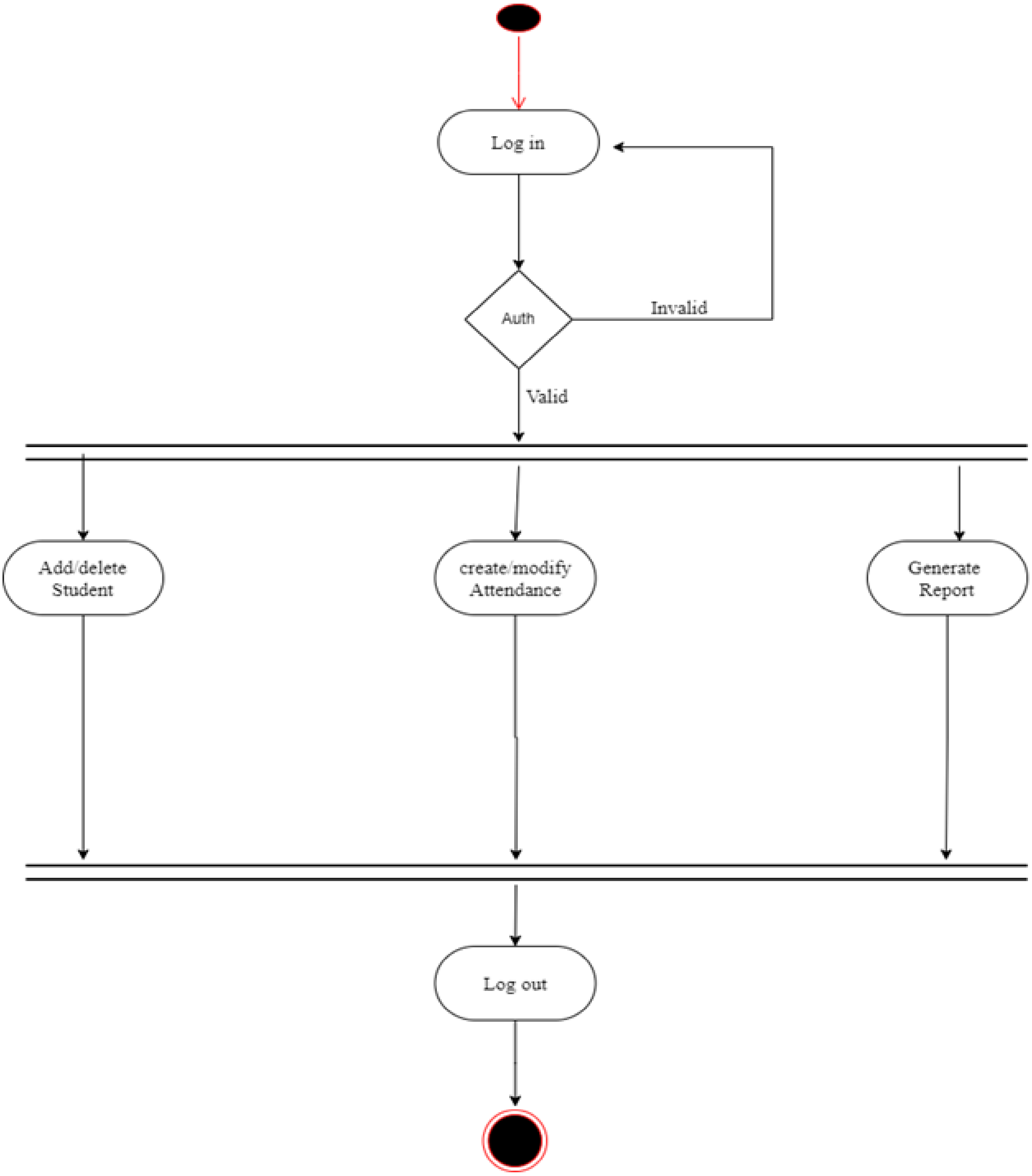


***Figure 4. 1** Tutor Activity Diagram*
***Source:** Study materials December (2023)*

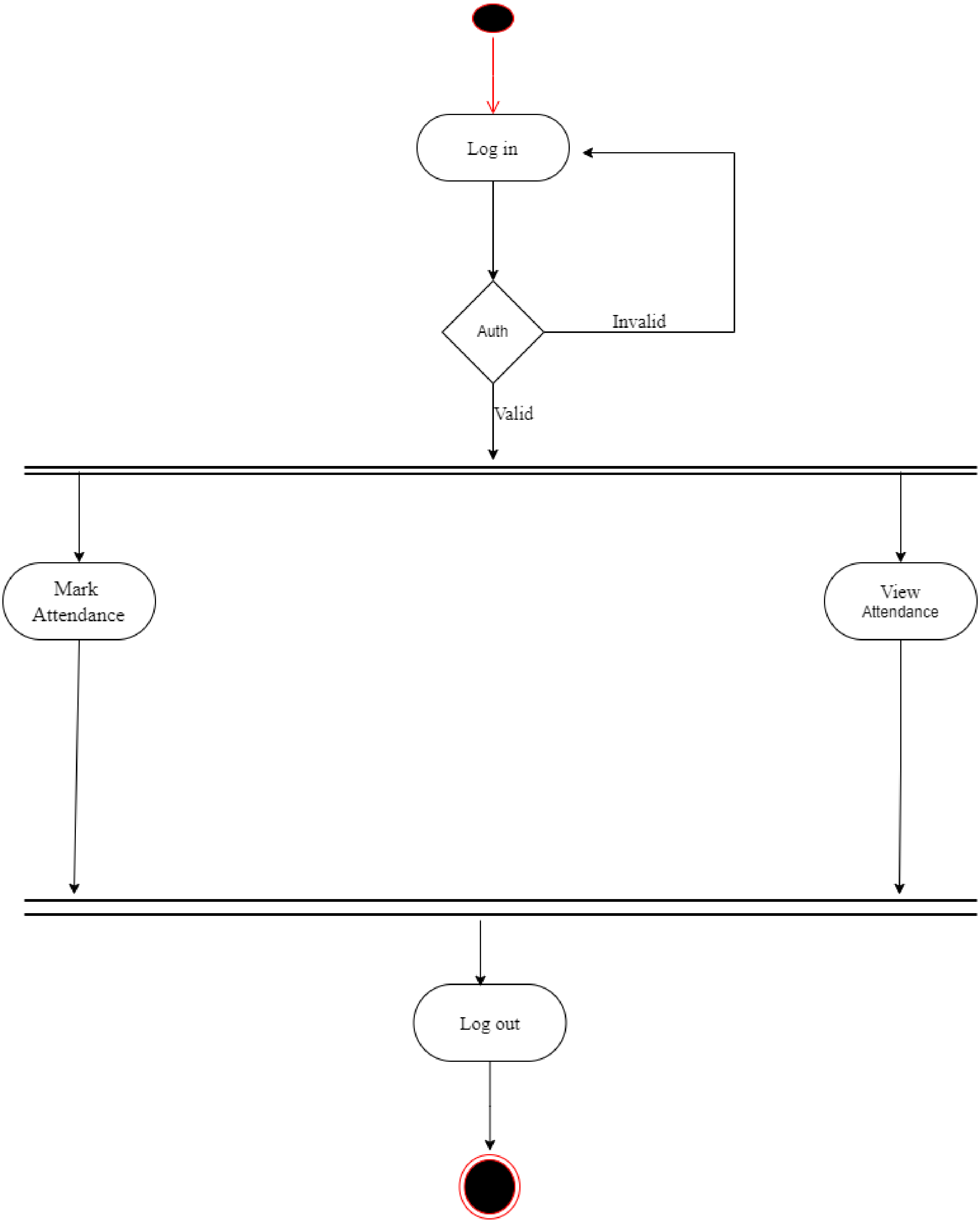


Figure 4. 2 *Student Activity Diagram*
Source: *Study materials December (2023)*

4.2.1.2 Use case diagrams.

The activity diagrams below depict the key workflows and actions of the users of the attendance system.

