VALLEY VIEW UNIVERSITY

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCES AND ENGINEERING

A PROJECT PROPOSAL IN FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE DEGREE (BSc.) IN COMPUTER SCIENCE.

PROJECT TOPIC:

ENHANCE OF QUALITY ASSURANCE SYSTEM FOR VALLEY VIEW UNIVERSITY

BY:

EBANA NZANG MARIA AMPARO NDOHO

ID:

222CS02000813

SUPERVISOR:

MR. SAMUEL Y. SEBUABE

DATE: 12/05/2025

DEDICATION

I dedicate this research project to my beloved parents, Pedro Ebana and Asunción Nzang, whose unwavering love, guidance, and support have been the foundation of my journey. I also dedicate it to Crispín Ekang, who has stood by me with encouragement and strength throughout this endeavor. Their presence and belief in me have been a constant source of motivation and inspiration.

AKNOWLEDEMENT

I give thanks to the Almighty God for His divine wisdom, knowledge, and guidance, which enabled me to successfully complete this research project. Throughout this academic journey, I consulted a wide range of textbooks, lecture notes, journals, and online resources. My sincere gratitude goes to all the authors and contributors whose works have supported and enriched my study.

I wish to express my heartfelt appreciation to all the lecturers of Valley View University, whose dedication and tireless efforts have shaped my academic growth. I am especially grateful to my supervisor, Mr. Samuel Y. Sebuabe, for his valuable guidance and support. I also extend my appreciation to the Head of Department, Mr. Prince Owusu Amoako, whose advice and encouragement have been instrumental from the beginning of this journey. My thanks also go to the entire faculty and staff of the Department of Computing Sciences and Engineering for their continuous support.

A profound thank you goes to my friends and family whose encouragement, love, and belief in me have been unwavering. Your support has made this achievement possible, and I am truly grateful.

My utmost gratitude is extended to the Quality Assurance Department at Valley View University. This opportunity made it possible for

my dream to come true, and for that, you will forever hold a special place in my heart.

Finally, to all my classmates, thank you for the shared experiences, challenges, and growth over the past four years. Without your presence, my time at Valley View University would not have been as meaningful. May God richly bless you all.

DECLARATION

This is to declare that, the research work underlying this thesis has been carried out by the under-mention student under the supervisor. Both the Student and the supervisor certify that the work documented in this thesis is the output of the research conducted by the student as part of her final year project work fulfilment of the requirements of the Bachelor of Science in Computer Science.

STUDENT	SUPERVISOR
EBANA NZANG MARIA A.N.	MR. SAMUEL Y. SEBUABE
SIGNATURE	SIGNATURE
----------	----------
DATE	DATE
----------	----------

TABLE OF CONTENT

ABSTRACT

The quality assurance system at Valley View University faces significant challenges in managing essential processes such as Accreditations, Assessments, and Auditing. Although these processes are not entirely manual, the current approach leads to inefficiencies and complications, particularly due to cumbersome communication and document management practices. For example, the exchange of forms and feedback among stakeholders creates delays and increases the potential for errors across all areas. To address these challenges, this project proposes the development of a comprehensive University Quality Assurance Management System (UniQAMS). UniQAMS aims to automate quality assurance processes, streamline document management, enhance accreditation, assessment, and auditing procedures, and improve departmental communication within the university setting. The anticipated benefits include increased efficiency, accuracy, and transparency in all quality assurance operations.

CHAPTER ONE

GENERAL INTRODUCTION

1.0 Introduction

Quality assurance is a systematic process aimed at ensuring that products, services, or systems meet specified requirements and maintain consistency in quality (Aburizaizah, 2022).

In the context of higher education, quality assurance encompasses policies, processes, and actions aimed at maintaining and improving the academic standards and overall institutional effectiveness of universities. It plays a pivotal role in promoting accountability, continuous improvement, and credibility in educational delivery. Globally, quality assurance has become a cornerstone in higher education as institutions strive to achieve excellence amidst increasing competition and dynamic academic demands.

1.1 Subject and Field of Study

The subject of this final-year project proposal is the development of a digital system to automate and streamline the core quality assurance processes at Valley View University. This system will enhance the management of accreditation procedures, assessments, auditing, and other essential QA activities, aiming to improve efficiency and ensure compliance with institutional standards.

In terms of the field of study, this project primarily falls within the domain of software development. Software development is integral to the creation of the UniQAMS, a sophisticated system that integrates various technologies and programming languages to automate quality assurance processes and enhance efficiency within the university setting.

1.2 Problem Statement

Valley View University faces significant challenges in managing its quality assurance processes, including accreditation, assessments, and auditing, due to reliance on outdated and inefficient manual systems. These traditional methods lead to delays and make it difficult to effectively monitor, evaluate, and improve quality standards across departments. Furthermore, the lack of a centralized digital platform limits communication and collaboration between departments, causing inefficiencies. To overcome these challenges and improve overall quality management,

there is a critical need for a comprehensive, mobile application Quality Assurance Management System (UniQAMS) that will automate key processes, enhance data preciseness, and facilitate better communication and collaboration among all stakeholders.

1.3 Study Objectives

1.3.1 General Objectives

The general objective of this project is to develop a comprehensive University Quality Assurance Management System (UniQAMS) that automates and streamlines key quality assurance processes, including accreditation, assessments, and auditing, to improve efficiency, reduce errors, and enhance communication across departments at Valley View University.

1.3.2 Specific Objectives

The following are the specific objectives of the project:

- To allow quality assurance staff to create questionnaire templates for accreditation, assessments, and auditing.
- To enable quality assurance staff to schedule assessments, audits, and accreditation activities within the system.
- To implement a notification mechanism that alerts users about upcoming events, such as scheduled assessments or audits.

1.4 Background of the Study

In the modern educational landscape, the demand for quality education has intensified. Higher education institutions are under

pressure to maintain high academic standards and ensure institutional effectiveness. Quality assurance (QA) plays a critical role in meeting these demands by ensuring that universities continuously improve their educational practices and adhere to accreditation standards (Bunyamin , Alamsyah and Fadhli, 2020). For Valley View University, like many other institutions, robust QA practices are essential for maintaining accreditation and improving both educational outcomes and institutional operations (Arifudin and Jesry, 2022).

However, despite the importance of quality assurance, Valley View University faces significant challenges in managing its QA processes, particularly in the areas of accreditation, assessment, and auditing. Currently, the QA department relies on outdated and inefficient manual systems to manage critical documentation. Accreditation forms, such as Program for Reaccreditation Forms, Accreditation Forms, and Institutional Programs for Accreditation Forms, are often sent between departments in Word format for completion. This results in delays as forms are repeatedly sent back and forth to correct mistakes or missing information. These back-and-forth exchanges create inefficiencies and increase the risk of human error, ultimately slowing down the accreditation process.

In addition to accreditation challenges, auditing procedures also face similar issues. QA staff conduct audits across departments to collect data and ensure compliance with institutional standards. However, these findings are often recorded on paper instead of being entered into a centralized digital platform. Once the data has been collected, it is transcribed into reports manually, which further increases the likelihood of errors and delays. The lack of a real-time digital system to store and track audit data creates inefficiencies and hinders the ability of the QA department to provide timely feedback and actionable insights to the relevant departments.

The assessment process is also affected by similar inefficiencies. While Valley View University has a system that tracks student data, including information related to assessments, the QA department still faces difficulties in utilizing this system effectively. After completing an assessment, the QA team has to extract data from the system in Word format, analyze it manually, and generate reports to assess the performance of lecturers and identify issues faced by students. This process is both time-consuming and prone to errors, limiting the ability to make informed decisions in a timely manner.

These issues underscore the pressing need for a more efficient and streamlined approach to quality assurance at Valley View

University. The absence of a centralized digital platform to automate key QA processes not only hampers efficiency but also complicates communication and collaboration across departments. To address these challenges, there is a critical need for the development of a mobile application Quality Assurance Management System (UniQAMS). UniQAMS will automate accreditation, assessment, and auditing processes, centralize data storage, and enable real-time communication between the QA department and other university departments. By leveraging modern app technologies, UniQAMS will improve workflow efficiency, reduce the likelihood of errors, and provide more timely and accurate feedback, thereby enhancing the university's overall quality assurance practices (Elbadiansyah & Masyni).

1.5 Scope of the Study

The scope of this study focuses on the development of a mobile application University Quality Assurance Management System (UniQAMS) for Valley View University. The system aims to automate and streamline key quality assurance processes, specifically in the areas of accreditation, assessments, and auditing. It will include features such as creating reusable questionnaire templates, scheduling assessments, audits, and accreditation activities, and sending notification alerts for scheduled events. Additionally, the system will incorporate role-based access

control with robust authentication and authorization mechanisms to ensure data security.

The research will be limited to Valley View University, addressing only the specific needs of its Quality Assurance Department. It will not cover other university management areas like student registration or financial systems. Furthermore, the study will focus solely on a mobile application platform, excluding web-based platform development. The goal is to enhance communication and efficiency within the QA department, without extending to unrelated administrative functions. While report generation is recognized as a valuable addition, it will be recommended as a feature for future iterations beyond the scope of this study.

1.6 Justification of the Study

The development of the University Quality Assurance Management System (UniQAMS) is driven by the need to streamline and automate the Quality Assurance (QA) processes at Valley View University. Currently, the university's QA department faces challenges related to managing accreditation, assessments, and audits using disconnected systems that are time-consuming, error-prone, and inefficient. By implementing a mobile application system, the project aimed to significantly improve the efficiency and accuracy of these processes, ensuring timely assessments and better compliance with accreditation standards.

Furthermore, the introduction of role-based access control, robust authentication, and authorization mechanisms will enhance data security, protecting sensitive information within the QA department. While existing systems may address some aspects of university management, they are often not tailored to the specific needs of quality assurance. This study fills a critical gap by providing a solution focused solely on improving the efficiency of QA operations at Valley View University, ultimately contributing to the institution's overall academic and operational excellence.

1.7 Methodology

The methodology for this project will involve several stages, including:

- Requirements Gathering

  Engage with stakeholders, including Valley View University's Quality Assurance (QA) department and university administration, to collect and analyze requirements for the University Quality Assurance Management System (UniQAMS). The goal is to identify the essential features and functionalities needed to automate and streamline accreditation, assessment, and auditing processes within the QA department.

- Design

  The user interface (UI) of UniQAMS will bedesigned using Figma to create mockups and wireframes, ensuring a collaborative

design process that aligns with stakeholder needs. This will include defining the layout, structure, and user experience for both administrators and QA personnel.

The system's architecture and database schema will be outlined to support data management and integration, focusing on a mobile application solution to enhance communication and operational efficiency within the QA department.

- Development

  The development of UniQAMS will be based on the React Native framework, allowing for cross-platform compatibility on both iOS and Android devices.

  Firebase Authentication will be used for user authentication and authorization, while Firebase Cloud Functions will handle server-side logic for managing requests related to the QA processes.

  The front end will be implemented with React Native, using its component-based structure to build a responsive and interactive mobile interface. The back end will integrate with Express.js with Node.js for API handling if additional backend logic is required, with data managed in postgreSQL.

  Git will be used for version control to enable efficient collaboration and maintain the integrity of the code-base.

- Testing

Detox or Appium will be used for end-to-end testing, verifying the mobile app's functionality across devices to meet requirements. Unit and integration tests will be conducted on individual components and their interactions using Jest and React Native Testing Library.

User Acceptance Testing (UAT) will also be conducted with key stakeholders to ensure the app meets their expectations and requirements.

- Deployment And Support

  In order to ensure scalability and optional performance, the system will be deployed using React Native. Expo Go enables users to deploy apps without app stores, so users of the proposed system will be able to scan a QR code from within the Expo Go app to download and run the app instantly on both iOS and Android.

  Training sessions will be held for users within the QA department to ensure smooth adoption of UniQAMS. Post-deployment, ongoing technical support and maintenance will be provided, using GitHub for issue tracking and collaboration to ensure system reliability and continuous improvement.

1.8 Expected Results of the Study

The expected results of this study are:

Improved Efficiency in QA Processes: Automation of accreditation, assessments, and auditing procedures will significantly reduce manual workloads, speeding up processes that currently face delays due to inefficient communication and manual data entry.

Enhanced Data Accuracy and Integrity: By digitizing form submissions and report generation, UniQAMS will reduce the likelihood of human errors commonly associated with manual documentation, leading to more accurate quality assurance outcomes.