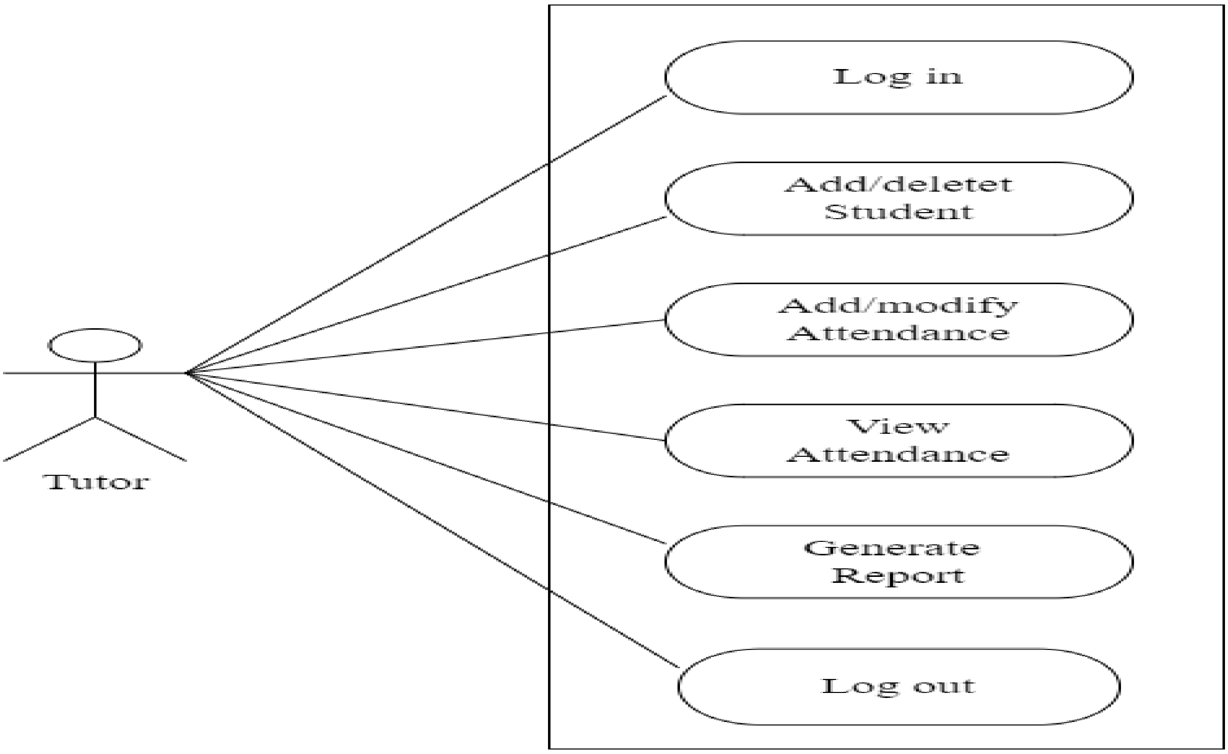


Figure 4. 3 *Student Activity Diagram*
Source: *Study materials December (2023)*

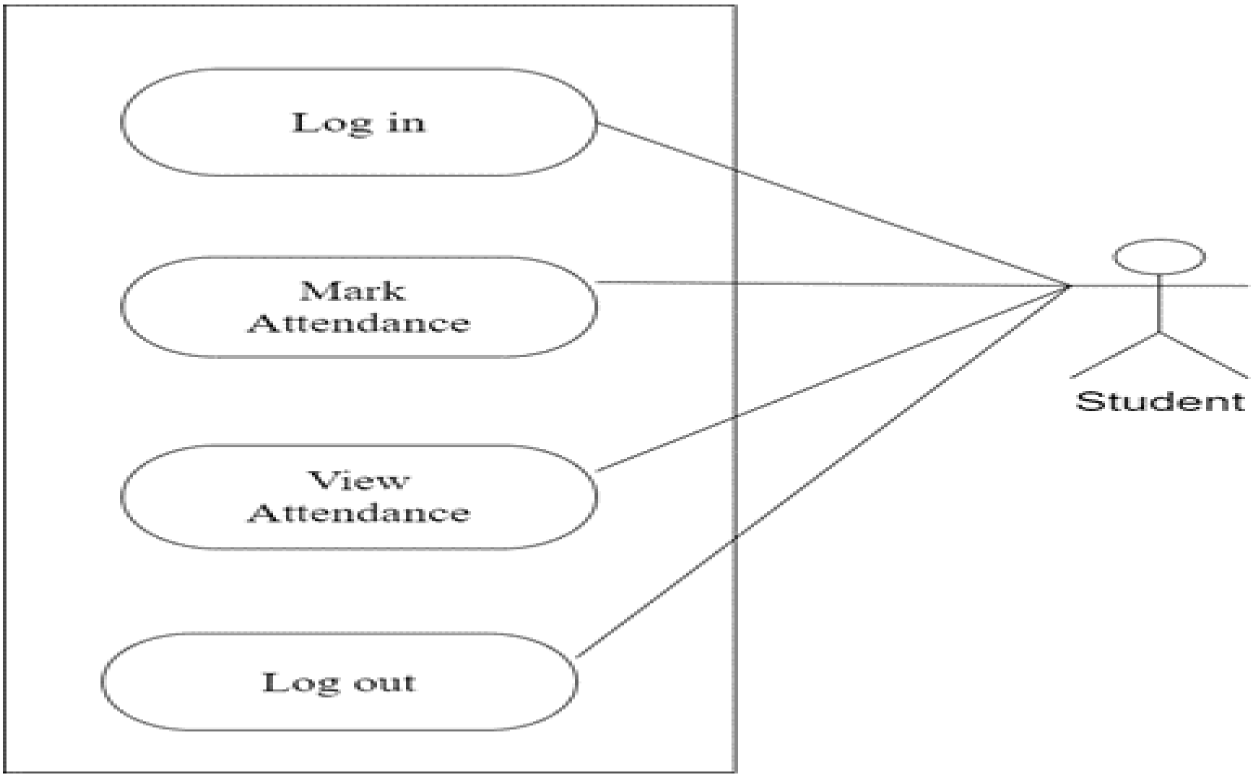


Figure 4. 4 *Student Activity Diagram*
Source: *Study materials December (2023)*

4.2.1.3 Data Flow diagrams

A Data Flow Diagram (DFD) is a tool for understanding the flow of information within a system, focusing on data mobility and transformation processes. In an attendance system, the Level 0 DFD provides a comprehensive view of the system, including attendance records, reports, and user information. The Level 1 breakdown reveals four sub-processes: user access management, attendance tracking, report production, and data storage.

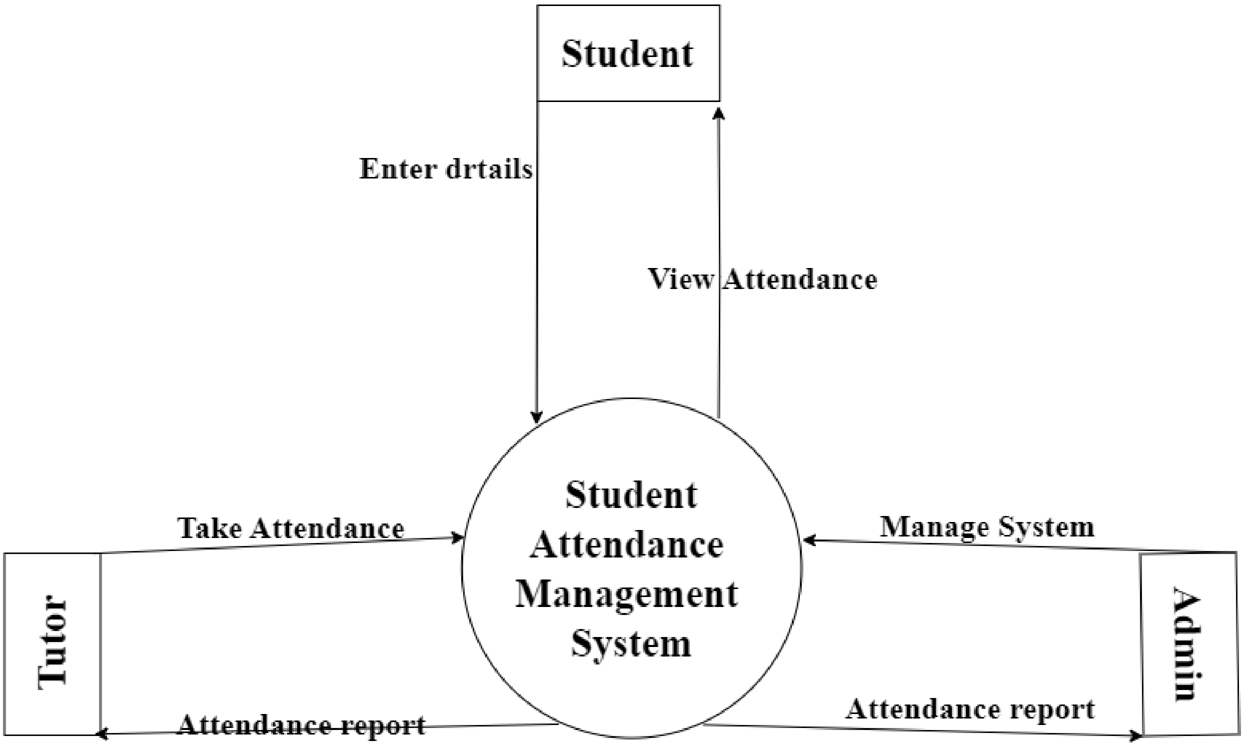


Figure 4. 5 *Level Zero Data Flow Diagram*

Source: *Study material December (2023)*

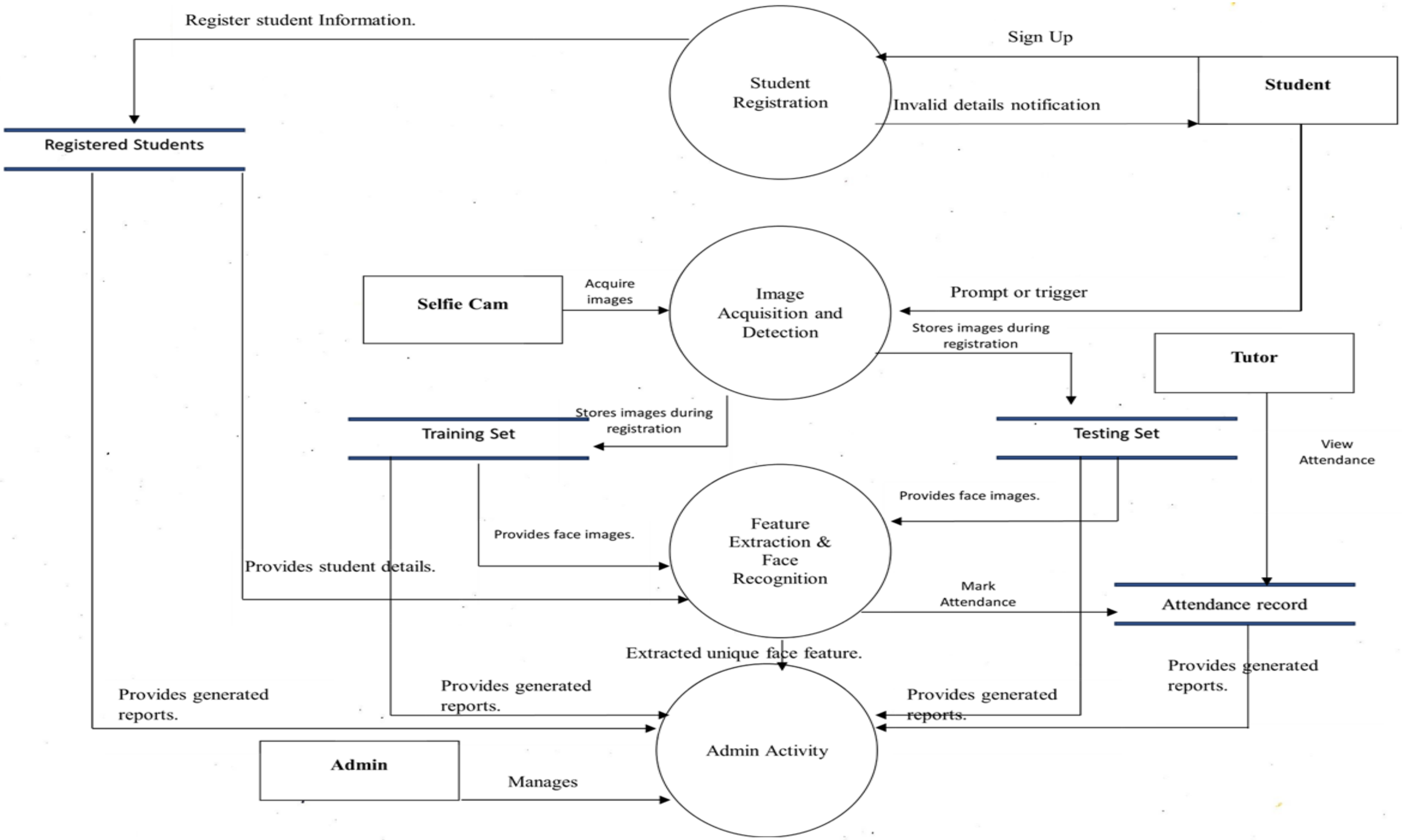

Register student Information.
Student Registration
Sign Up
Student
Invalid details notification
Registered Students
Selfie Cam
Acquire images
Image Acquisition and Detection
Prompt or trigger
Stores images during registration
Tutor
Training Set
Stores images during registration
Testing Set
View Attendance
Provides face images.
Provides face images.
Feature Extraction & Face Recognition
Provides student details.
Mark Attendance
Attendance record
Extracted unique face feature.
Provides generated reports.
Provides generated reports.
Provides generated reports.
Provides generated reports.
Admin
Admin Activity
Manages

Figure 4. 6 *Level 1 Data Flow Diagram*

Source: *Study material December (2023)*

4.2.1.4 Entity Relationship Diagram

The ER diagram demonstrates sound modeling principles and data analysis to develop a schema that meets the attendance system requirements from an entity relationship perspective.

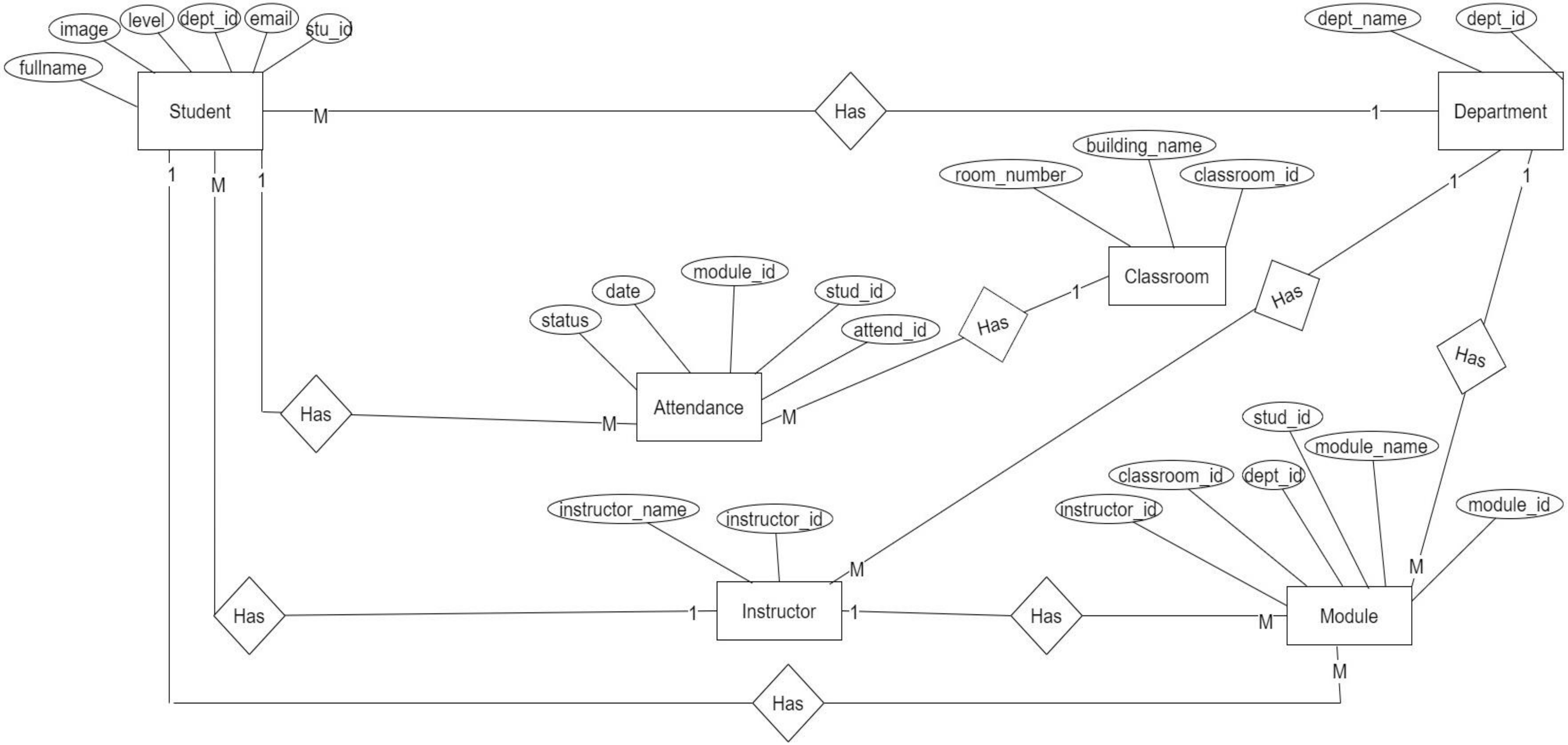


Figure 4. 7 Entity Relationship Diagram

***Source:** Study material December (2023)*

4.2.1.5 Sequence Diagrams

A sequence diagram models the chronological flow of messages during a process between system parts or actors. It depicts the sequential interactions arranged in time order down the page, with messages passed from left to right or top to bottom. Sequence diagrams are used a lot in software design to visualize and specify logic, especially for complex use cases involving several objects. Overall, it shows the big picture flow while highlighting time sequence and messaging ordering.

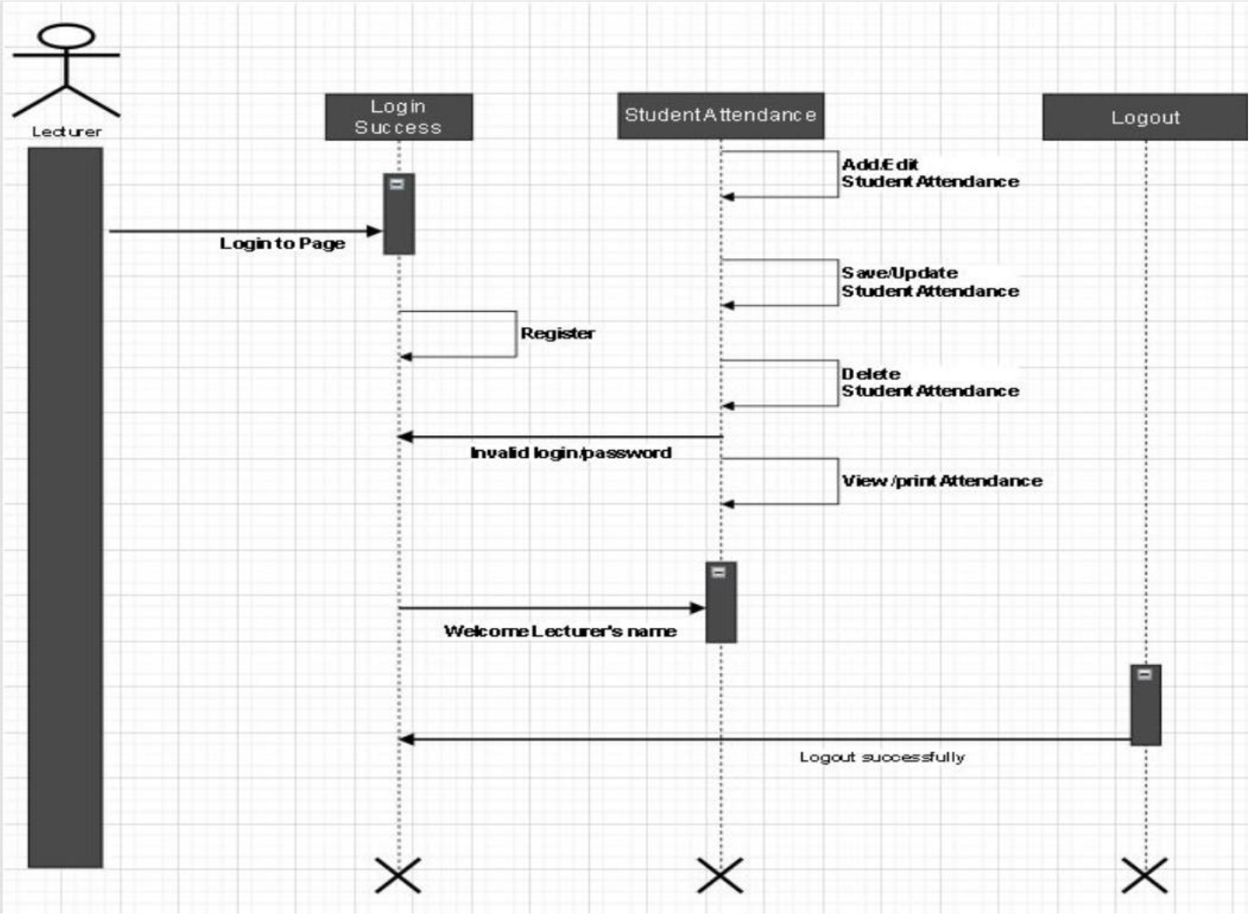


Figure 4. 8 *Sequence diagram for Tutor*

Source: *Study material December (2023)*

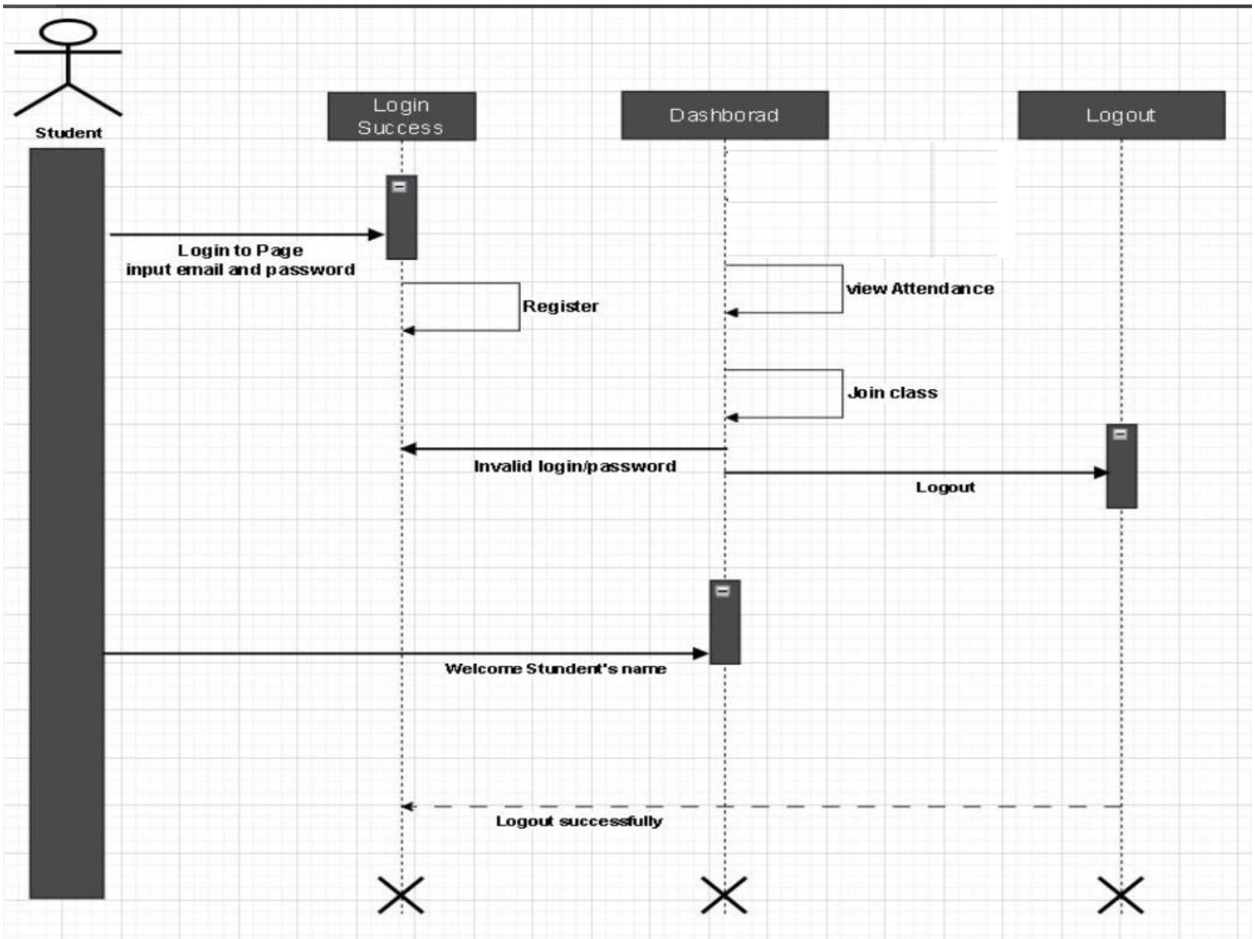


Figure 4. 9 *Student Sequence diagram*

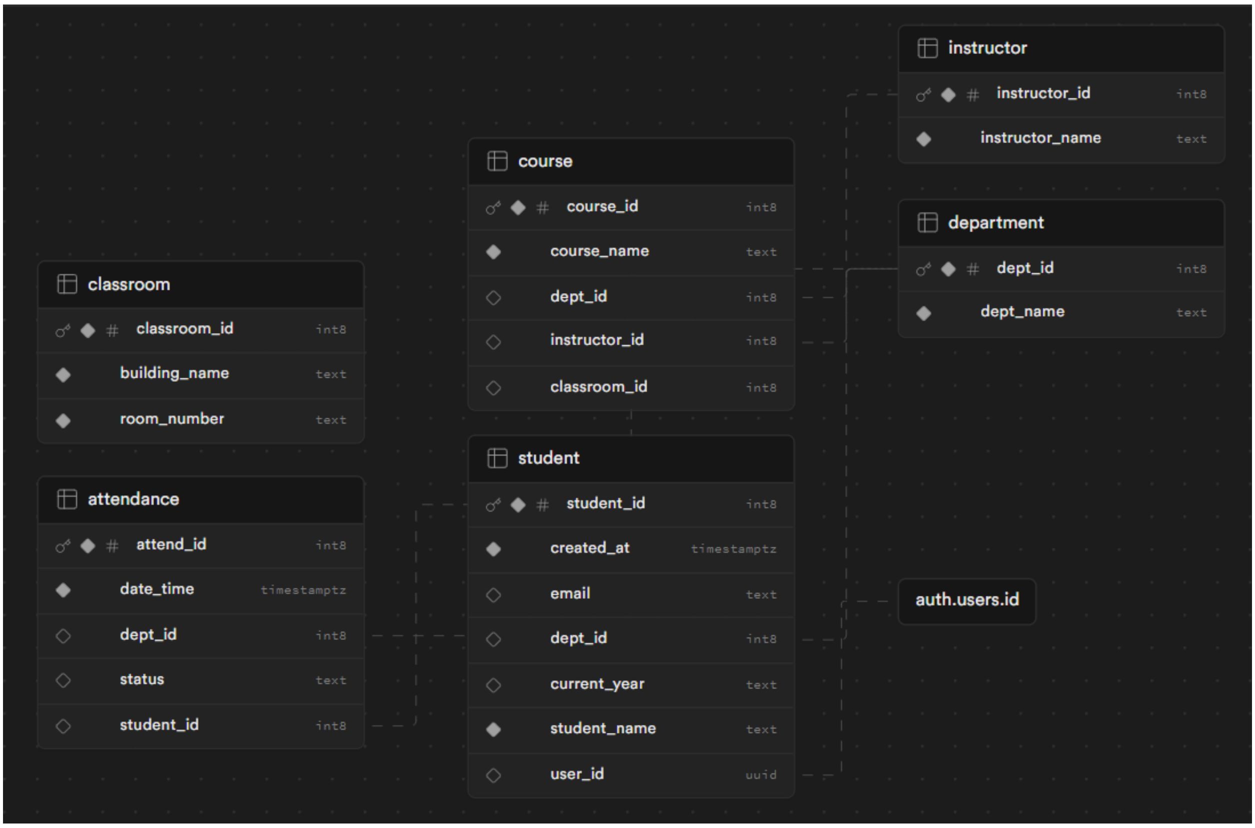
Source: *Study material December (2023)*