Better Communication and Collaboration: The system will streamline communication between the Quality Assurance (QA) department and other departments by providing a centralized platform, ensuring that all stakeholders are informed and aligned.

Improved Compliance with Accreditation Standards: Automating processes will help Valley View University maintain compliance with accreditation bodies by providing timely and accurate reports, thus improving its standing in academic evaluations.

Increased Transparency and Accountability: With role-based access control and a robust notification system, stakeholders will have clear responsibilities and real-time visibility into the status of QA activities.

1.9 Presentation of Project

The work for this proposed project will be presented in the following format:

Chapter one: General Introduction

Chapter Two: Literature Review

Chapter Three: Crystallization of the Problem

Chapter Four: Analysis of the Proposed System

Chapter Five: Detailed Analyses and Design

Chapter Six: System Implementation

Chapter Seven: System Documentation

Chapter Eight: Conclusion and Recommendation

1.10 Study Work Plan

The work is expected to be completed in twelve months.

CHAPTER TWO

LITERATURE REVIEW

This chapter includes a review of related literature on existing Quality Assurance Management Systems that are relevant to the problem, as well as their strengths and weaknesses.

2.1 (Kyvik, 2009)

This work examines the necessity for effective quality assurance systems in higher education. It underscores the importance of understanding organizational dynamics to achieve continuous improvement and accountability. The book highlights challenges in maintaining quality standards, especially in non-university sectors, and advocates for robust quality assurance mechanisms to enhance educational outcomes and stakeholder satisfaction. Kyvik employs a comprehensive analysis of quality assurance processes, focusing on institutional structures, policy frameworks, and stakeholder engagements to derive strategies for organizational change and quality management. The insights offered in the book are valuable for designing tailored quality assurance systems, fostering continuous improvement, and enhancing institutional performance. However, its focus on Western European contexts may limit its applicability elsewhere, and it does not fully address emerging trends post-2008. Future research could compare global quality assurance systems, explore common challenges, and assess

the long-term impact of these interventions on higher education governance and management.

2.2 (Chalaris, 2011)

Monolis Chalaris and colleagues address the need for clear objectives and strategic planning in Higher Education Institutions (HEIs). They focus on the lack of a strategic management system to improve quality and performance. The study implemented a Management Information System (MIS) named QAIS TEI-A, which uses the Balanced Scorecard (BSC) methodology to define objectives across teaching, research, partnerships, processes, and resources. QAIS TEI-A effectively monitored and optimized educational and research services, supporting administrative decision-making and strategic planning. Despite its success, challenges included defining relevant indicators and ensuring reliable data collection. Future improvements could focus on refining data processes and enhancing interoperability between systems.

2.3 (Vernez, 2016)

This paper identifies the lack of a coordinated system for monitoring school performance in the Kurdistan Region-Iraq. It reviews existing practices and proposes a framework to align indicators with educational goals, set performance targets, and build capacity for quality assurance. The study reveals limited data scope, poor evaluation quality, and lack of transparency as

significant challenges. Future directions include developing a monitoring framework, improving alignment with educational objectives, and enhancing the Ministry of Education's capacity for managing performance.

2.4 (Haskova, 2017)

Alena Haskova discusses the critical role of information systems in quality assurance for higher education. The research focuses on the implementation of European Standards and Guidelines (ESG) and highlights achievements and limitations in these systems, such as interconnection complexity and bureaucratic challenges. Recommendations include simplifying data access, reducing duplication, and developing systems for analyzing graduate employability. The paper emphasizes improving communication, data compatibility, and the role of information systems in enhancing quality assurance processes.

2.5 (Ni Ketut Dewi Ari Jayanti and Ni Luh Ayu Kartika Yuniastari Sarja, 2019)

Ni Ketut Dewi Ari Jayanti and Ni Luh Ayu Kartika Yuniastari Sarja address the lack of IT integration in quality assurance systems. The study developed a model combining organizational culture, IT, and human resources to enhance quality assurance. It emphasizes the importance of aligning IT systems with institutional goals while acknowledging challenges in data availability and resource

constraints. Future research could test the model's applicability across institutions using Structural Equation Modeling, enhancing its relevance to diverse contexts.

2.6 (Hadzhikoleva, 2020)

Stanka Hadzhikoleva and colleagues propose a centralized software system to streamline quality assurance in higher education. Their model, based on the COMPASS application, supports self-assessment, expert evaluation, and evidence collection. It improves efficiency, transparency, and standardization in quality assurance processes. Challenges include scalability, data security, and integration with existing systems. Future efforts could refine the model using cloud computing for scalability and explore innovative technologies for quality assurance. This work highlights digitalization's role in improving collaboration, reducing errors, and enhancing overall quality standards in education.

2.7 (EFMD Global Network's Quality Assurance Academy)

The EFMD Global Network's Quality Assurance Academy is a global, non-profit, membership-driven organization dedicated to management development. It is recognized globally as an accreditation body for business schools, business school programmers, and corporate universities. The system focuses on enhancing the capabilities of quality assurance professionals through workshops, webinars, best practice sharing, and access to a network of experts.

Strengths

- Comprehensive Training Programmers

The EFMD Global Network's Quality Assurance Academy offers a wide array of comprehensive training programmers tailored to the needs of management professionals and academic institutions.

- Networking Opportunities with Industry Experts

Membership in the EFMD Global Network's Quality Assurance Academy provides unparalleled networking opportunities with industry leaders, accreditation agencies, peer institutions, and renowned experts in management education.

- Access to a Wide Range of Resources and Best Practices

Members of the academy benefit from a vast repository of resources encompassing research publications, case studies, benchmarking reports, toolkits, and best practice guides.

- Professional Development and Continuous Learning

The academy prioritizes professional development and lifelong learning by offering continuous learning opportunities, professional certification programmers, and skill-building workshops.

- Global Recognition and Credibility

Being affiliated with the EFMD Global Network's Quality Assurance Academy enhances institutional visibility, credibility, and recognition on a global scale.

- The academy's accreditation standards and quality assurance frameworks are internationally recognized and respected, providing member institutions with a competitive edge and validation of their quality assurance practices.

Weaknesses

- Document Version Control System

  Limited ability to manage versions of important documents effectively.

- Questionnaire templates

  No functionality for creating customized questionnaire templates for accreditation, assessments and audits.

- Scheduling activities

  Limited capability to schedule accreditation or audit activities within the system.

- Notification mechanism

  Lacks an automated notification mechanism to alert user of upcoming events or deadlines.

- Feedback Mechanism

  No robust method to collect or process user feedback on processes or outcomes.

2.8 (Quality Assurance Agency-Mauritius)

The Quality Assurance Authority (QAA) is a statutory body established under the Higher Education Act 2017 (Act No. 23 of 2017 under Section 28). The QAA became operational on 17 January

2020. The objectives of the QAA are to promote, maintain and enhance quality assurance of higher education in line with international high-quality standards in higher education through appropriate quality assurance mechanisms.

Amongst its functions, the QAA shall ensure that standards for qualifications in every higher educational institutions were met and shall carry out regular quality audits of higher education institutions. The QAA is also required to monitor the delivery of online and inter-institutional programmers.

Strengths

- Robust accreditation management system with comprehensive compliance checks.
- Automated quality assessment tools for evaluating educational programs and institutions.
- Integration with national education databases for data synchronization and reporting.
- User-friendly interface with customizable dashboards for stakeholders.

Weaknesses

- Document Version Control System

  Difficulty in managing revisions and tracking changes in documents.
- Notification mechanism

No alert system to notify users about scheduled activities or deadlines.

- Questionnaire template

Does not support template creation for tailored accreditation or assessment purposes.

- Forms submissions management

Challenges in handling and managing form submissions efficiently.

2.9 (TouchNetix-Quality Assurance Accreditation)

TouchNetix-Quality Assurance Accreditation is a system designed to manage quality assurance and accreditation processes for educational institutions. It includes features for document management, compliance tracking, and assessment tools.

TouchNetix's Quality Management System (QMS) for its headquarters, manufacturing locations and regional design centers are certified to ISO9001:2015 by BSI. The axiom IC’s of the system are at the heart of many automotive embedded control systems. TouchNetix offers an increasingly diverse portfolio of automotive touch ICs, qualified to the AEC-Q100 standard and manufactured in accordance with the AIAG Production Part Approval Process.

Strengths

- Comprehensive document management system with version control and audit trails.
- Automated compliance checks and reporting functionalities.

- Integration with learning management systems for data exchange and analysis.
- Access Privileges
- Document Management
- Submissions Management

Weaknesses

- Document Version Control System

  Difficulty in managing revisions and tracking changes in documents.

- Notification mechanism

  No alert system to notify users about scheduled activities or deadlines.

- Questionnaire template

  Does not support template creation for tailored accreditation or assessment purposes.

- Forms submissions management

  Challenges in handling and managing form submissions efficiently.

- Scheduling activities

  No functionality to schedule accreditation or assessment tasks directly within the system.

2.10 Valley View University Quality Assurance System

Valley View University Quality Assurance System is a proposed system aimed at enhancing quality assurance processes and document

management within Valley View University. It focuses on automating workflows, improving communication, and ensuring compliance with accreditation standards.

Strengths:

- Document Management
- Access privileges
- Forms submissions management
- Database management system

Weaknesses:

- Questionnaire templates

  The current system does not provide a way to create questionnaire templates for assessments, accreditation and audits.

- Scheduling activities

  The system does not have the capability to schedule activities digitally.

- Notification mechanism

  The system lacks a notification mechanism to alert users about important events or deadlines.

CHAPTER THREE

CRYSTALIZATION OF THE PROBLEM

3.0 Background

Accreditation and quality assurance in higher education have long been critical for maintaining academic standards and ensuring continuous improvement. Traditionally, universities relied on manual processes involving paperwork, physical reviews, and face-to-face meetings. Over time, as the complexity of academic programs increased and the demand for higher educational standards grew, there was a need for a more structured and efficient approach to manage these processes. Many universities, including the one in focus, still rely on semi-manual systems for quality assurance. These systems involve a combination of physical documents, spreadsheets, and basic digital tools that lack integration and automation. This leads to inefficiencies, delays, and potential errors in the accreditation, auditing and assessment processes.

3.1 How it operates

At Valley View University, the Quality Assurance (QA) processes encompass a variety of activities, including accreditation, assessments, and audits, internal and external reviews. Currently, these processes are managed using a combination of physical documents, spreadsheets, and basic digital tools. Here is an overview of how these processes typically operate:

Departments and units fill out physical forms or spreadsheets for various QA processes (such as assessments, audits, and accreditation) and submit them to the QA office.

The QA office manually reviews the submitted forms, ensuring completeness and compliance with the relevant standards, whether they are for accreditation, internal assessments, or auditing.