4.2.1.6 Database Design

It is necessary to store and retrieve the system's data from a database. A component of system design is database design. At the analysis stage, the data structures and elements to be stored have been determined. To design the data storage and retrieval system, they are organized and combined.

4.2.1.6.1 Class Diagram

A class diagram describes the static structure and relationships of classes or objects in a software system. It visualizes classes with attributes, operations, and connections modeled as a diagram with boxes and lines. Class diagrams helps design and document the logical view depicting software classes and their relationships such as inheritance, aggregation, and association



classroom		
◆ #	classroom_id	int8
◆	building_name	text
◆	room_number	text

attendance		
◆ #	attend_id	int8
◆	date_time	timestamptz
◇	dept_id	int8
◇	status	text
◇	student_id	int8

course		
◆ #	course_id	int8
◆	course_name	text
◇	dept_id	int8
◇	instructor_id	int8
◇	classroom_id	int8

student		
◆ #	student_id	int8
◆	created_at	timestamptz
◇	email	text
◇	dept_id	int8
◇	current_year	text
◆	student_name	text
◇	user_id	uuid

instructor		
◆ #	instructor_id	int8
◆	instructor_name	text

department		
◆ #	dept_id	int8
◆	dept_name	text

auth.users.id

Figure 4. 10 *Class Diagram and the relationships*

Source: *Study material December (2023)*

CHAPTER FIVE

System Implementation and Testing

Chapter Summary

The chapter discusses techniques for turning software design into executable systems using coding standards, databases, APIs, and modular integrations. Comprehensive testing ensures functionality, interoperability, performance, and security through unit, integration, frontend, backend, user acceptability, and other methods. Specific target areas include workflow, scalability, database components, security hardening, and gaining user validation. The goals are early defect discovery, achieving requirements, ensuring all components perform harmoniously, validating alignment with specifications, and securing deployment signoff following rigorous assessment via manual and automated test suites based on test cases.

5.1.System Implementation

The system implementation phase translates the logical software representation created during the design phase into a physical, working product through coding, testing and integration. It takes the detailed system architecture, modules specifications, database models and UI prototypes and realizes them using programming languages like Java, Python, JavaScript etc. (Kasthuri, 2018).

The process begins by setting up the development and testing environments. Coding follows conventions like naming standards, modularity, documentation, and version management using GIT, SVN etc. (Li et al., 2021). Databases are created based on entity-relationship diagrams using MySQL and query optimization techniques. Connectivity layers leverage REST or SOAP APIs and error handling is incorporated.

Developers integrate individual modules systematically following incremental or top-down approach through a build management system like Jenkins. They define test data requirement specifications focusing on edge cases early. Rigorous unit testing of all modules using white box methods like statement coverage establishes confidence before integration. End-to-end functionality testing follows with black box methods leveraging test cases defined much earlier aligning to IEEE standards.

Performance testing puts the system under high load, volume and stress situations measuring response times, bottlenecks, and tuning SQL queries. Security reviews inject malware and encryption is implemented to strengthen data protection in transit and rest conforming to OWASP guidelines. Finally, user acceptance signoff verifies if the implementation meets all specified functional, non-functional, and business expectations (Xu et al., 2022).

5.2.System Testing

System testing is a vital element of the software development lifecycle that holistically assesses and confirms the functionality, interoperability, performance, and security of the fully integrated information system (Hu et al., 2021). The overarching objective is to detect software faults and deviations that may have emerged after individual modules and components have been unit tested and integration issues handled. Comprehensive system testing gives assurance that all essential parts including functional processes, backend procedures, user interfaces, network communications and database interactions work in harmony to match the overall system requirements and specifications as first signed off on.

A combination of system testing approaches covering black box testing, white box testing, manual testing and automation testing are applied at this point. The scope comprises in-depth testing across unit, integration, front-end, performance, security, regression, and user acceptability testing levels (*29119-4-2015 - ISO/IEC/IEEE International Standard - Software and Systems engineering--Software testing--Part 4: Test Techniques*, 2015). Hundreds of test scenarios and test cases based on similar partitions, boundary values and real production values are conducted to evaluate the system under different conditions. Test management solutions monitor coverage and defect triaging as tests progress through scripted, ad hoc, and exploratory approaches. The climax evaluates alignment with both commercial expectations and technical standards established early on.

5.2.1. Test Case Specifications

Well-defined test cases are crucial for complete system evaluation. Test case specifications explicitly define the entire methods, inputs, intended outcomes and pass/fail criteria to validate essential functional and non-functional elements (Crnkovic, 2018). Scoping begins at system design with prioritized business scenarios, real-world workflows and performance benchmarks. Parameters like security standards, global regulations and technology limits further guide test coverage.

Careful attention goes into assembling and organizing test data throughout test case authoring. Typical, border, edge, invalid, unsupported, and erroneous scenarios get recognized based on equivalence partitioning and boundary value analysis (*CCNC 2021: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, 2021). Test harness codifies the preconditions, steps, data sets, expected system state changes, postconditions and formal delivery acceptance norms for every defined test procedure. Traceability matrices examine requirement coverage through manual and automated approaches.

5.2.2. Front-End Testing Strategy

As the user interface and interaction pathways have direct business impact, front-end testing focuses on validating look, feel, usability and workflows based on ISO 9241 standards. Unit, integration, system, regression, and user testing across devices is conducted (Testim, 2020).

Front-end testing validates what users visually see and interact with as per ISO 9241 standards covering usability metrics like efficiency, satisfaction, and accessibility (Bose, 2023). Testing evaluates consistent look across browser environments, validation of design elements, intuitive feel of workflows, responsive iterations, and mobile friendliness. Unit testing establishes modular front-end performance while integration examines interplay with backend.

System testing analyzes entire workflows functionally and at scale, gauging overall consistency and speed. Regression testing compares outputs after modifications to find visual regressions. User acceptance testing engages real users for qualitative feedback on convenience and ease of goal achievement through tasks on supported devices (Hamilton, 2023). Ongoing exploratory testing uncovers further usability enhancements for the application front-end. Metrics on session times, click paths and conversions provide data-driven input.

5.2.2.1. Unit Testing

Unit testing represents the first level of software testing where smallest testable units of application code are individually and independently analyzed to ensure the functional correctness before integration (Contributor, 2023). Test scenarios imitate numerous input possibilities to carefully test discrete code units that encapsulate related operations using stubs and drivers. White box testing methodologies provide analysis of internal structures enabling developers to 'test early and often’ at code level harmonizing with test driven development philosophy (GeeksforGeeks, 2023).

Important test metrics including statement, path, branch, and condition coverage statistically analyze completeness and quality offering significant indicators on complexity (Dizdar, 2023). Automation with xUnit frameworks result in repeatable regression testing suites eliminating new issues creeping up due to code modifications. Unit testing creates a stable foundation enabling incremental integration of verified components culminating in system level validation success. Overall, it assists in isolating faults early, push modular designs, expedite debug, and embed quality via prevention over inspection.

5.2.2.2. Functional Testing

Functional testing approaches validate that the system delivers all behaviors and outputs required to meet the goals that were originally described and scoped (Bose, 2023b). Systematic test procedures generated from use case models, workflow diagrams and GUI prototypes check end-to-end activity paths exercising full stack layers. Equivalence partitioning, boundary value analysis, decision table and state transition approaches enable test data selection concentrating on valid, invalid and edge cases (*What Is Functional Testing? Types & Examples | OpenText*, n.d.).

Test execution follows defined documentation formats employing formal test plans detailing objectives, requirements mapping, environmental needed and hazards. Detailed test cases and processes log inputs, actions, outputs, and pass/fail outcomes adhering to IEEE 829 for traceability. Specially built stubs and drivers serve to simulate components interactions enabling isolated analysis (GeeksforGeeks, 2023b). Test completion metrics statistically examine the breadth and depth across established test base providing measure of progress. Functional testing offers assurance in the system working correctly before dedicated user validation.

5.2.2.3. Integration Testing

Integration testing is a critical element in the software development life cycle that systematically joins independently validated software modules and components. This technique tries to evaluate the integrated system, uncovering any faults in its interactions and interfaces (*Integration Testing - Javatpoint*, n.d.). A methodical testing methodology is necessary, commencing with the investigation of interactions between modules that are focused on the same application layer before proceeding across tiers. Test complexity grows steadily, starting from API integrations and concluding in user interface integrations, leveraging industry-standard stub and mock object methodologies (Katalon, 2023).

The effectiveness of integration testing relies primarily on documented test cases and processes that comply with component specifications. These tests check coupled workflows, error handling systems, and transitions across system layers. To enhance the testing environment, simulations employing middleware mimic links with external systems, permitting full subsystem evaluations. Dedicated metrics are applied to analyze the coverage across integrated modules, providing insights into stability through successful executions. Overall, integration testing serves to cumulatively validate modular interactions, preventing the spread of undetected lower-level flaws while eventually proceeding towards performance and reliability testing (Wikipedia contributors, 2024b).

5.3.Performance Testing

Performance testing focuses on dependability, reaction times, scalability, and efficient resource consumption when systems experience expected real-world concurrent loads and data quantities (Lteif, 2023). Volume testing analyzes maximum throughput whereas stress testing evaluates behavior under excessive prolonged loads nearing breakpoint. Spike testing replicates abrupt spikes in traffic arising from events whereas soak testing evaluates durability.

Scalability testing incrementally magnifies people, data dimensions and complexity to gauge peak capabilities ("Performance Testing and Reporting," 2008). Industry standard technologies like JMeter script tests based on production usage patterns and provide reports on key parameters like response times, concurrent connections, and memory utilization at each level. The outcomes guide engineering efforts on code savings, bandwidth increases, computing upgrades and SQL optimizations needed for fulfilling business expansion requirements without performance lags.

5.4.Back-End Testing

Back-end testing focuses on evaluating the server-side components of a software system comprising database, APIs, security layers, communication modules and configuration routines (Hamilton, 2023a). The scope encompasses testing database schemas, queries, triggers, and processes employing SQL statements to minimize data loss or corruption (Lytvynenko, 2022). API testing auto-generates test cases for parameters, request types and endpoint flow to validate integration integrity.

Rigorous authentication, access control and encryption testing safeguards security by revealing dangers including injections, data breaches and violations. Network testing assesses optimal data routing and failover providing continued connectivity across remote components. Configuration testing validates application behavior matching to specific settings across instances and environments. Test metrics assess executions, assertions and code coverage particular to backstage functions indicating robustness (Vijay, 2023b). Back-end testing creates confidence in hidden yet crucial software operations supporting front-end needs.

5.5.Schema Testing

Schema testing assesses the integrity of the logical database model and its physical implementation concentrating on data structures, constraints, transitions and interactions (*IBM Documentation*, n.d.). The ANSI/ISO SQL standard guides test case design to affirm correct execution of DDL statements defining column datatypes, primary/foreign keys, indexes, defaults, and rules (Weaver et al., 2019). Insert and select query statements supply relevant test data sets to evaluate accuracy across entities and rows.

5.5.1. Table and Column Testing

Table testing checks accurate formation of database tables whereas column testing examines comprehensive integrity constraints placed on each column for data types, field lengths, not null, unique, primary key settings as per specifications. SELECT and INSERT test statements try insertion of valid and invalid data sets analyzing suitable processing.

5.5.2. Index Testing

Index testing analyzes the access speed and selectivity improvement of optimized indexes developed to improve performance of table join and search queries. Analysis of index column order, duplication and partitioning is quantified through execution plans and trace flags utilizing customized test query suites.

5.5.3. Trigger Testing

Database trigger testing confirms the conditional execution logic and integrity enforcement activities across transactions that affect tables across app workflows. Test scenarios invoke triggering statement types before and after insert, update and delete operations assessing ripple effects.

5.5.4. Stored Procedure Testing

Procedure testing conducts parameterized test cases with varied inputs to examine execution pathways, variable assignments, cursor iterations, error handling correctness and optimizations. Integration testing calls procedures from apps assessing end-to-end flows.

5.5.5. Database Server Validation Testing

Tuning the DB server entails testing memory, security, concurrency, blocking, deadlocks, replication limitations and failovers to confirm optimal configurations, load balancing and isolations that conform to application response needs.

5.6.Additional Tests

While functional and non-functional testing ensures system validity, additional methodologies enable rounded evaluations supporting deployment decisions and sustaining application stability (Sommerville, 2011). User acceptance testing brings in business user views examining procedures, ease and output usefulness based on success criteria early determined. Training trains users dealing with existing or new skills through guided task-based simulations monitoring knowledge transfer efficacy. Ongoing security testing presents threats and vulnerabilities investigating system robustness supporting continual hardening efforts alongside design improvements. Analyzing logs for utilization trends, monitoring uptime and controlling technical debt incurred forms part of maintenance testing methods suited to incremental delivery models.

5.6.1. User Acceptance Testing

User acceptance testing involves final business users to check that quickly implemented technology corresponds with true needs driving adherence and adoption goals through real-world case walkthroughs and scenario assessments giving actionable feedback.

5.6.2. User Training

Effective user training ensures system capabilities convert into productivity outcomes after deployment. Training test modules focus on task-based simulations reflecting essential application scenarios and monitoring time, errors and recall (Galitz, 2007). Testing methods comprise information organization, conceptual model explanations, demonstrations, use case walkthroughs, individual to group hands-on exercises and quizzes assessing performance. Pre-training and post-training assessments measure knowledge transfer offering meaningful inputs on enhancing learning content, structure, delivery, and real-world alignment. Surveys record qualitative reactions defining satisfaction that promotes workplace adoption while long term evaluations examine maintained proficient utilization.

5.6.3. Security & Maintenance

Proactive security testing shields systems against escalating cyber threats. Testing offers risks and vulnerabilities to examine system robustness, attack resilience and backup efficacy (Greiler et al., 2012). Rapid penetration testing hunts for holes in firewall setups, input validations and encryption coverage pushing hardening solutions. Monitoring intrusion detection trends, reviewing controls and tech debt backlogs constitute system maintenance alongside version upgrades and dependency updates. Analyzing usage trends drives performance optimization and retirement of older capabilities. Regression testing checks continuing stability assuring modifications do not introduce new issues. Security and maintenance testing uphold systems value delivering.

CHAPTER SIX
Conclusion, Lessons Learnt and Recommendations for Future Enhancement

Chapter Summary

Chapter Six concludes the research by recapping primary findings showing how automating attendance tracking enhanced administrative efficiency, outlining valuable takeaways like the importance of clear requirements, documentation, agile delivery, user-centric design and comprehensive testing for guiding future efforts, and presenting recommendations to refine interfaces, strengthen security, expand analytics, research adaptive policies, pursue system integrations and conduct longitudinal studies to maximize benefits over time. It proposes continual enhancements across areas through ongoing investigations and modifications attentive to developing needs to advance attendance modernization. The multi-faceted synthesis of outcomes, insights and realistic next steps comes full circle in contributing to both knowledge accumulation and roadmaps for continuing progress.

6.1 Conclusion

The inception of the student attendance tracking system intended to modernize participation monitoring leveraging facial recognition capabilities within a tailored mobile application. The automation of manual processes promised efficiencies for administrators while providing actionable analytics. Adopting agile delivery with continuous testing and refinement centered the project on user needs for intuitive interactions. Though adapting to new complexities caused occasional timeline shifts, iterative reprioritization kept development aligned to core goals.

In closing, the solution demonstrates the value of deeply understanding end-user perspectives and co-evolving technology alongside their changing expectations. Further customizations like enhanced security, refined interfaces and extended integrations can underscore that commitment. Expanding research on emerging policies and measuring long-term impacts can reveal additional ways to maximize outcomes from this centralized attendance platform. Prioritizing experience and outcomes over specifications signifies the project’s ultimate duty to improve processes for the campus community.

6.2 Lessons Learnt

i. **Clear Requirement Communication**-Defining clear and comprehensive requirements early through stakeholder interviews was important for designing a system that addressed the actual needs. Continuing engagement ensured the solution evolved to fully meet objectives.

ii. **Project Documentation for Knowledge Transfer**-Thorough documentation of processes, decisions, designs, and tests facilitated effective knowledge sharing within the team. This supported continuity as team members changed over time.

iii. **Project Technology Stack Selection Process**-Carefully evaluating technology options against factors like feasibility, costs and timelines was important. This allowed selecting stacks like cloud infra and cross-platform mobile frameworks for optimized implementation.

iv. **Adopting Agile Methodologies**-Utilizing agile principles with rapid iterations, user feedback loops and flexibility enabled timely delivery while still adapting to new insights. This balanced responsiveness with stability.

v. **Emphasizing User-Centric Design**-Focusing on usability and meeting end-user workflows through design drove uptake. User research uncovered needs beyond initial ideas for a solution people will readily adopt.

vi. **Prioritizing Data Security and Compliance** -Addressing governance standards and regulations upfront ensured sensitive data collection and usage aligned with expectations to safeguard integrity and build trust.

vii. **Planning Thorough Testing Regimens**-Rigorous testing across all areas found defects early, preventing post-launch issues. It validated every component's and integrated system's quality and dependability.

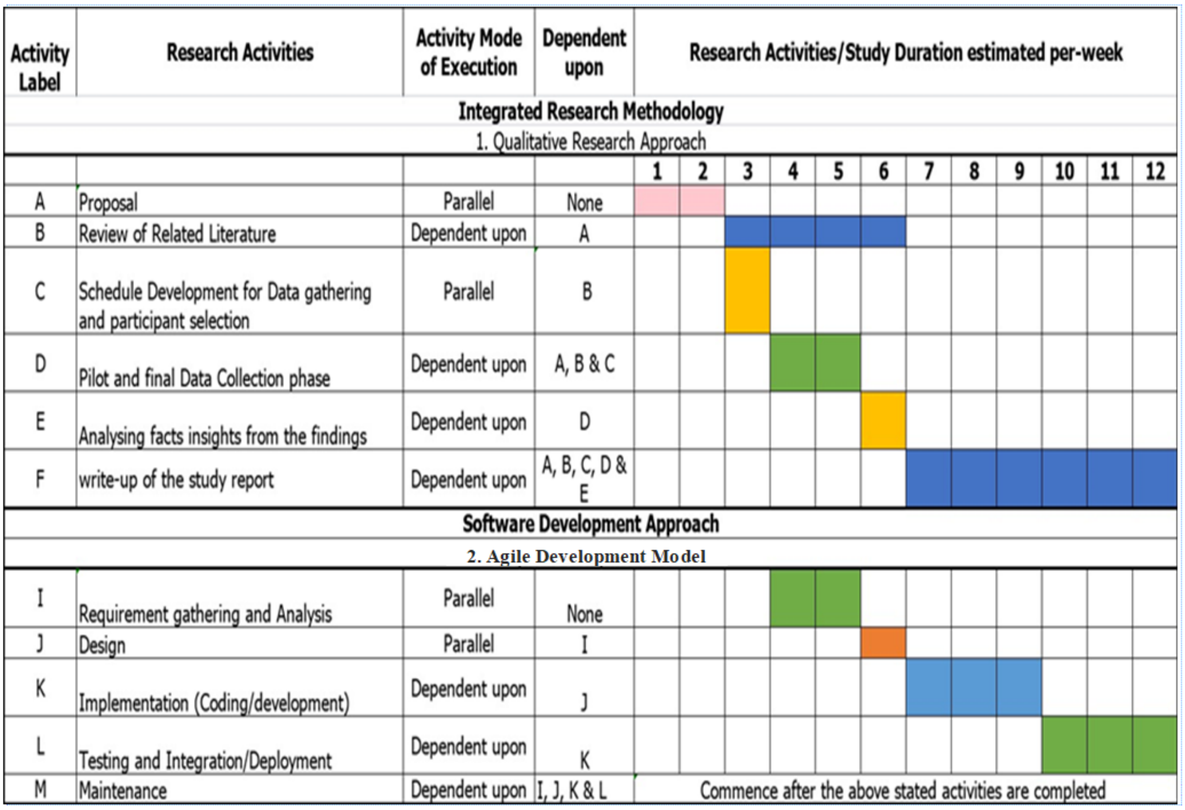
6.3 Recommendations for Future System Enhancements

A range of future enhancements can deepen impacts and preserve optimal value from ongoing attendance upgrading efforts. Conducting additional user testing studies to develop mobile interfaces and workflows can increase accessibility, performance and ease-of-use over time depending on usage input. Enhancing security methods with stronger encryption, access limits and activity audit trails can further reinforce sensitive student data. Expanding built-in analytical tools can give deeper insights from longitudinal attendance trends. Exploring connection with academic systems like learning management platforms can synchronize essential student records.

Evaluating adaptive attendance strategies adapted to different programs as well as testing across more mobile devices and varying networks might increase relevance. Developing focused training content and job aids appropriate to varied users can smooth adoption. Assessing long-term metrics on parameters like retention, achievement and engagement by monitoring pilot groups provides substantial indicator of policy effects. Planning for future campus expansions by estimating scalability demands allows wise growth. Pursuing ongoing usability research and harmonizing with developing experience expectations supports solution positioning. Overall, constant modifications on numerous parameters can maximize benefits from the attendance platform.