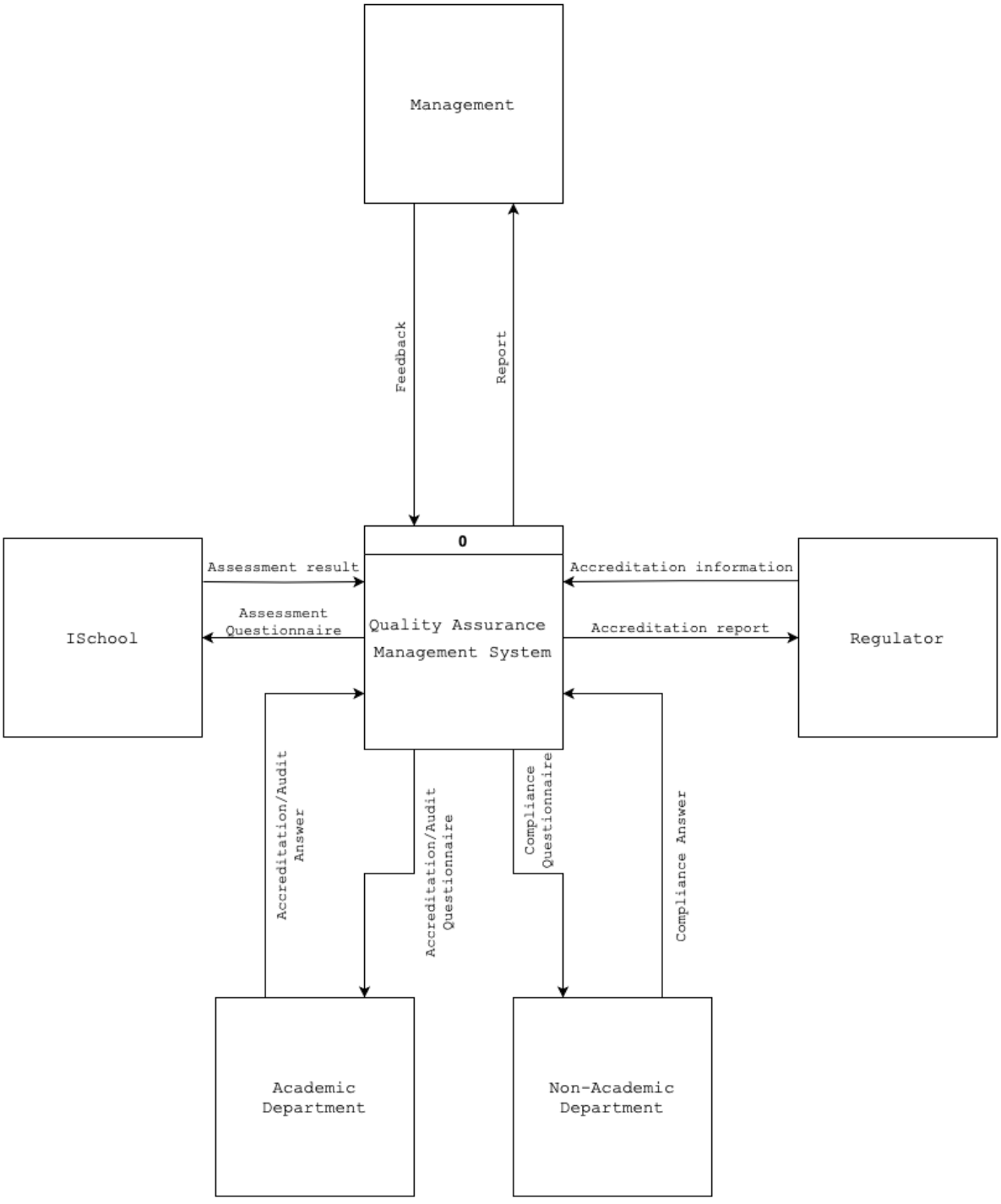
The QA office tracks the progress of each form and process manually, including updating statuses and ensuring all steps are completed on time.

The QA office compiles reports from various sources, aggregating data from different departments and processes, often requiring manual data collection and review.

Communication & Coordination: Feedback is communicated to departments through emails, meetings, or physical documents, and coordination with accreditation bodies or internal reviewers is done manually, often involving face-to-face interactions or separate email chains.

This manual and semi-digital approach creates inefficiencies, delays, and potential for errors. There is a need for a streamlined and integrated system that can automate key aspects of these processes and improve efficiency and communication across the QA department and the university at large.

3.2 Context Diagram for the Existing System



3.3 Strength/Advantages

Document Management

Valley View University's current Quality Assurance (QA) system ensures that critical documents, such as accreditation reports, assessment results, and audit records, are systematically organized. This makes them easily accessible for reviews or audits. The well-structured documentation process minimizes the risk of misplacing important files and supports efficient retrieval, which is essential for maintaining institutional quality.

Access Privileges

The QA system enforces strict access controls, ensuring that only authorized personnel can access sensitive documents. This feature protects the integrity and confidentiality of QA data, reducing the likelihood of unauthorized changes of breaches. Such as controls contribute to safeguarding institutional information and fostering trust in the QA processed.

Forms Submissions Management

The system facilitates the submission and management of QA-related forms. Departments and individuals can submit documents for assessments, audits, or accreditation using established procedures. This standardized approach promotes uniformity and

consistency across various departments, ensuring that all submissions meet the QA requirements.

Database Management System

The QA department leverages basic database management tools to track and manage essential data. Tools like spreadsheets are used to monitor deadlines, departmental compliance records, and audit feedback. This centralized repository supports timely decision-making and enables the QA team to maintain an organized view of critical data.

3.4 Weaknesses/Disadvantages

Questionnaire Templates

A significant limitation of the current QA system is its lack of a mechanism for creating and managing standardized questionnaire templates. Staff are forced to design questionnaires from scratch for assessments, accreditation, or audits, resulting in inconsistent formats and repetitive efforts. This shortcoming increases the likelihood of errors and inefficiencies during data collection.

Scheduling Activities

The absence of a digital scheduling feature hinders the coordination of QA tasks. Currently, QA officers rely on manual methods such as physical calendars or email communication, which

often result in scheduling conflicts, miscommunication, and missed deadlines. This limitation negatively affects the efficiency of planning and executing QA activities.

Notification Mechanism

The system does not provide notifications to alert stakeholders about important events, such as upcoming deadlines or scheduled audits. Without such alerts, users may miss critical actions, leading to delays in meeting compliance requirements or completing QA tasks on time.

Limited Collaboration and Silos

Processes like document reviews and feedback sharing are conducted manually, with limited collaboration between departments. This siloed approach causes delays and inefficiencies in QA operations, as communication gaps often slow down decision-making and feedback cycles.

Absence of Analytical Tools

The current QA system lacks analytical tools to track performance metrics and compliance. QA officers must rely on manual analyses, which are time-consuming and prone to errors. The absence of automated analytics limits the ability to gain insights into the effectiveness of QA activities and make data-driven improvements.

3.5 Feature-Based Analysis

Name of System	**Document version Control system**	**Questionnaire templates**	**Scheduling activities**	**Notification mechanism**	**Feedback Mechanism**	**Access control**
The Dynamics of Change in Higher Education	YES	NO	YES	YES	NO	YES
Developing an Information System for Quality Assurance in Higher Education	YES	NO	YES	NO	NO	YES
Developing a School Quality Assurance System	YES	NO	YES	NO	NO	YES
Information Systems as a Component Part of Tertiary Education Quality Assurance	YES	NO	YES	NO	NO	YES
Quality Assurance System in Higher Education	YES	NO	YES	YES	NO	YES
Model of a Centralized System for Quality Assurance in Higher Education	YES	NO	YES	NO	YES	YES
EFMD Global Network's Quality Assurance	YES	NO	YES	YES	NO	YES

Quality Assurance Agency-Mauritius	YES	NO	YES	NO	YES	YES
TouchNetix-Quality Assurance Accreditation	YES	NO	YES	NO	NO	YES
Valley View University Quality Assurance system	YES	NO	YES	NO	NO	YES
Proposed System	YES	YES	YES	YES	YES	YES

CHAPTER FOUR

ANALYSIS OF THE PROPOSED SYSTEM

4.0 Overview of the proposed system

The Valley View University Quality Assurance Management System (UniQAMS) is a mobile-based platform designed to improve the quality assurance processes at Valley View University. The system aims to address the current challenges faced by the QA department, such as manual workflows, lack of digitized templates, inefficient communication, and challenges in maintaining accreditation standards.

The system focuses on automating repetitive tasks, enhancing document and workflow management, and ensuring compliance with accreditation requirements. It will feature modules for questionnaire templates, activity scheduling, document management, notifications, and database management.

-Functional Requirements

The functional requirements of the proposed system include:

- User Authentication and Authorization; allow users to log in with different roles (admin, department heads, and staff).
- Questionnaire Template Management; enable users to create, edit, and manage customizable templates for assessments, accreditations, and audits.

- Activity Scheduling; provide a calendar-based interface to schedule activities such as audits, assessments, and accreditation reviews.
- Notification; alert users about upcoming deadlines, scheduled activities, and accreditation updates.

-Non-Functional Requirements

- Scalability; the system should handle an increasing number of users and data as the university grows.
- Usability; provide an intuitive and user-friendly interface.
- Performance; deliver a responsive and fast user experience, even under heavy usage.
- Reliability; guarantee high availability with minimal downtime.

4.1 Major Features/Component of the Proposed System

- Dashboard

  A centralized hub displaying key metrics, notifications, and an overview of QA activities across departments.
- Questionnaire Templates Management

  Tools for creating reusable templates for accreditation, assessments, and auditing processes, customized to the university's needs.
- Activity

A digital calendar for scheduling and tracking QA-related activities.

- Document Management System

  Secure storage for QA documents, with version control and easy sharing capabilities.

- Notifications

  Automated alerts for upcoming deadlines, accreditation milestones, or incomplete tasks.

4.2 Benefit/Advantages of the Proposed System

- Enhanced Efficiency

  Automating repetitive tasks, such as document tracking and activity scheduling, reduces workload and improves productivity.

- Improved With real-time notifications and centralized information, stakeholders stay informed and collaborate effectively.

- Compliance with Accreditation Standards

  The system provides tools and templates that ensure alignment with accreditation requirements, reducing the risk of non-compliance.

- Data Integrity and Security

  Role-based access control and secure data storage ensure that sensitive information remains protected.

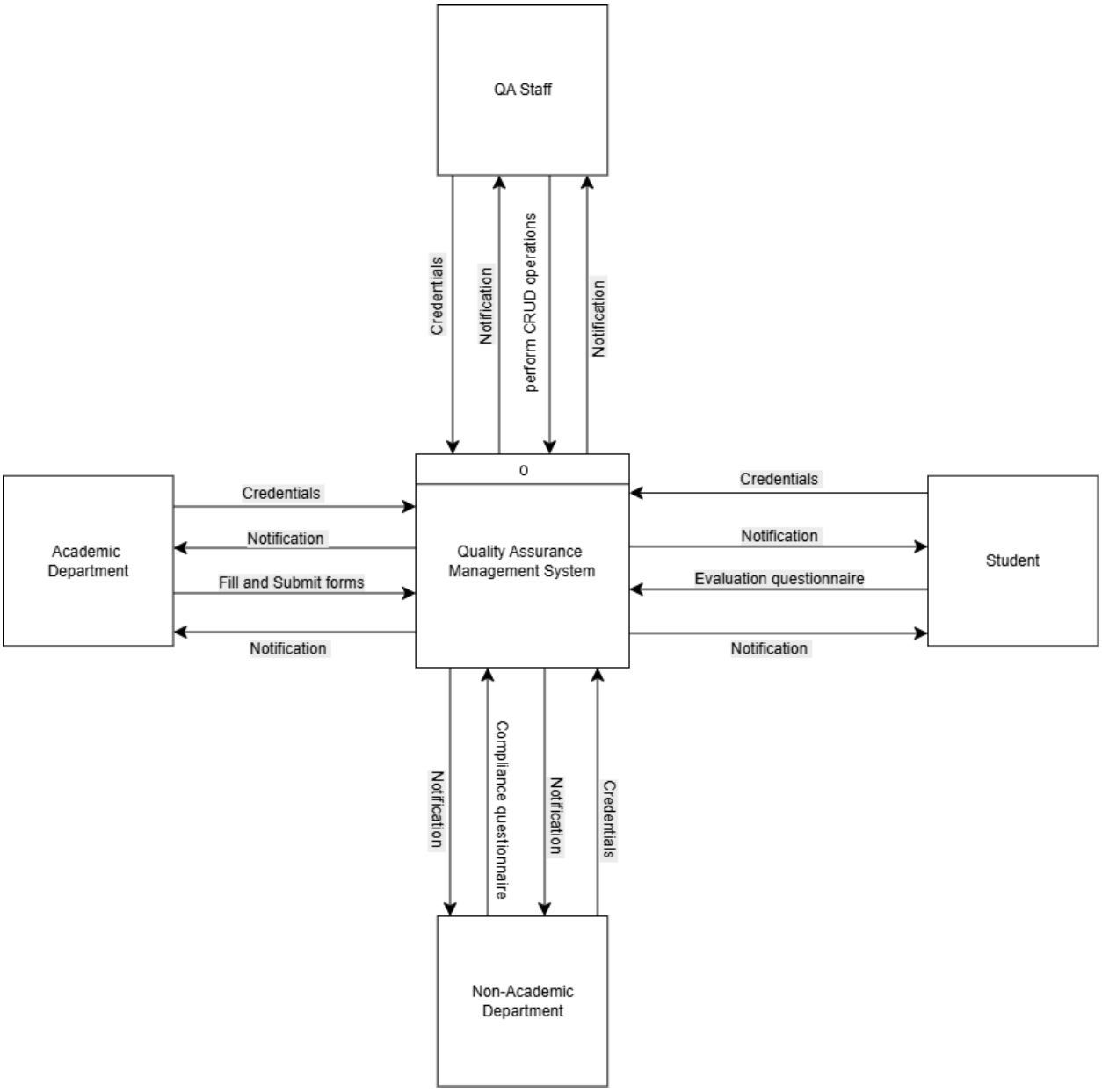
- Scalability for Future Growth

  The system is designed to accommodate the growing needs of the university, ensuring long-term usability.