APPENDIX - A:

Research Study Activity Gantt Chat

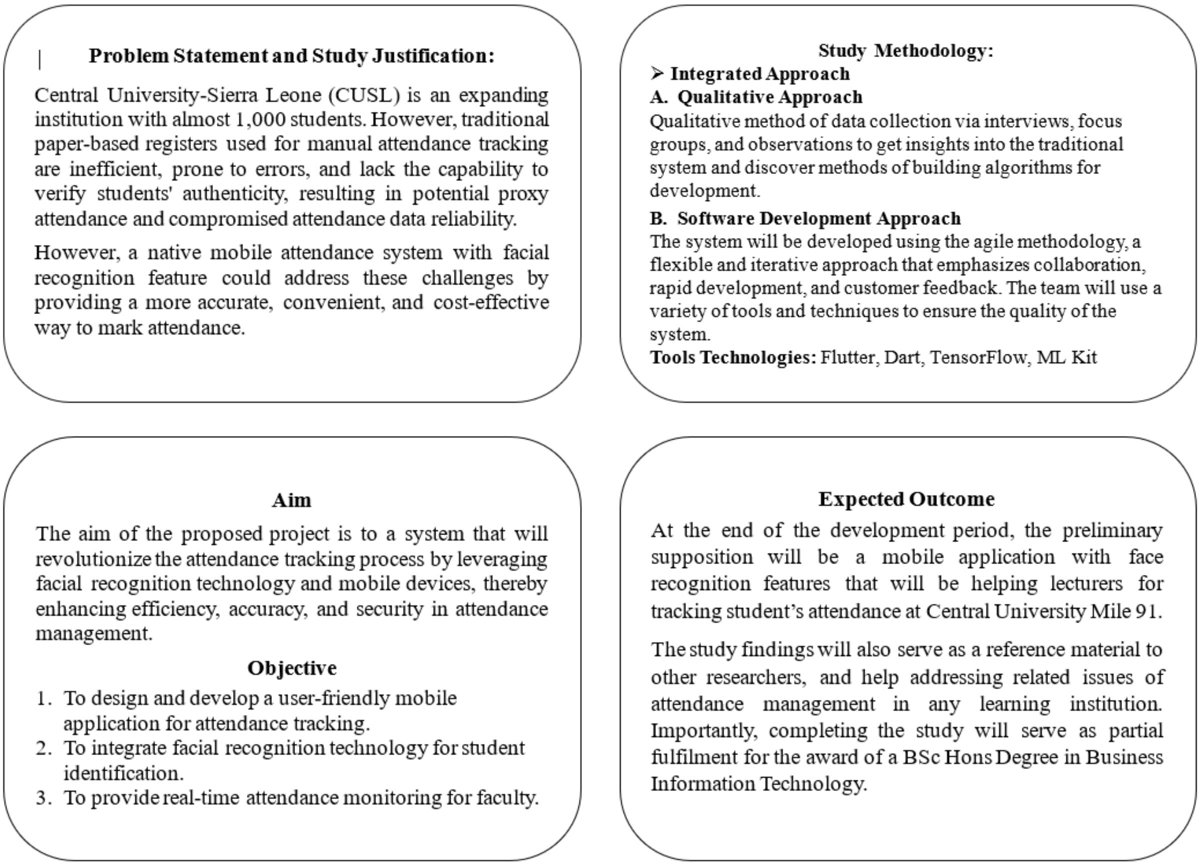


Activity Label	Research Activities	Activity Mode of Execution	Dependent upon	Research Activities/Study Duration estimated per-week											
Integrated Research Methodology															
1. Qualitative Research Approach															
				1	2	3	4	5	6	7	8	9	10	11	12
A	Proposal	Parallel	None	🟥	🟥										
B	Review of Related Literature	Dependent upon	A			🟦	🟦	🟦	🟦						
C	Schedule Development for Data gathering and participant selection	Parallel	B			🟨									
D	Pilot and final Data Collection phase	Dependent upon	A, B & C				🟩	🟩							
E	Analysing facts insights from the findings	Dependent upon	D						🟨						
F	write-up of the study report	Dependent upon	A, B, C, D & E							🟦	🟦	🟦	🟦	🟦	🟦
Software Development Approach															
2. Agile Development Model															
I	Requirement gathering and Analysis	Parallel	None				🟩	🟩							
J	Design	Parallel	I						🟧						
K	Implementation (Coding/development)	Dependent upon	J							🟦	🟦	🟦			
L	Testing and Integration/Deployment	Dependent upon	K										🟩	🟩	🟩
M	Maintenance	Dependent upon	I, J, K & L	Commence after the above stated activities are completed											

Source: *Study material September (2023)*

APPENDIX - B:

Research Study Quad Chart



| **Problem Statement and Study Justification:**

Central University-Sierra Leone (CUSL) is an expanding institution with almost 1,000 students. However, traditional paper-based registers used for manual attendance tracking are inefficient, prone to errors, and lack the capability to verify students' authenticity, resulting in potential proxy attendance and compromised attendance data reliability.

However, a native mobile attendance system with facial recognition feature could address these challenges by providing a more accurate, convenient, and cost-effective way to mark attendance.

Study Methodology:

- ➢ **Integrated Approach**

A. Qualitative Approach

Qualitative method of data collection via interviews, focus groups, and observations to get insights into the traditional system and discover methods of building algorithms for development.

B. Software Development Approach

The system will be developed using the agile methodology, a flexible and iterative approach that emphasizes collaboration, rapid development, and customer feedback. The team will use a variety of tools and techniques to ensure the quality of the system.

Tools Technologies: Flutter, Dart, TensorFlow, ML Kit

Aim

The aim of the proposed project is to a system that will revolutionize the attendance tracking process by leveraging facial recognition technology and mobile devices, thereby enhancing efficiency, accuracy, and security in attendance management.

Objective

1. To design and develop a user-friendly mobile application for attendance tracking.
2. To integrate facial recognition technology for student identification.
3. To provide real-time attendance monitoring for faculty.

Expected Outcome

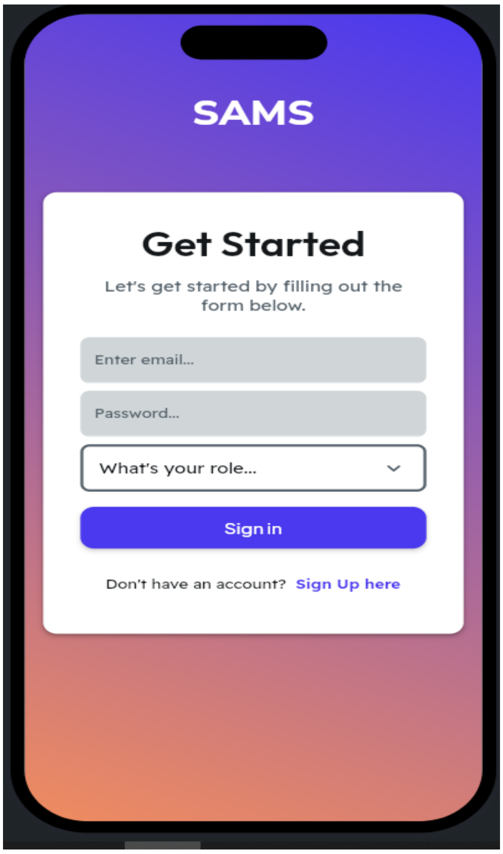
At the end of the development period, the preliminary supposition will be a mobile application with face recognition features that will be helping lecturers for tracking student's attendance at Central University Mile 91.

The study findings will also serve as a reference material to other researchers, and help addressing related issues of attendance management in any learning institution. Importantly, completing the study will serve as partial fulfilment for the award of a BSc Hons Degree in Business Information Technology.

Source: *Study material September (2023)*

APPENDIX - C:

System Login Page



SAMS

Get Started

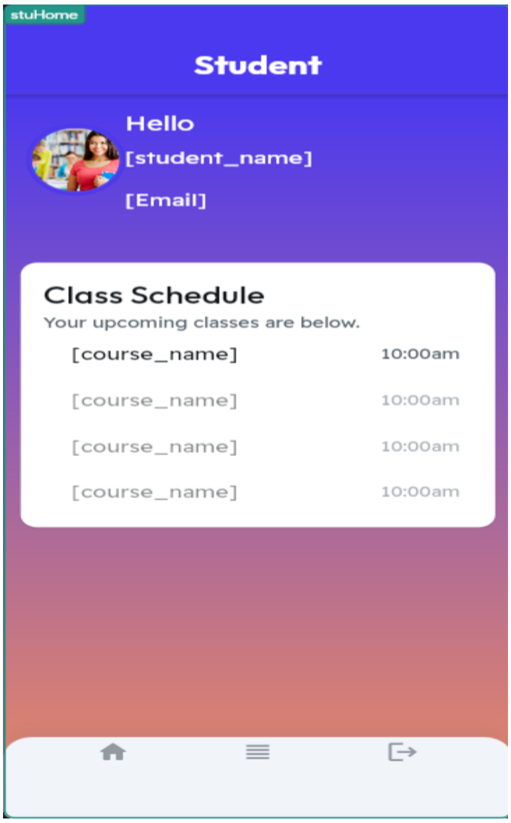
Let's get started by filling out the form below.

Source: *Study material September (2023)*

```
import '/auth/supabase_auth/auth_util.dart';
import '/flutter_flow/flutter_flow_animations.dart';
import '/flutter_flow/flutter_flow_drop_down.dart';
import '/flutter_flow/flutter_flow_theme.dart';
import '/flutter_flow/flutter_flow_util.dart';
import '/flutter_flow/flutter_flow_widgets.dart';
import '/flutter_flow/form_field_controller.dart';
import 'package:flutter/gestures.dart';
import 'package:flutter/material.dart';
import 'package:flutter/scheduler.dart';
import 'package:flutter/services.dart';
import 'package:flutter_animate/flutter_animate.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';
import 'sign_in_login_model.dart';
export 'sign_in_login_model.dart';
class SignInLoginWidget extends StatefulWidget {
  const SignInLoginWidget({Key? key}) : super(key: key);
  @override
  _SignInLoginWidgetState createState() => _SignInLoginWidgetState();
```

APPENDIX - D:

Student Dashboard



Hello
[student_name]
[Email]

Class Schedule
Your upcoming classes are below.

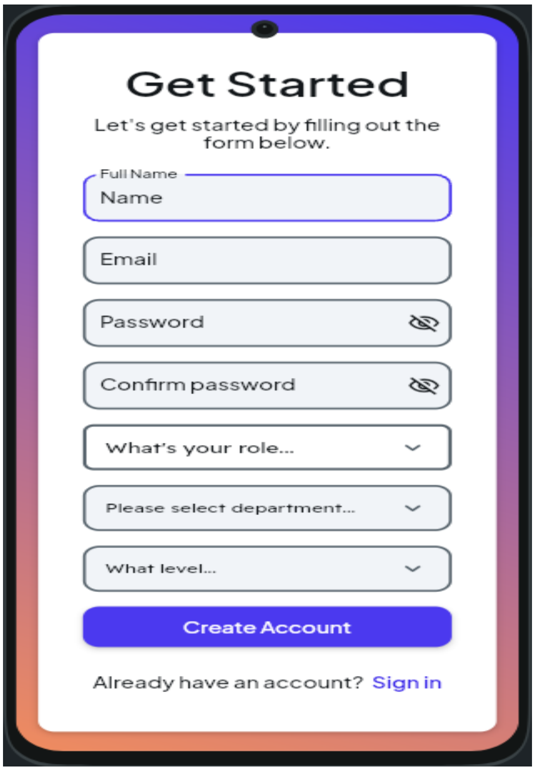
[course_name] 10:00am
[course_name] 10:00am
[course_name] 10:00am
[course_name] 10:00am

Source: Study material September (2023)

```
import '/auth/supabase_auth/auth_util.dart';
import '/backend/backend.dart';
import '/flutter_flow/flutter_flow_icon_button.dart';
import '/flutter_flow/flutter_flow_theme.dart';
import '/flutter_flow/flutter_flow_util.dart';
import '/flutter_flow/flutter_flow_widgets.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';

import 'stu_home_page_model.dart';
export 'stu_home_page_model.dart';

class StuHomePageWidget extends StatefulWidget {
  const StuHomePageWidget({Key? key}) : super(key: key);

  @override
  _StuHomePageWidgetState createState() => _StuHomePageWidgetState();
}
```

APPENDIX - E:

Create Account View



Get Started

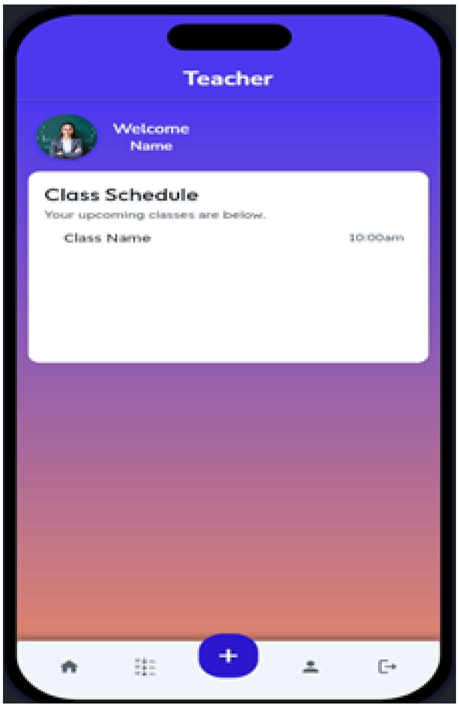
Let's get started by filling out the form below.

Source: *Study material September (2023)*

```
import '/auth/supabase_auth/auth_util.dart';
import '/backend/supabase/supabase.dart';
import '/flutter_flow/flutter_flow_animations.dart';
import '/flutter_flow/flutter_flow_drop_down.dart';
import '/flutter_flow/flutter_flow_theme.dart';
import '/flutter_flow/flutter_flow_util.dart';
import '/flutter_flow/flutter_flow_widgets.dart';
import '/flutter_flow/form_field_controller.dart';
import 'package:flutter/gestures.dart';
import 'package:flutter/material.dart';
import 'package:flutter/scheduler.dart';
import 'package:flutter/services.dart';
import 'package:flutter_animate/flutter_animate.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';

import 'create_account_model.dart';
export 'create_account_model.dart';

class CreateAccountWidget extends StatefulWidget {
  const CreateAccountWidget({Key? key}) : super(key: key);
```

APPENDIX-F

Teacher Dashboard



Class Schedule

Your upcoming classes are below.

Class Name 10:00am

Source: *Study material September (2023)*

```
import '/auth/supabase_auth/auth_util.dart';
import '/flutter_flow/flutter_flow_icon_button.dart';
import '/flutter_flow/flutter_flow_theme.dart';
import '/flutter_flow/flutter_flow_util.dart';
import '/flutter_flow/flutter_flow_widgets.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:provider/provider.dart';
import 'teach_home_model.dart';
export 'teach_home_model.dart';
class TeachHomeWidget extends StatefulWidget {
  const TeachHomeWidget({Key? key}) : super(key: key);
  @override
  _TeachHomeWidgetState createState() => _TeachHomeWidgetState();
}
class _TeachHomeWidgetState extends State<TeachHomeWidget> {
  late TeachHomeModel _model;
  final scaffoldKey = GlobalKey<ScaffoldState>();
```

REFERENCES

Nordin, N. and Mohd Fauzi, N.H. (2020). A Web-Based Mobile Attendance System with Facial Recognition Feature. *International Journal of Interactive Mobile Technologies (iJIM)*, 14(05), p.193. https://doi.org/10.3991/ijim.v14i05.13311.

Kunda, D., &Chishimba, M. (2018). A survey of android mobile phone authentication schemes. Mobile Networks and Applications, 1-9. https://doi.org/10.1007/s11036-018-1099-7

Ong, Y.S. (2016) "Reframing non-attendance of students: case study of a Malaysia private university," European Journal of Education Studies [Preprint]. Available at: https://doi.org/10.46827/ejes.v0i0.64

Pani, P.K. and Kishore, P. (2016) "Absenteeism and performance in a quantitative module A quantile regression analysis," *Journal of Applied Research in Higher Education* [Preprint]. Available at: https://doi.org/10.1108/jarhe-05-2014-0054

Ujan, I.A. and Ismaili, I.A. (2011) *Biometric attendance system*. Available at: https://doi.org/10.1109/iccme.2011.5876792.

Abstract summary of the journal "The digital transformation of education: connecting schools, empowering learners" by UNESCO International Institute for Information Technologies in Education (IITE) and Microsoft (2016):

Li, L., Mu, X., Li, S., & Peng, H. (2020). A review of face recognition technology. *IEEE Access*, *8*, 139110–139120. https://doi.org/10.1109/access.2020.3011028.

Azhaguraj, R., Kumar, P. A., Kadalarasan, S., Karthick, K., & Shunmugalakshmi, G. (2022). Smart attendance marking system using face recognition. *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*. https://doi.org/10.1109/icoei53556.2022.9776879.

Hashem Al-Ghaili. (2019, July 23). *How does facial recognition work?* [Video]. YouTube. https://www.youtube.com/watch?v=agGEDdj05U0.

Strickley, A. (2020). Attendance records, educational management. In *Springer eBooks* (pp. 195–202). https://doi.org/10.1007/978-3-030-10576-1_117

Sawhney, S., Kacker, K., Jain, S., Singh, S., & Garg, R. (2019). Real-Time Smart Attendance System using Face Recognition Techniques. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. https://doi.org/10.1109/confluence.2019.8776934

Qasim, I., Azam, F., Anwar, M. W., Tufail, H., & Qasim, T. (2018). Mobile User Interface Development Techniques: A Systematic Literature Review. *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. https://doi.org/10.1109/iemcon.2018.8614764

Yang, S., Lee, T., & Feng, T. (2018). User Experience of Mobile Application's Interface. *Conference: The 5th Multidisciplinary International Social Networks Conference*. https://doi.org/10.1145/3227696.3227698

Goel, S., Nagpal, R., & Mehrotra, D. (2017). Mobile Applications Usability Parameters: Taking an insight view. In *Lecture notes in networks and systems* (pp. 35–43). https://doi.org/10.1007/978-981-10-3932-4_4

Gasmi, I., Soui, M., Chouchane, M., & Abed, M. (2017). Evaluation of mobile interfaces as an optimization problem. *Procedia Computer Science*, *112*, 235–248. https://doi.org/10.1016/j.procs.2017.08.234

Zohud, T., & Zein, S. (2021). Cross-Platform Mobile App Development in Industry: A Multiple Case-Study. *International Journal of Computing*, 46–54. https://doi.org/10.47839/ijc.20.1.2091

Pinto, C. M., & Coutinho, C. (2018). From Native to Cross-platform Hybrid Development. *2018 International Conference on Intelligent Systems (IS)*. https://doi.org/10.1109/is.2018.8710545

Singh, R., & Singh, R. (2023, January 12). Top 10 Best Cross-Platform App Development Frameworks for 2023. *Richestsoft*. https://richestsoft.com/blog/cross-platform-app-development-frameworks/

Lano, K., & Tehrani, S. Y. (2023). Mobile Application Architectures. In *Undergraduate topics in computer science* (pp. 99–136). https://doi.org/10.1007/978-3-031-44143-1_6

Alatalo, P., Järvenoja, J., Karvonen, J., Keronen, A., & Kuvaja, P. (2007). Mobile Application Architectures. In *Springer eBooks* (pp. 572–586). https://doi.org/10.1007/3-540-36209-6_47

Vaidya, R. (2023, October 17). *Mobile App Architecture – complete guide on basics to building in 2023*. Elluminati Inc. https://www.elluminatiinc.com/mobile-app-architecture/

Vaidya, R. (2023b, October 17). *Mobile App Architecture – complete guide on basics to building in 2023*. Elluminati Inc. https://www.elluminatiinc.com/mobile-app-architecture/

Global App Testing (Spa Worldwide Limited) is a company registered in England under no. 07606704. (n.d.). *Mobile app deployment explained*. https://www.globalapptesting.com/mobile-app-deployment

Patel, B. (2024, January 4). Mobile App development Process in 2024 [9 steps guide]. *Homepage*. https://www.spaceotechnologies.com/blog/mobile-app-development-process-steps/

Kolodiy, R. (2023, October 18). *Best Application deployment Strategies in 2023 — TechMagic*. Blog | TechMagic. https://www.techmagic.co/blog/best-application-deployment-strategies/

Rasmussen, R. (2022, April 20). *How to write an Effective Data Acquisition Strategy: An In-Depth Guide*. UrbanLogiq. https://urbanlogiq.com/how-to-write-an-effective-data-acquisition-strategy/

Mobile Machine Learning. (2023, December 12). *Face Recognition & Detection in Flutter - Build attendance & security systems for Android & IOS* [Video]. YouTube. https://www.youtube.com/watch?v=3Uv0zvE_gNM

What is data compliance? | IBM. (n.d.). https://www.ibm.com/topics/data-compliance

Babalola, O. (2023). Data Protection Legal Regime and Data Governance in Africa: An Overview. In *Springer eBooks* (pp. 71–96). https://doi.org/10.1007/978-3-031-24498-8_4

Wikipedia contributors. (2024, January 3). *Flutter (software)*. Wikipedia. https://en.wikipedia.org/wiki/Flutter_(software)

Cost of a data breach 2023 | IBM. (n.d.). https://www.ibm.com/reports/data-breach?utm_content=SRCWW&p1=Search&p4=43700077724063991&p5=e&gclid=CjwKCAjw1t2pBhAFEiwA_-A-NEOKDgJ74mj_e6FLr44Df-SNhLmsfEeaQiM773qQ6JzfrAccZhIT3hoCF1UQAvD_BwE&gclsrc=aw.ds

29119-4-2015 - ISO/IEC/IEEE International Standard - Software and systems engineering--Software testing--Part 4: Test techniques. (2015, December 8). IEEE Standard | IEEE Xplore. https://ieeexplore.ieee.org/servlet/opac?punumber=7346373

B, B. A. (2021, June 18). Mobile App Development Process | Infographic guide, stages, Life cycle plan | JoinSoft. *JoinSoft*. https://joinsoft.com/blog/mobile-app-development-process/

Bose, S. (2023a, March 16). *Front end testing: A Beginner’s guide | BrowserStack*. BrowserStack. https://www.browserstack.com/guide/front-end-testing

Bose, S. (2023b, August 17). *Functional Testing : Definition, Types & Examples | BrowserStack*. BrowserStack. https://www.browserstack.com/guide/functional-testing

CCNC 2021: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC): 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC) Took Place 9-13 January 2021 in Virtual. (2021).

Contributor, T. (2023, May 1). *unit testing*. Software Quality. https://www.techtarget.com/searchsoftwarequality/definition/unit-testing#:~:text=Unit%20testing%20is%20a%20software,tests%20during%20the%20development%20process.

Crnkovic, I. (2018). *2018 ACM/IEEE 40th International Conference on Software Engineering: Companion Proceeedings: ICSE-Companion 2018 : 27 May-3 June 2018, Gothenburg, Sweden : Proceedings*.