- Streamlined Feedback Process

  Post-assessment reporting tools enable swift analysis of strengths and weaknesses, facilitating continuous improvement.

4.3 System Context Diagram of the Proposed System



CHAPTER FIVE

DETAIL DESIGN OF THE PROPOSED SYSTEM

5.0 Functional Processes of the Proposed System

User Authentication and Authorization

This process ensures secure access to the system by implementing Login functionality with ID and password. Role-based access control grants permissions specific to user roles, such as admins, head of departments, or QA staff, enabling users to access only relevant data.

Questionnaire Template Management

Designed to standardize and simplify the creation of templates for quality assurance activities, this process allows users to build reusable templates for accreditation, assessments, and audits using drag-and-drop functionality for questions, dropdowns, and checkboxes.

Activity Scheduling

This feature organizes QA activities by providing a calendar interface for scheduling audits, assessments, and reaccreditation events. Activities can be assigned to specific departments or individuals, with detailed descriptions and objectives added for clarity.

Notification and Alerts

To promote timely communication of critical QA tasks and deadlines, this process automates alerts for pending tasks like submitting assessment results or uploading audit documents. It sends reminders for upcoming events such as accreditation deadlines or meetings, alongside real-time notifications for urgent updates like failed audits.

5.1 Algorithm and Flowchart of the Processes

5.1.1 Algorithm of the Processes

Step 1: Start

Step 2: Display the Home Page

Step 3: Navigate to the Login Page

Step 4: Prompt the User to Enter ID and Password

Step 5: Check Credential Validity

Step 6: If credentials are invalid, display an error message and return to Step 4

Step 7: Display the Dashboard

Step 8: Determine if the User is an Admin

Step 9: If the user is not an admin, proceed to Step 13

Step 10: Perform CRUD Operations on Users and Roles

Step 11: Assign Permissions to Users and Roles

Step 12: Proceed to Step 15

Step 13: Determine if the User is QA Staff

Step 14: If the user is not QA Staff, proceed to Step 17

Step 15: Create Questionnaire Templates

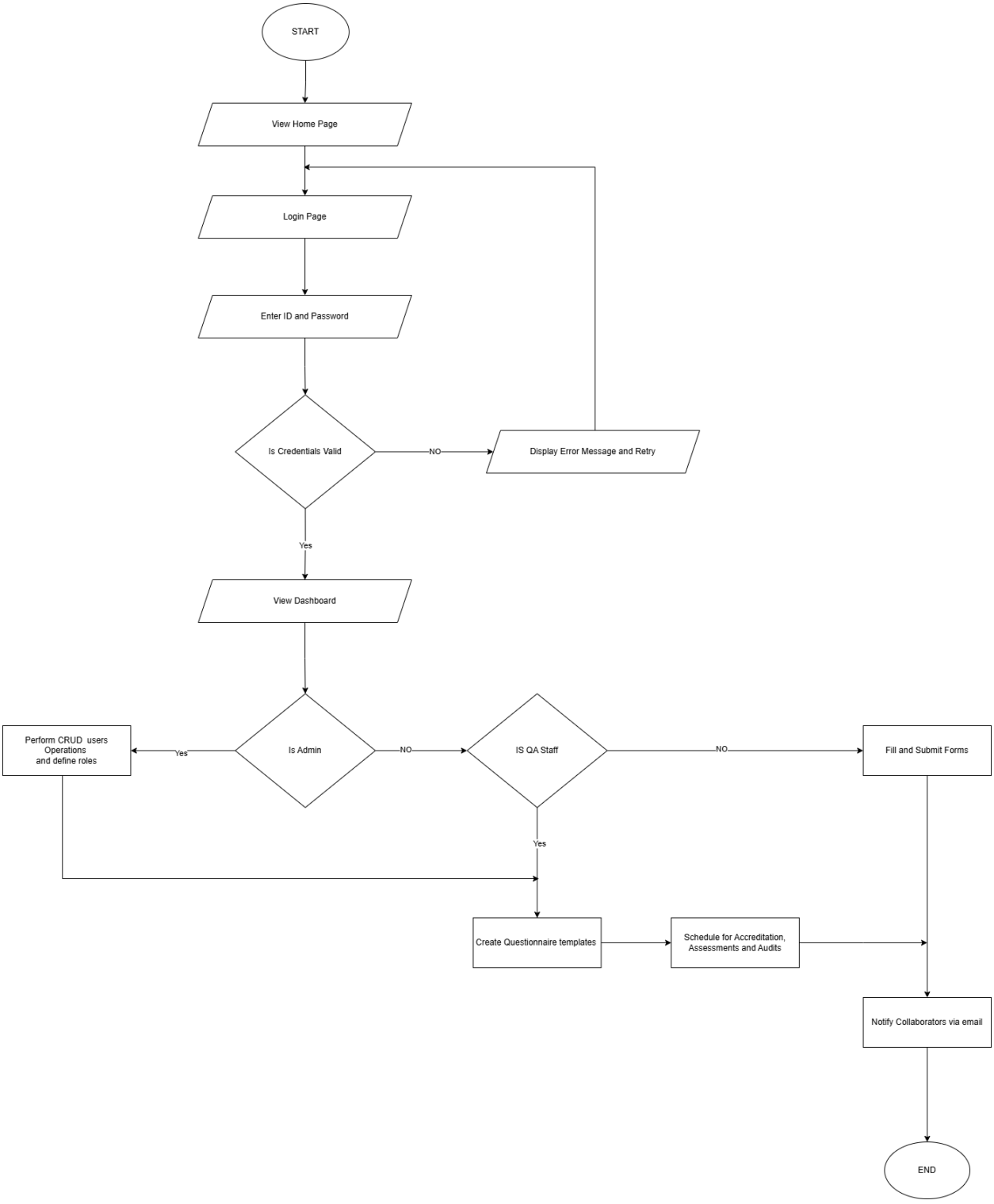
Step 16: Schedule QA activities (Accreditations, Audits, and Assessments)

Step 17: Participate in Assigned Forms

Step 18: Notify Collaborators via Email

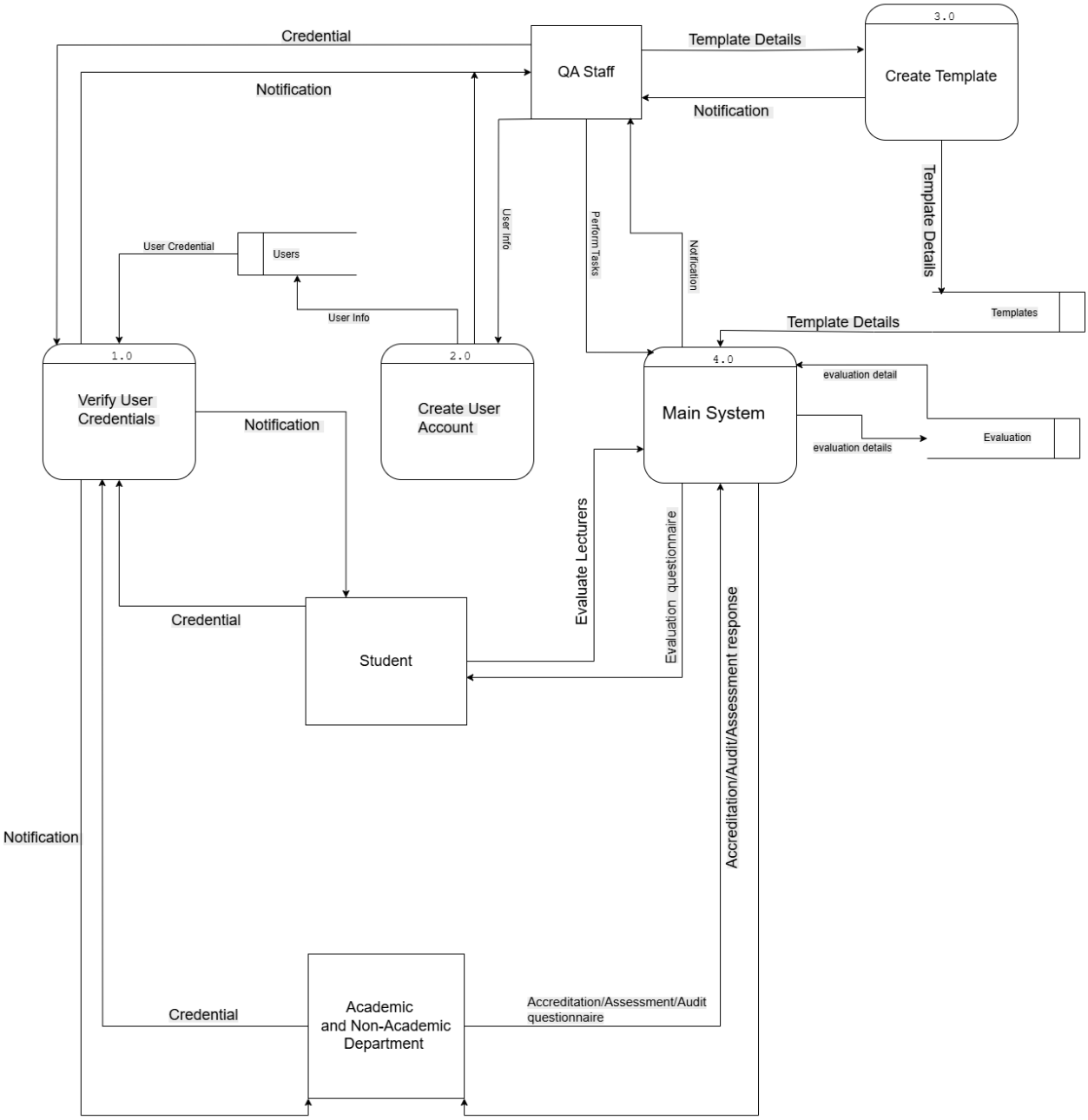
Step 19: End

5.1.2 Flowchart of the Processes



5.2 Data Flow Diagram

5.2.1 DFD 1 Diagram



5.3 Data Dictionary

5.3.1 Database Schema and the Tables

Table: Users

Field Name	Data Type	Size	Description	Constraints
id	VARCHAR	50	Unique Identifier for each user	Primary Key, Auto Increment
name	VARCHAR	100	Full name of the user	Not Null
email	TEXT	50	User email address	Unique, Not Null
imageUrl	JSON	N/A	URL for the user's image	Null
password	TEXT	N/A	Hashed password	Not Null
roleId	INT	4	Identifier for the user's role	Foreign Key → role.id, Not Null
campusId	INT	N/A	Associated campus identifier	Foreign Key → campus.id, Not Null
schoolId	INT	N/A	Associated school identifier	Foreign Key → schools.id, Not Null
departmentId	INT	N/A	Associated department identifier	Foreign Key → departments.id, Not Null
offBoarded	BOOLEAN	N/A	Indicates if the user is active	True/False
createdAt	TIMESTAMP	N/A	Timestamp of user creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Campus

Field Name	Data Type	Size	Description	Constraints

id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
name	VARCHAR	100	Name of the campus	Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Schools

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
name	VARCHAR	100	Name of the school	Not Null
campusId	INT	N/A	Identifier for associated campus	Foreign Key → campus.id, Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Departments

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
name	VARCHAR	100	Name of the department	Not Null
campusId	INT	N/A	Identifier for associated campus	Foreign Key → campus.id, Not Null
schoolId	INT	N/A	Identifier for associated school	Foreign Key → school.id, Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null

updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Programmes

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
name	VARCHAR	100	Name of the program	Not Null
campusId	INT	N/A	Identifier for associated campus	Foreign Key → campus.id, Not Null
schoolId	INT	N/A	Identifier for associated school	Foreign Key → schools.id, Not Null
departmentId	INT	N/A	Identifier for associated department	Foreign Key → departments.id , Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Roles

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
name	VARCHAR	100	Role name	Not Null
description	TEXT	N/A	Description of the role	Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Role_base_Permissions

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
roleId	INT	N/A	Role identifier	Foreign Key → roles.id, Not Null
actions	JSON	N/A	JSON structure defining permissions	Nullable
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: User_base_Permissions

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
userId	VARCHAR	50	User identifier	Foreign Key → users.id, Not Null
actions	JSON	N/A	JSON structure defining user-specific perms	Nullable
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Questionnaire_Templates

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
name	TEXT	N/A	Name of the questionnaire template	Not Null

description	TEXT	N/A	Description of the questionnaire template	Not Null
type	ENUM	N/A	Type (audit, accreditation, assessment, etc.)	Not Null
createdById	VARCHAR	50	Identifier for the template's creator	Foreign Key → users.id, Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Questions

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
templateId	INT	N/A	Identifier for the related template	Foreign Key → questionnaire_templates.id, Not Null
questionText	TEXT	N/A	Text content of the question	Not Null
inputType	ENUM	N/A	Input type (e.g., text, radio, checkbox)	Not Null
inputOptions	JSONB	N/A	Options for input, if applicable	Nullable
required	BOOLEAN	N/A	Indicates if the question is mandatory	Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null

updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Audit_Sessions

Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique identifier for the audit session	Primary Key, Auto Increment
title	VARCHAR	100	Title of the audit session	Not Null
description	TEXT	N/A	Detailed description of the audit session	Nullable
date	DATE	N/A	Date scheduled for the audit session	Not Null
startTime	TIME	N/A	Start time of the audit session	Not Null
endTime	TIME	N/A	End time of the audit session	Not Null
templateId	INT	N/A	Identifier for the associated questionnaire template	Foreign Key → questionnaire_templates.id, Not Null
auditType	ENUM	N/A	Type of audit (e.g. “Academic”, “Non-academic”)	Not Null
campusId	INT	N/A	Identifier for associated campus	Foreign Key → campus.id (optional)

schoolId	INT	N/A	Identifier for associated school	Foreign Key → schools.id (optional)
departmentId	INT	N/A	Identifier for associated department	Foreign Key → departments.id (optional)
createdById	VARCHAR	50	Identifier for the user who created the audit session	Foreign Key → users.id, Not Null
pendingReviewById	VARCHAR	50	Identifier for the user responsible for initial review	Foreign Key → users.id (optional)
reviewedById	VARCHAR	50	Identifier for the user who reviewed the session	Foreign Key → users.id (optional)
approvedById	VARCHAR	50	Identifier for the user who approved the session	Foreign Key → users.id (optional)
completedById	VARCHAR	50	Identifier for the user who marked the session as completed	Foreign Key → users.id (optional)
rejectedById	VARCHAR	50	Identifier for the user who rejected the session	Foreign Key → users.id (optional)
cancelledById	VARCHAR	50	Identifier for the user who cancelled the session	Foreign Key → users.id (optional)
rejectedReason	TEXT	N/A	Reason provided when the session is rejected	Nullable

cancelledReason	TEXT	N/A	Reason provided when the session is cancelled	Nullable
pendingReviewAt	TIMESTAMP	N/A	Timestamp when the session was marked as pending review	Nullable
reviewedAt	TIMESTAMP	N/A	Timestamp when the session was reviewed	Nullable
approvedAt	TIMESTAMP	N/A	Timestamp when the session was approved	Nullable
cancelledAt	TIMESTAMP	N/A	Timestamp when the session was cancelled	Nullable
rejectedAt	TIMESTAMP	N/A	Timestamp when the session was rejected	Nullable
status	ENUM	N/A	Current status (e.g., pending review, approved, rejected)	Default: “pending”, Not Null
collaboratorsIds	JSON	N/A	JSON array of collaborator identifiers	Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

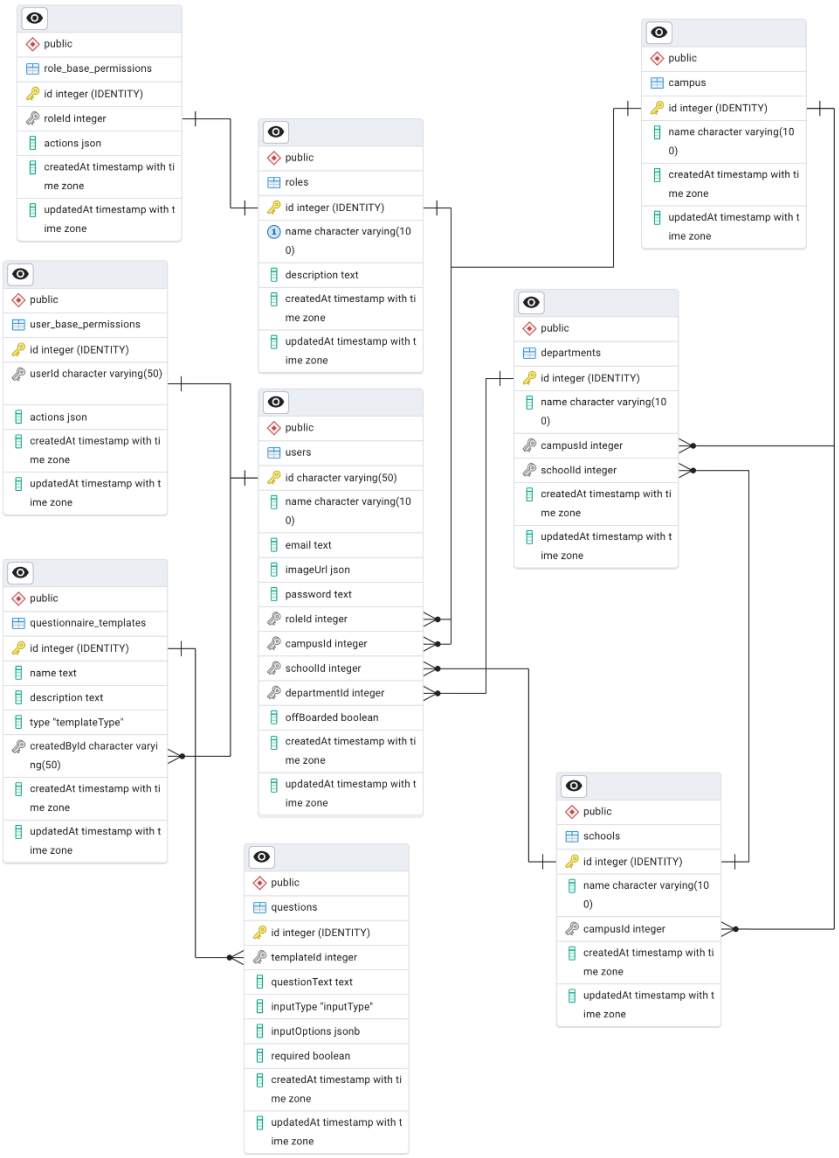
Table: Audit_Session_Questions

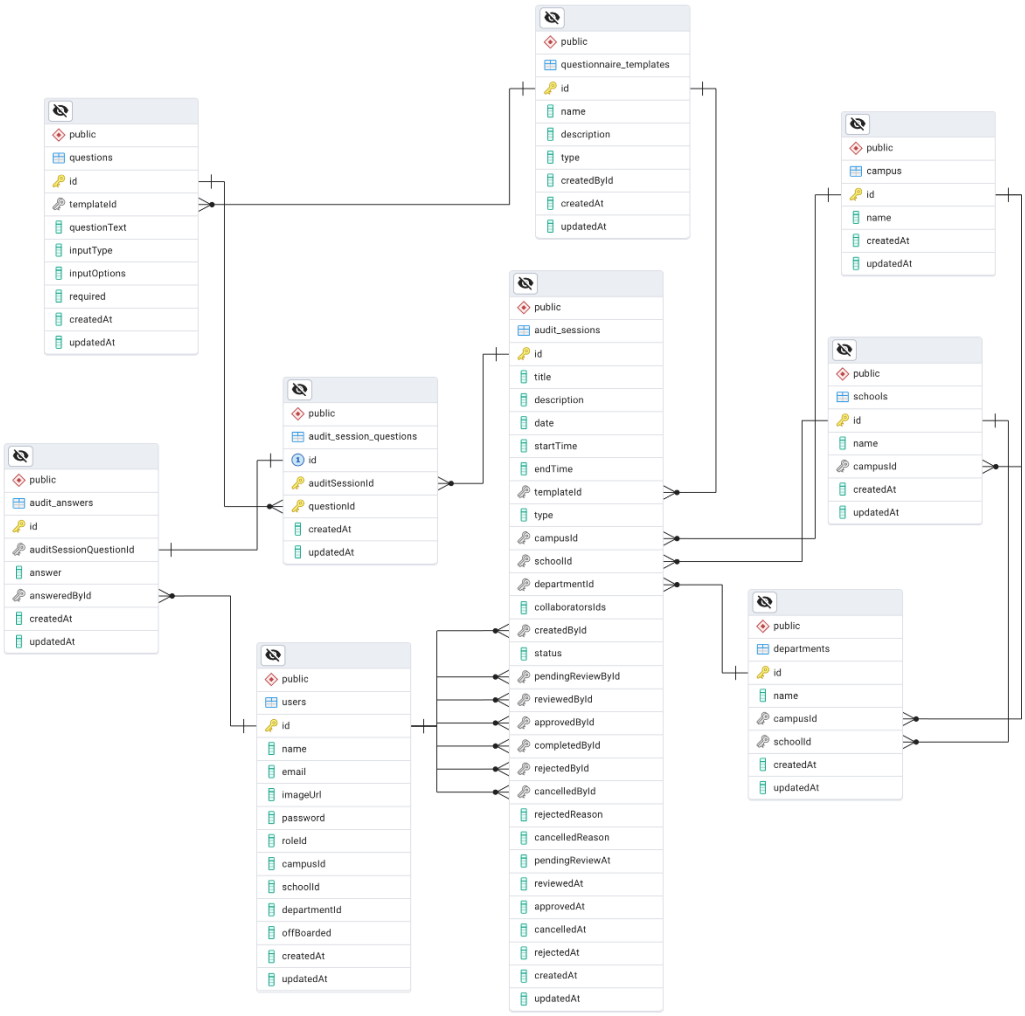
Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique Identifier	Primary Key, Auto Increment
auditSessionId	INT	N/A	Identifier for the audit session	Not Null
questionId	INT	N/A	Identifier for the question	Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

Table: Audit_Answers

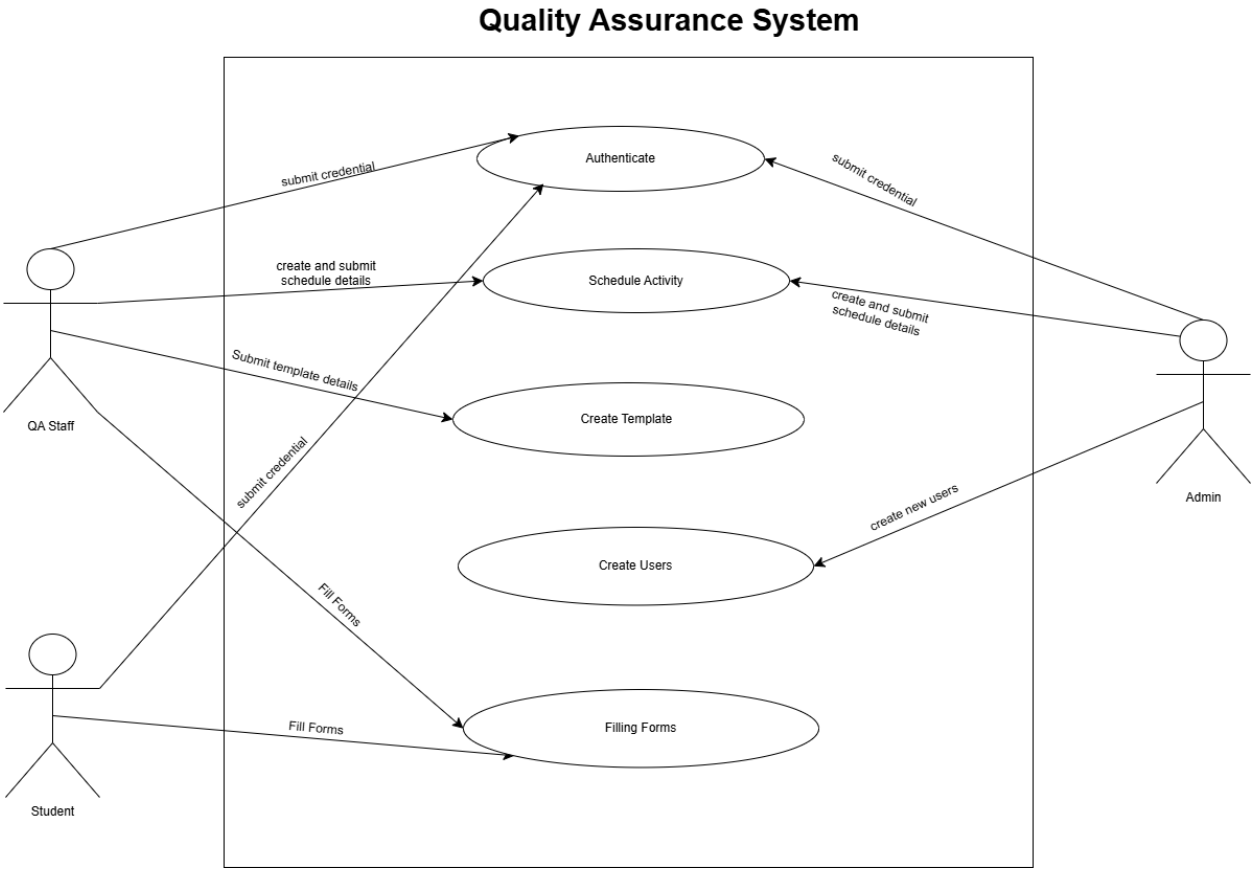
Field Name	Data Type	Size	Description	Constraints
id	INT	N/A	Unique identifier for the audit answer	Primary Key, Auto Increment
auditSessionQuestionId	INT	N/A	Identifier linking to the audit session question	Foreign Key → audit_session_questions.id, Not Null
answer	TEXT	N/A	JSON structure containing the answer	Not Null
answeredById	VARCHAR	50	Identifier for the user who answered	Foreign Key → users.id, Not Null
createdAt	TIMESTAMP	N/A	Timestamp of creation	Not Null
updatedAt	TIMESTAMP	N/A	Timestamp of last update	Default: Current Timestamp