Dizdar, A. (2023, November 14). *Unit testing: definition, examples, and critical best practices*. Bright Security. https://brightsec.com/blog/unit-testing/

GeeksforGeeks. (2023a, November 16). *Unit testing software testing*. https://www.geeksforgeeks.org/unit-testing-software-testing/

GeeksforGeeks. (2023b, December 6). *Functional testing software testing*. https://www.geeksforgeeks.org/software-testing-functional-testing/

Hamilton, T. (2023a, December 6). *Backend testing Tutorial (Examples)*. Guru99. https://www.guru99.com/what-is-backend-testing.html

Hamilton, T. (2023b, December 9). *What is Front End Testing?* Guru99. https://www.guru99.com/frontend-testing.html

Hu, Y., Chen, J., Yang, L., Yue, H., & Du, X. (2021). System Implementation and Testing for Unified Frequency Control Strategy Adapted to Frequency Regulation Auxiliary Service Market in Southern China. *2021 International Conference on Control Science and Electric Power Systems (CSEPS)*. https://doi.org/10.1109/cseps53726.2021.00050

IBM documentation. (n.d.). https://www.ibm.com/docs/en/rational-clearquest/9.0.0?topic=schemas-testing

Integration Testing - javatpoint. (n.d.). www.javatpoint.com. https://www.javatpoint.com/integration-testing

Kasthuri, M. (2018). System implementation and testing of proposed language Independent Stemmer. *International Journal of Computer Sciences and Engineering*, *6*(11), 265–273. https://doi.org/10.26438/ijcse/v6i11.265273

Katalon. (2023, January 31). What is integration testing and its types? *katalon.com*. https://katalon.com/resources-center/blog/integration-testing

Li, H., Hu, Z., Xu, R., & Wu, X. (2021). Design and implementation of a GNSS navigation terminal automation test system. *2021 33rd Chinese Control and Decision Conference (CCDC)*. https://doi.org/10.1109/ccdc52312.2021.9602403

Lteif, G. (2023, November 18). *Stress, load, and performance testing in software and IT systems*. https://softwaredominos.com/home/software-design-development-articles/stress-load-performance-testing-in-software-and-it-systems/

Lytvynenko, D. (2022, November 21). Backend testing: the definitive guide — MindK Blog. *Web and Mobile App Development Company — MindK.com*. https://www.mindk.com/blog/backend-testing/

Performance testing and reporting. (2008). In *Springer eBooks* (pp. 81–97). https://doi.org/10.1007/978-0-387-77627-9_5

Testim. (2020, May 27). Front End testing: A complete conceptual overview. *AI-driven E2E automation with code-like flexibility for your most resilient tests*. https://www.testim.io/blog/front-end-testing-complete-overview/

Vijay. (2023a, October 7). *How to perform backend testing*. Software Testing Help. https://www.softwaretestinghelp.com/how-to-perform-backend-testing/

Vijay. (2023b, October 7). *How to perform backend testing*. Software Testing Help. https://www.softwaretestinghelp.com/how-to-perform-backend-testing/

Weaver, V. T., Ramos, J., & Nielsen, S. W. (2019). Composability Applications for Test System Development. *2019 IEEE AUTOTESTCON*. https://doi.org/10.1109/autotestcon43700.2019.8961074

What is Functional Testing? Types & Examples | OpenText. (n.d.). OpenText. https://www.opentext.com/what-is/functional-testing#:~:text=Functional%20testing%20is%20a%20type,with%20the%20end%20user&apos;s%20expectations.

Wikipedia contributors. (2024a, January 3). *Flutter (software)*. Wikipedia. https://en.wikipedia.org/wiki/Flutter_(software)

Wikipedia contributors. (2024b, January 6). *Integration testing*. Wikipedia. https://en.wikipedia.org/wiki/Integration_testing

Xu, Z., Peng, K., Li, Z., Feng, Y., & Peng, Y. (2022). Design and implementation of automatic test system for multi-cabin joint test of space station. *2022 34th Chinese Control and Decision Conference (CCDC)*. https://doi.org/10.1109/ccdc55256.2022.10034298

Chandramohan, J., Nagarajan, R., Kumar, M., Dineshkumar, T., G.Kannan, & Prakash, R. (2017). Attendance monitoring system of students based on biometric and GPS tracking system. *International Journal of Advanced Engineering, Management and Science*, *3*(3), 241–246. https://doi.org/10.24001/ijaems.3.3.16

A framework for automated quality assessment of software requirement specification based on part-of-speech tagging, multi-agent k-means clustering and case-based reasoning - UTHM Institutional Repository. (n.d.). http://eprints.uthm.edu.my/8438/

Alatalo, P., Järvenoja, J., Karvonen, J., Keronen, A., & Kuvaja, P. (2007). Mobile Application Architectures. In *Springer eBooks* (pp. 572–586). https://doi.org/10.1007/3-540-36209-6_47

Ali, N., & Lai, R. (2015). A method of software requirements specification and validation for global software development. *Requirements Engineering*, *22*(2), 191–214. https://doi.org/10.1007/s00766-015-0240-4

Babalola, O. (2023). Data Protection Legal Regime and Data Governance in Africa: An Overview. In *Springer eBooks* (pp. 71–96). https://doi.org/10.1007/978-3-031-24498-8_4

Bussa, S., Mani, A., Bharuka, S., & Kaushik, S. (2020). Smart Attendance System using OPENCV based on Facial Recognition. *International Journal of Engineering Research and Technology*, *V9*(03). https://doi.org/10.17577/ijertv9is030122

Chen, S., Liu, Y., Gao, X., & Han, Z. (2018). MobileFaceNets: Efficient CNNs for accurate Real-Time face verification on mobile devices. In *Lecture Notes in Computer Science* (pp. 428–438). https://doi.org/10.1007/978-3-319-97909-0_46

Cost of a data breach 2023 | IBM. (n.d.). https://www.ibm.com/reports/data-breach?utm_content=SRCWW&p1=Search&p4=43700077724063991&p5=e&gclid=CjwKCAjw1t2pBhAFEiwA_-A-NEOKDgJ74mj_e6FLr44Df-SNhLmsfEeaQiM773qQ6JzfrAccZhIT3hoCF1UQAvD_BwE&gclsrc=aw.ds

Deng, X., & Zhang, R. (2019). Smart Learning Environment: A Case on the Construction of Smart Classrooms in Colleges and Universities in Guangzhou. *Smart Learning*

Environment: A Case on the Construction of Smart Classrooms in Colleges and Universities in Guangzhou. https://doi.org/10.1109/iset.2019.00062

Gasmi, I., Soui, M., Chouchane, M., & Abed, M. (2017). Evaluation of mobile interfaces as an optimization problem. *Procedia Computer Science*, *112*, 235–248. https://doi.org/10.1016/j.procs.2017.08.234

Geetha, K., Sreerambabu, J., & Santhosh, N. (2022a). Face recognition based Automated Student Attendance system. *International Journal for Research in Applied Science and Engineering Technology*, *10*(8), 965–969. https://doi.org/10.22214/ijraset.2022.46316

Geetha, K., Sreerambabu, J., & Santhosh, N. (2022b). Face recognition based Automated Student Attendance system. *International Journal for Research in Applied Science and Engineering Technology*, *10*(8), 965–969. https://doi.org/10.22214/ijraset.2022.46316

Global App Testing (Spa Worldwide Limited) is a company registered in England under no. 07606704. (n.d.). *Mobile app deployment explained*. https://www.globalapptesting.com/mobile-app-deployment

Goel, S., Nagpal, R., & Mehrotra, D. (2017). Mobile Applications Usability Parameters: Taking an insight view. In *Lecture notes in networks and systems* (pp. 35–43). https://doi.org/10.1007/978-981-10-3932-4_4

Hashem Al-Ghaili. (2019, July 23). *How does facial recognition work?* [Video]. YouTube. https://www.youtube.com/watch?v=agGEDdj05U0

Kadasah, S. F., Galil, F. M. a. A., Kolhe, B., & Shinde, S. M. (2022). *Scientific research Methodology Principles, methods, and techniques*. Book Rivers.

Kolodiy, R. (2023, October 18). *Best Application deployment Strategies in 2023 — TechMagic*. Blog | TechMagic. https://www.techmagic.co/blog/best-application-deployment-strategies/

Kuhrmann, M., Diebold, P., Münch, J., Tell, P., Trektere, K., McCaffery, F., Garousi, V., Felderer, M., Linssen, O., Hanser, E., & Prause, C. R. (2019). Hybrid Software Development approaches in Practice: A European perspective. *IEEE Software*, *36*(4), 20–31. https://doi.org/10.1109/ms.2018.110161245

Lano, K., & Tehrani, S. Y. (2023). Mobile Application Architectures. In *Undergraduate topics in computer science* (pp. 99–136). https://doi.org/10.1007/978-3-031-44143-1_6

Li, L., Mu, X., Li, S., & Peng, H. (2020). A review of face recognition technology. *IEEE Access*, *8*, 139110–139120. https://doi.org/10.1109/access.2020.3011028

McCombes, S. (2023, June 22). *What is a research methodology? | Steps &#038; tips*. Scribbr. https://qa.scribbr.com/dissertation/methodology/

ML-Music. (2018, August 8). *This Is Me - The Greatest Showman {LYRICS}* [Video]. YouTube. https://www.youtube.com/watch?v=5J29YsEfYlo

Mobile application development & its importance in your business. (2016, September 26). [Slide show]. PPT. https://www.slideshare.net/MYQSOFT/mobile-application-development-its-importance-in-your-business

Mobile Machine Learning. (2023, December 12). *Face Recognition & Detection in Flutter - Build attendance & security systems for Android & IOS* [Video]. YouTube. https://www.youtube.com/watch?v=3Uv0zvE_gNM

ÖzdemiR, D., & Ugur, M. E. (2020). MODEL PROPOSAL ON THE DETERMINATION OF STUDENT ATTENDANCE IN DISTANCE EDUCATION WITH FACE RECOGNITION TECHNOLOGY. *The Turkish Online Journal of Distance Education*, *22*(1), 19–32. https://doi.org/10.17718/tojde.849872

Pandey, P. (2015). *Research methodology: Tools & Techniques*.

Patel, B. (2024, January 4). Mobile App development Process in 2024 [9 steps guide]. *Homepage*. https://www.spaceotechnologies.com/blog/mobile-app-development-process-steps/

Patel, R., & Patel, R. (2023, December 27). *How to design mobile app architecture in 2024?* Aglowid IT Solutions. https://aglowiditsolutions.com/blog/design-app-architecture/

Pinto, C. M., & Coutinho, C. (2018). From Native to Cross-platform Hybrid Development. *2018 International Conference on Intelligent Systems (IS)*. https://doi.org/10.1109/is.2018.8710545

Prieto, R. J. (2018, June 19). The Agile Classroom: Embracing an Agile Mindset In Education. *Medium*. https://medium.com/laboratoria/the-agile-classroom-embracing-an-agile-mindset-in-education-ae0f19e801f3

Rasmussen, R. (2022, April 20). *How to write an Effective Data Acquisition Strategy: An In-Depth Guide*. UrbanLogiq. https://urbanlogiq.com/how-to-write-an-effective-data-acquisition-strategy/

Sajid, M., Hussain, R., & Usman, M. (2014). A conceptual model for automated attendance marking system using facial recognition. *A Conceptual Model for Automated Attendance Marking System Using Facial Recognition*. https://doi.org/10.1109/icdim.2014.6991407

Sawhney, S., Kacker, K., Jain, S., Singh, S., & Garg, R. (2019). Real-Time Smart Attendance System using Face Recognition Techniques. *Real-Time Smart Attendance System Using Face Recognition Techniques*. https://doi.org/10.1109/confluence.2019.8776934

Scrum Agile Process PowerPoint Charts template for presentation. (n.d.). ImagineLayout. https://imaginelayout.com/powerpoint_chart-template-11609/

Vaidya, R. (2023a, October 17). *Mobile App Architecture &#8211; complete guide on basics to building in 2023*. Elluminati Inc. https://www.elluminatiinc.com/mobile-app-architecture/

Vaidya, R. (2023b, October 17). *Mobile App Architecture &#8211; complete guide on basics to building in 2023*. Elluminati Inc. https://www.elluminatiinc.com/mobile-app-architecture/

Varadharajan, E., Dharani, R., Jeevitha, S., Kavinmathi, B., & Hemalatha, S. (2022). Automatic attendance management system using face detection. *Automatic Attendance Management System Using Face Detection*. https://doi.org/10.1109/get.2016.7916753

Wagh, P. D., Thakare, R., Chaudhari, J., & Patil, S. (2015). Attendance system based on face recognition using eigen face and PCA algorithms. *Attendance System Based on Face Recognition Using Eigen Face and PCA Algorithms*. https://doi.org/10.1109/icgciot.2015.7380478

What is data compliance? | IBM. (n.d.). https://www.ibm.com/topics/data-compliance

Xiao, J., Jiang, G., & Liu, H. (2018). A Lightweight Face Recognition Model based on MobileFaceNet for Limited Computation Environment. *EAI Endorsed Transactions on Internet of Things*, 173547. https://doi.org/10.4108/eai.28-2-2022.173547

Zohud, T., & Zein, S. (2021). Cross-Platform Mobile App Development in Industry: A Multiple Case-Study. *International Journal of Computing*, 46–54. https://doi.org/10.47839/ijc.20.1.2091