5.3.2 Entity relationship diagrams

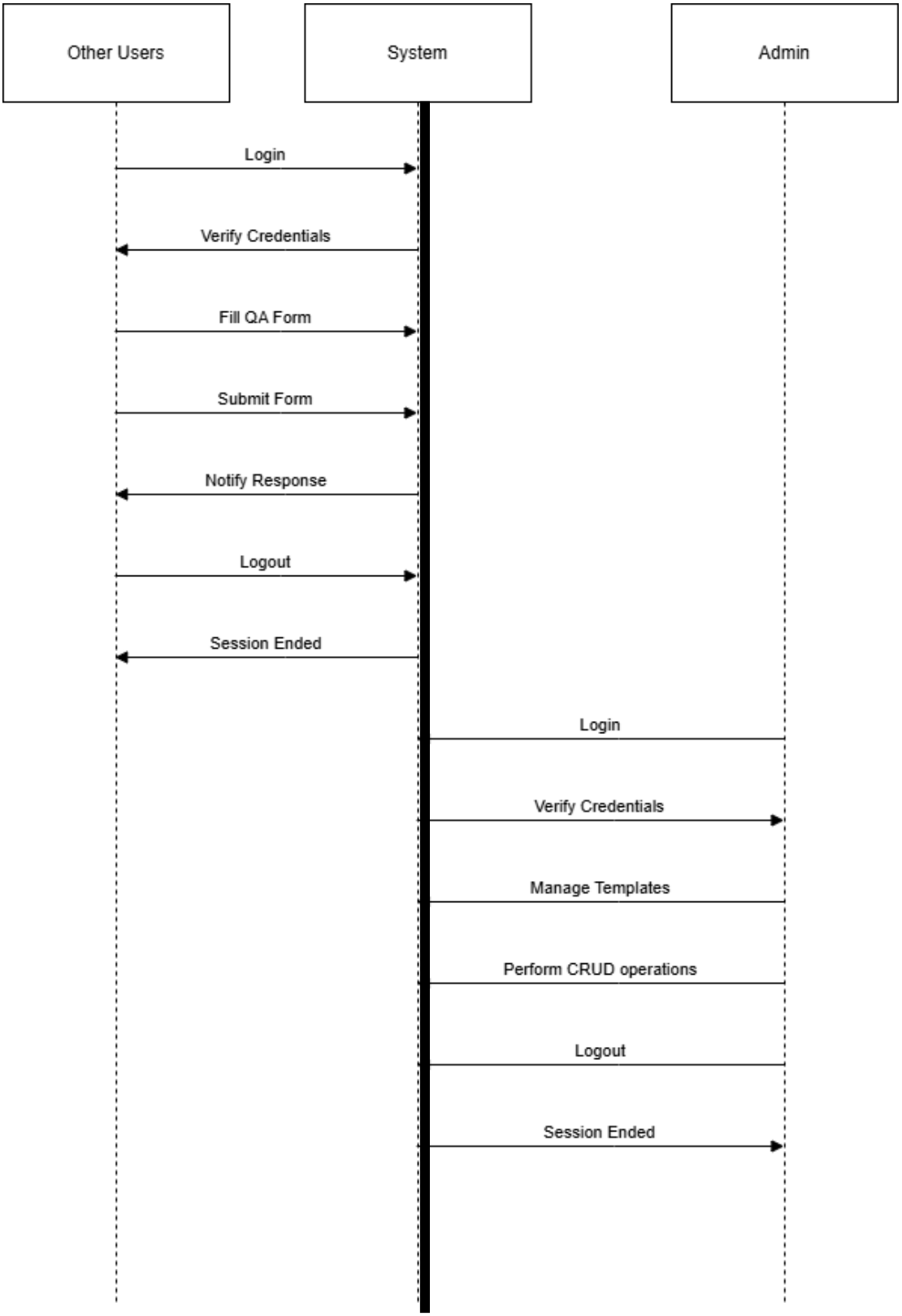



public
questionnaire_templates
id
name
description
type
createdById
createdAt
updatedAt
public
questions
id
templateId
questionText
inputType
inputOptions
required
createdAt
updatedAt
public
campus
id
name
createdAt
updatedAt
public
audit_sessions
id
title
description
date
startTime
endTime
templateId
type
campusId
schoolId
departmentId
collaboratorsIds
createdById
status
pendingReviewById
reviewedById
approvedById
completedById
rejectedById
cancelledById
rejectedReason
cancelledReason
pendingReviewAt
reviewedAt
approvedAt
cancelledAt
rejectedAt
createdAt
updatedAt
public
schools
id
name
campusId
createdAt
updatedAt
public
audit_session_questions
id
auditSessionId
questionId
createdAt
updatedAt
public
audit_answers
id
auditSessionQuestionId
answer
answeredById
createdAt
updatedAt
public
departments
id
name
campusId
schoolId
createdAt
updatedAt
public
users
id
name
email
imageUrl
password
roleId
campusId
schoolId
departmentId
offBoarded
createdAt
updatedAt

5.3.3 Use Case or User Scenarios UML



5.3.4 Sequence Diagram



CHAPTER SIX

SYSTEM IMPLEMENTATION & TESTING

6.0 Implementation

System implementation represents the critical transition phase where theoretical designs become operational solutions. This stage encompasses hardware provisioning, software installation, system configuration, and deployment - transforming specifications into a functional mobile application for quality assurance management. System design is concerned mainly with the coordination of activities, job procedures and equipment utilization in order to achieve organizational objectives. It addresses data input and output data, processing and interface.

This stage involves the design of the new duty management System.

6.1 Programming Language

Choosing a programming language depends on one’s language experience and the kind and scope of the application to be built. While small applications are often created using only one language, it is not uncommon to develop large applications using multiple languages.

The proposed application to be built is an application that needs internet facilities to function.

The choice of programming languages and frameworks was based on the mobile nature of the application and the need for cross-platform compatibility. The primary technologies selected were

React Native for frontend development and PostgreSQL for database management, with Expo serving as the development toolkit.

6.2 The Choice of React Native and PostgreSQL

In the implementation process, React Native and PostgreSQL were preferred for this project because they represent the most suitable technologies for developing performant cross-platform mobile applications with robust data management capabilities. This choice was informed by the specific requirements of a university quality assurance system needing both mobile accessibility and reliable data storage.

6.2.1 React Native

React Native is an open-source, powerful JavaScript framework used to create native-like mobile applications for both Android and iOS platforms. Developed by Facebook, it allows developers to build mobile apps using React along with native platform capabilities. React Native components render to native platform UI elements, resulting in applications that look and feel truly native while sharing most of the codebase across platforms. The framework supports hot-reloading, which significantly speeds up development by instantly reflecting code changes in the running application.

6.2.2 PostgreSQL

PostgreSQL is an advanced, enterprise-class open-source relational database system that extends the SQL language with powerful

features. Known for its reliability, feature robustness, and performance, PostgreSQL provides sophisticated data management capabilities ideal for academic quality assurance systems. It offers comprehensive support for JSON data types, advanced indexing techniques, and robust security features - making it perfectly suited for handling the complex data relationships in university quality management.

6.2.3 Advantages

The popularity of this technology stack results from several key advantages:

- Cross-Platform Efficiency: React Native enables simultaneous development for both Android and iOS platforms, significantly reducing development time and costs while maintaining native performance characteristics.
- Strong Data Integrity: PostgreSQL provides ACID-compliant transactions and sophisticated data validation mechanisms, ensuring the reliability of critical quality assurance data.
- Rapid Development Cycle: The Expo toolkit combined with React Native’s hot reloading feature allows for immediate viewing of changes during development, dramatically speeding up the iteration process.
- Active Community Support: Both React Native and PostgreSQL benefit from large, active developer communities that continuously

contribute improvements, security updates, and third-party libraries.

• Cost-Effectiveness: The entire technology stack is open-source, eliminating licensing costs while providing enterprise-grade capabilities suitable for academic institutions.

These are the intangible components of the tools used to implement the system, and they included:

- Sublime: Sublime is an asynchronous event-driven JavaScript runtime, sublime is designed to build scalable network applications

- Notepad++: Notepad++ is a code editor redefined and optimized for building and debugging modern web-based and cloud applications.

- VS Code: VS codes comes with built-in support for JavaScript, Typescript, and Node.js. It has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, HTML Go) and runtimes (such as .NET and Unity).

6.3 Hardware Requirement

The system would require the following hardware devices for its installation.

1. High-performance laptop

2. Processor: Intel Core i7(11th Gen)
3. RAM: 16GB DDR4
4. Storage: 1TB SSD
5. Keyboard
6. Mouse
7. Server

Purpose:

Running React Native (Expo) development environment

Hosting local PostgreSQL database for initial testing

Executing Android and iOS phone

6.4 Software Requirement

1. Windows operating system such as Windows 11
2. React Native Development

-Node.js

-Expo CLI

-Physical phone

3. Database Setup

PostgreSQL installed locally for testing

pgAdmin for database management

4. Testing Process: the app was tested using Expo Go (QR code scanning) on real devices.

6.5 Testing

Testing ensures the quality of the software. This phase is very crucial because quality, verification, and validation, and reliability are to be ensured. The following tests were done on the application to ensure it works as expected and meets the requirements that guided its design.

The following Test cases were considered:

- To sign in with the portal provided.
- To create, edit, and remove questionnaires with the portal provided.
- To create user profile and to be able to add other users in the system.

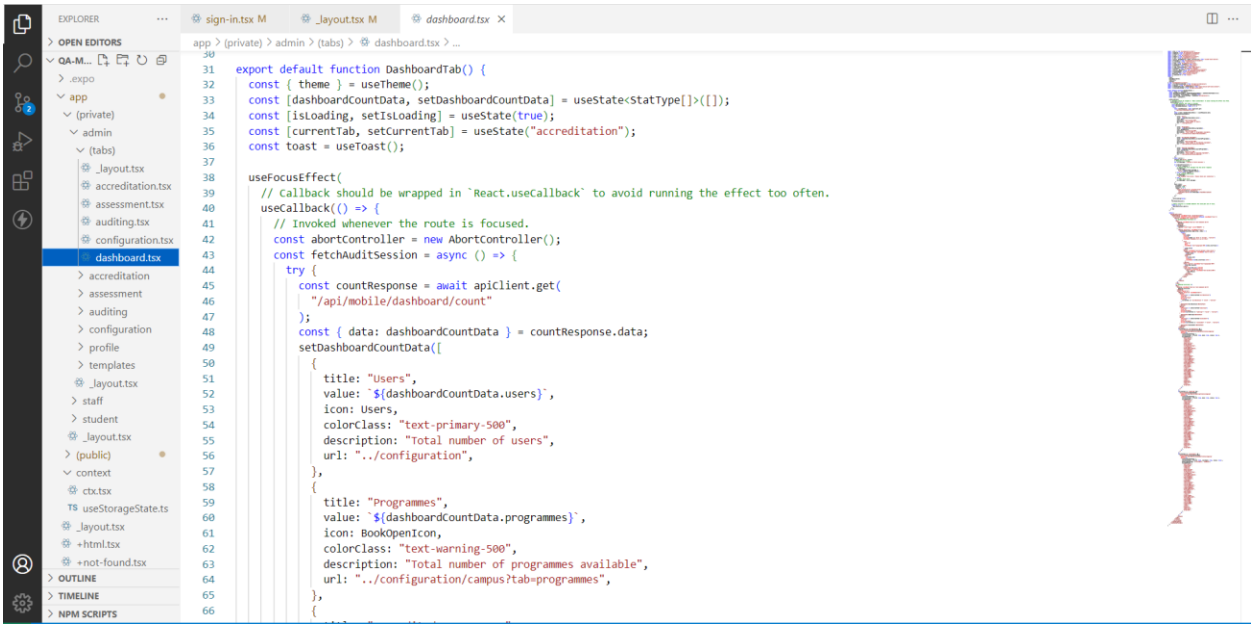
6.6 Test Cases

Table 6.6.1 Test cases implemented

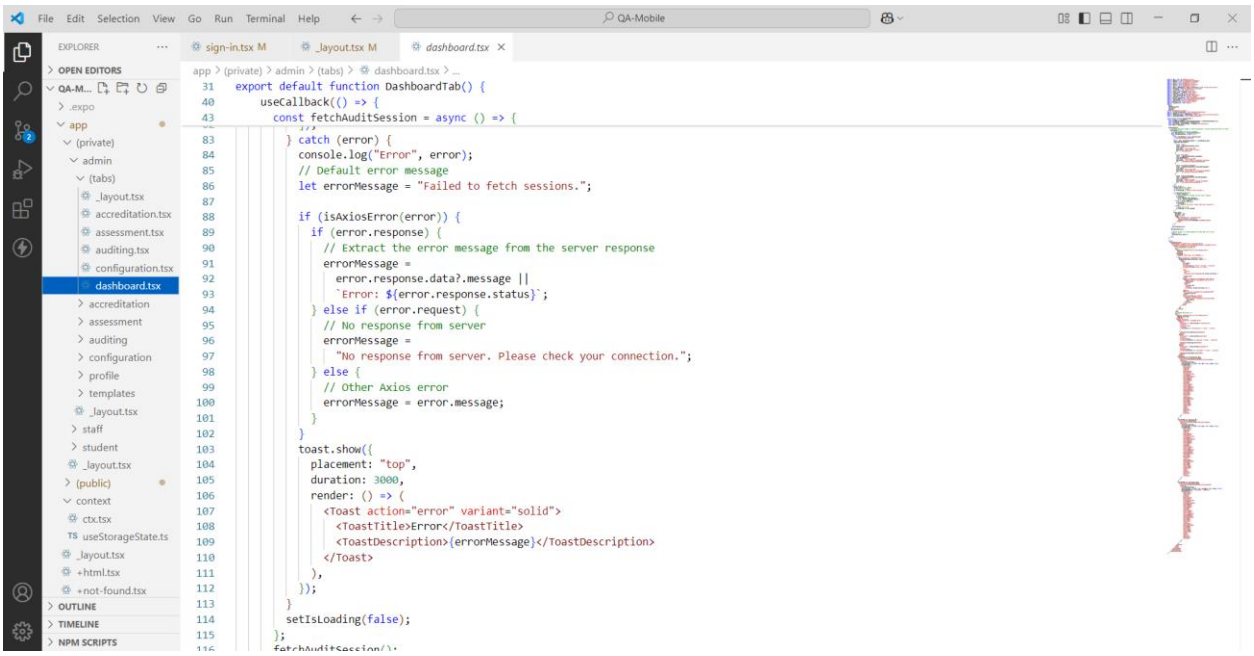
	TEST DISCRIPTION	**EXPECTED RESULT**	**TYPE OF TEST CONDUCTED**	**TEST RESULT**
1	ADMIN SIGN IN	Login portal should authenticate QA admin credentials successfully	Black-box	Pass
2	CREATE QUESTIONNAIRE TEMPLATE	System should allow QA staff to create new templates for accreditation/auditing	Black box	Pass
3	SEND NOTIFICATIONS TO THE COLLABORATORS	System should notify the collaborators about upcoming auditing/accreditation event.	Black box	Pass
4	DARK MODE TOGGLE	UI adapts to dark/light theme preferences	Compatibility	Pass
5	DATA SYNCHRONIZATION	Mobile app should sync data with PostgreSQL server when online	Integration	Pass

6	CROSS-PLATFORM COMPATIBILITY	App should maintain functionality across Android/iOS devices	Compatibility	Pass
7	ADMIN PROFILE CREATION	System allows super-admin to create his/her profile	Black-box	Pass

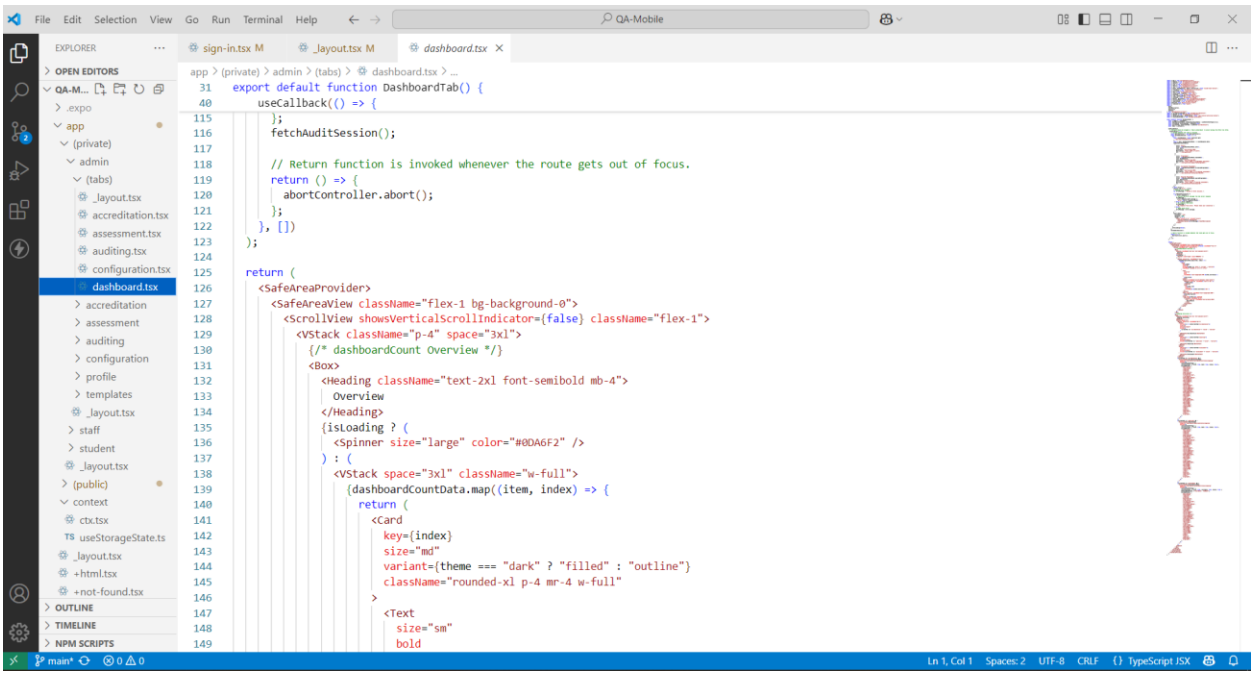
6.7 Sample codes

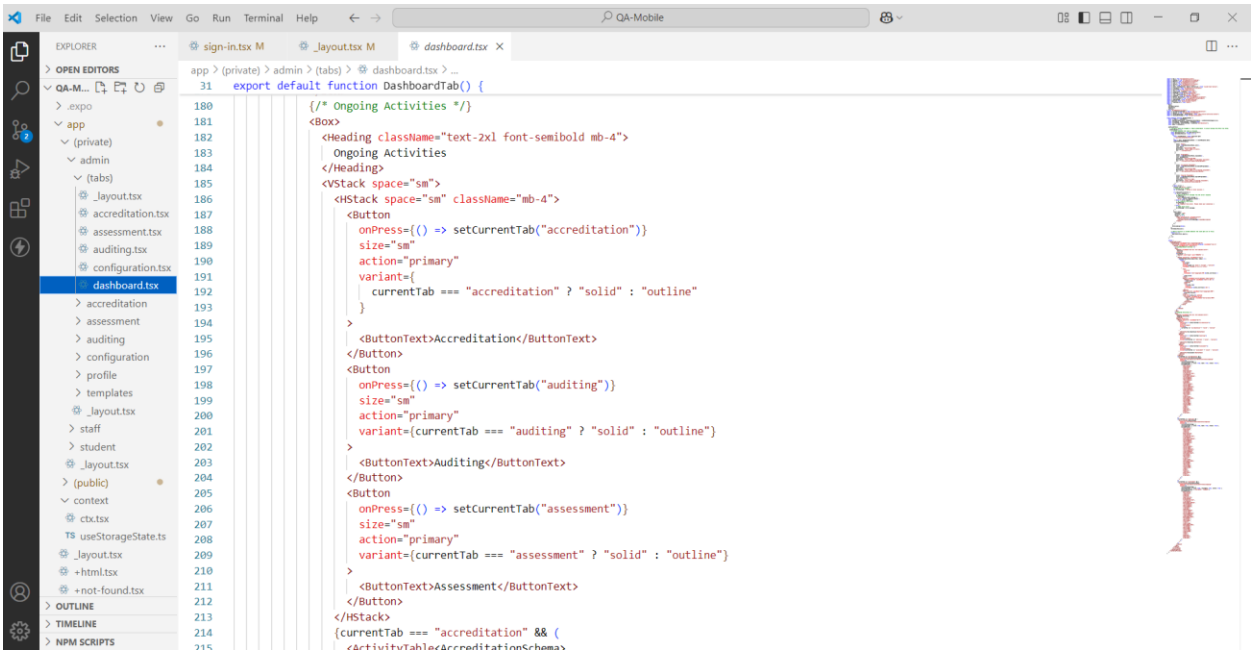


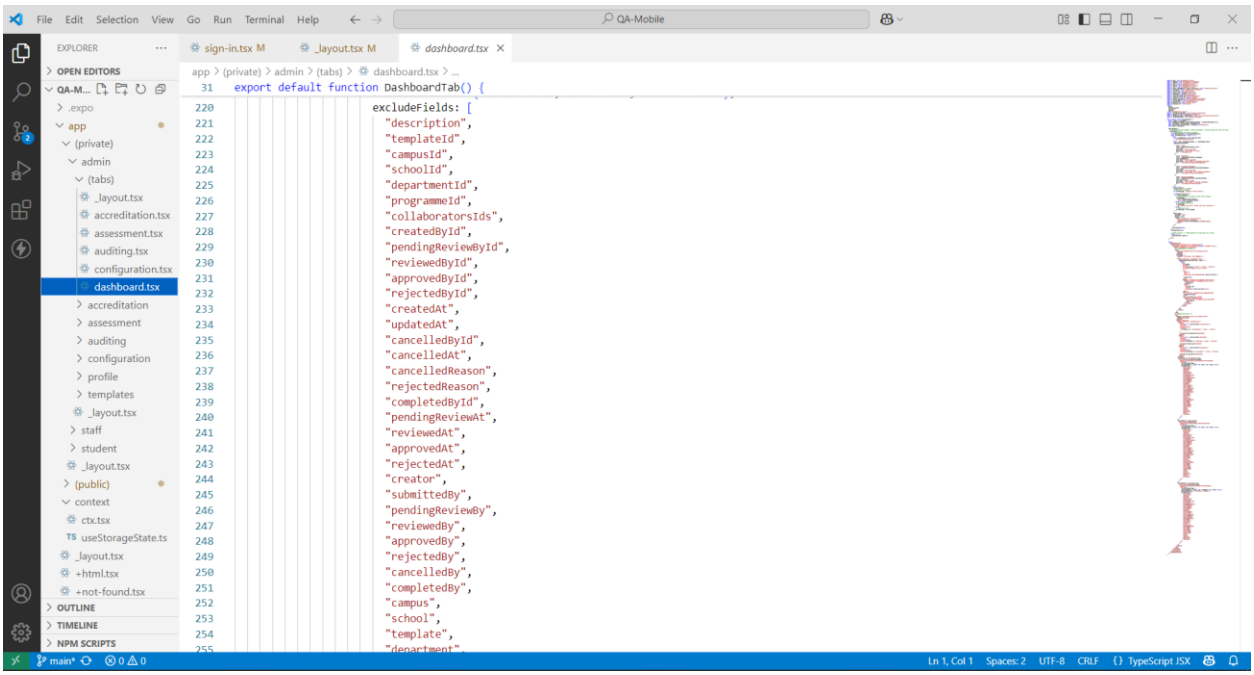
```tsx
export default function DashboardTab() {
  const { theme } = useTheme();
  const [dashboardCountData, setDashboardCountData] = useState<StatType[]>([]);
  const [isLoading, setIsLoading] = useState(true);
  const [currentTab, setCurrentTab] = useState("accreditation");
  const toast = useToast();

  useFocusEffect(
    // Callback should be wrapped in `React.useCallback` to avoid running the effect too often.
    useCallback(() => {
      // Invoked whenever the route is focused.
      const abortController = new AbortController();
      const fetchAuditSession = async () => {
        try {
          const countResponse = await apiClient.get(
            "/api/mobile/dashboard/count"
          );
          const { data: dashboardCountData } = countResponse.data;
          setDashboardCountData([
            {
              title: "Users",
              value: `${dashboardCountData.users}`,
              icon: Users,
              colorClass: "text-primary-500",
              description: "Total number of users",
              url: "../configuration",
            },
            {
              title: "Programmes",
              value: `${dashboardCountData.programmes}`,
              icon: BookOpenIcon,
              colorClass: "text-warning-500",
              description: "Total number of programmes available",
              url: "../configuration/campus?tab=programmes",
            },
            {
```

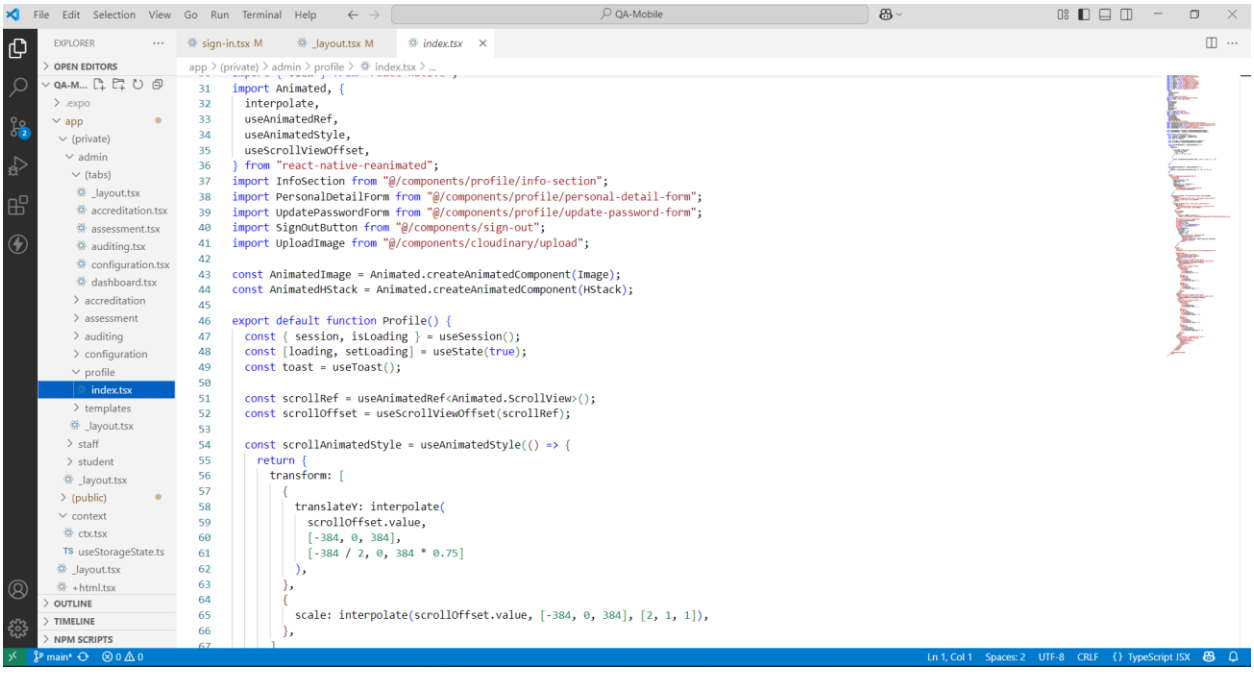


```tsx
export default function DashboardTab() {
    useCallback(() => {
      const fetchAuditSession = async () => {
        } catch (error) {
          console.log("Error", error);
          // Default error message
          let errorMessage = "Failed to fetch sessions.";

          if (isAxiosError(error)) {
            if (error.response) {
              // Extract the error message from the server response
              errorMessage =
                error.response.data?.message ||
                `Error: ${error.response.status}`;
            } else if (error.request) {
              // No response from server
              errorMessage =
                "No response from server. Please check your connection.";
            } else {
              // Other Axios error
              errorMessage = error.message;
            }
          }
          toast.show({
            placement: "top",
            duration: 3000,
            render: () => (
              <Toast action="error" variant="solid">
                <ToastTitle>Error</ToastTitle>
                <ToastDescription>{errorMessage}</ToastDescription>
              </Toast>
            ),
          });
        }
        setIsLoading(false);
      };
```


File Edit Selection View Go Run Terminal Help
QA-Mobile
EXPLORER
sign-in.tsx M
_layout.tsx M
dashboard.tsx
app > (private) > admin > (tabs) > dashboard.tsx > ...
OPEN EDITORS
QA-M...
.expo
app
(private)
admin
(tabs)
_layout.tsx
accreditation.tsx
assessment.tsx
auditing.tsx
configuration.tsx
dashboard.tsx
accreditation
assessment
auditing
configuration
profile
templates
_layout.tsx
staff
student
_layout.tsx
(public)
context
ctx.tsx
useStorageState.ts
_layout.tsx
+html.tsx
+not-found.tsx
OUTLINE
TIMELINE
NPM SCRIPTS
31 export default function DashboardTab() {
40 useCallback(() => {
115 };
116 fetchAuditSession();
117
118 // Return function is invoked whenever the route gets out of focus.
119 return () => {
120 abortController.abort();
121 };
122 }, [])
123);
124
125 return (
126 <SafeAreaProvider>
127 <SafeAreaView className="flex-1 bg-background-0">
128 <ScrollView showsVerticalScrollIndicator={false} className="flex-1">
129 <VStack className="p-4" space="3xl">
130 {/* dashboardCount Overview */}
131 <Box>
132 <Heading className="text-2xl font-semibold mb-4">
133 Overview
134 </Heading>
135 {isLoading ? (
136 <Spinner size="large" color="#0DA6F2" />
137) : (
138 <VStack space="3xl" className="w-full">
139 {dashboardCountData.map((item, index) => {
140 return (
141 <Card
142 key={index}
143 size="md"
144 variant={theme === "dark" ? "filled" : "outline"}
145 className="rounded-xl p-4 mr-4 w-full"
146 >
147 <Text
148 size="sm"
149 bold
main* 0 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} TypeScript JSX


File Edit Selection View Go Run Terminal Help
QA-Mobile
EXPLORER
sign-in.tsx M
_layout.tsx M
dashboard.tsx
app > (private) > admin > (tabs) > dashboard.tsx > ...
OPEN EDITORS
QA-M...
.expo
app
(private)
admin
(tabs)
_layout.tsx
accreditation.tsx
assessment.tsx
auditing.tsx
configuration.tsx
dashboard.tsx
accreditation
assessment
auditing
configuration
profile
templates
_layout.tsx
staff
student
_layout.tsx
(public)
context
ctx.tsx
useStorageState.ts
_layout.tsx
+html.tsx
+not-found.tsx
OUTLINE
TIMELINE
NPM SCRIPTS
31 export default function DashboardTab() {
180 {/* Ongoing Activities */}
181 <Box>
182 <Heading className="text-2xl font-semibold mb-4">
183 Ongoing Activities
184 </Heading>
185 <VStack space="sm">
186 <HStack space="sm" className="mb-4">
187 <Button
188 onPress={() => setCurrentTab("accreditation")}
189 size="sm"
190 action="primary"
191 variant={
192 currentTab === "accreditation" ? "solid" : "outline"
193 }
194 >
195 <ButtonText>Accreditation</ButtonText>
196 </Button>
197 <Button
198 onPress={() => setCurrentTab("auditing")}
199 size="sm"
200 action="primary"
201 variant={currentTab === "auditing" ? "solid" : "outline"}
202 >
203 <ButtonText>Auditing</ButtonText>
204 </Button>
205 <Button
206 onPress={() => setCurrentTab("assessment")}
207 size="sm"
208 action="primary"
209 variant={currentTab === "assessment" ? "solid" : "outline"}
210 >
211 <ButtonText>Assessment</ButtonText>
212 </Button>
213 </HStack>
214 {currentTab === "accreditation" && (


File Edit Selection View Go Run Terminal Help
QA-Mobile
EXPLORER
sign-in.tsx M
_layout.tsx M
dashboard.tsx
app > (private) > admin > (tabs) > dashboard.tsx > ...
export default function DashboardTab() {
excludeFields: [
"description",
"templateId",
"campusId",
"schoolId",
"departmentId",
"programmeId",
"collaboratorsIds",
"createdById",
"pendingReviewById",
"reviewedById",
"approvedById",
"rejectedById",
"createdAt",
"updatedAt",
"cancelledById",
"cancelledAt",
"cancelledReason",
"rejectedReason",
"completedById",
"pendingReviewAt",
"reviewedAt",
"approvedAt",
"rejectedAt",
"creator",
"submittedBy",
"pendingReviewBy",
"reviewedBy",
"approvedBy",
"rejectedBy",
"cancelledBy",
"completedBy",
"campus",
"school",
"template",
"department"
OPEN EDITORS
QA-M...
.expo
app
(private)
admin
(tabs)
_layout.tsx
accreditation.tsx
assessment.tsx
auditing.tsx
configuration.tsx
dashboard.tsx
accreditation
assessment
auditing
configuration
profile
templates
_layout.tsx
staff
student
_layout.tsx
(public)
context
ctx.tsx
useStorageState.ts
_layout.tsx
+html.tsx
+not-found.tsx
OUTLINE
TIMELINE
NPM SCRIPTS
main*
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
TypeScript JSX


File Edit Selection View Go Run Terminal Help
QA-Mobile
EXPLORER
sign-in.tsx M
_layout.tsx M
index.tsx
app > (private) > admin > profile > index.tsx > ...
import Animated, {
interpolate,
useAnimatedRef,
useAnimatedStyle,
useScrollViewOffset,
} from "react-native-reanimated";
import InfoSection from "@/components/profile/info-section";
import PersonalDetailForm from "@/components/profile/personal-detail-form";
import UpdatePasswordForm from "@/components/profile/update-password-form";
import SignOutButton from "@/components/sign-out";
import UploadImage from "@/components/cloudinary/upload";
const AnimatedImage = Animated.createAnimatedComponent(Image);
const AnimatedHStack = Animated.createAnimatedComponent(HStack);
export default function Profile() {
const { session, isLoading } = useSession();
const [loading, setLoading] = useState(true);
const toast = useToast();
const scrollRef = useAnimatedRef<Animated.ScrollView>();
const scrollOffset = useScrollViewOffset(scrollRef);
const scrollAnimatedStyle = useAnimatedStyle(() => {
return {
transform: [
{
translateY: interpolate(
scrollOffset.value,
[-384, 0, 384],
[-384 / 2, 0, 384 * 0.75]
),
},
{
scale: interpolate(scrollOffset.value, [-384, 0, 384], [2, 1, 1]),
},
OPEN EDITORS
QA-M...
.expo
app
(private)
admin
(tabs)
_layout.tsx
accreditation.tsx
assessment.tsx
auditing.tsx
configuration.tsx
dashboard.tsx
accreditation
assessment
auditing
configuration
profile
index.tsx
templates
_layout.tsx
staff
student
_layout.tsx
(public)
context
ctx.tsx
useStorageState.ts
_layout.tsx
+html.tsx
OUTLINE
TIMELINE
NPM SCRIPTS
main*
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
TypeScript JSX

CHAPTER SEVEN

SYSTEM DOCUMENTATION

User Manual for the system

7.0 System Components

7.1 User Login Page

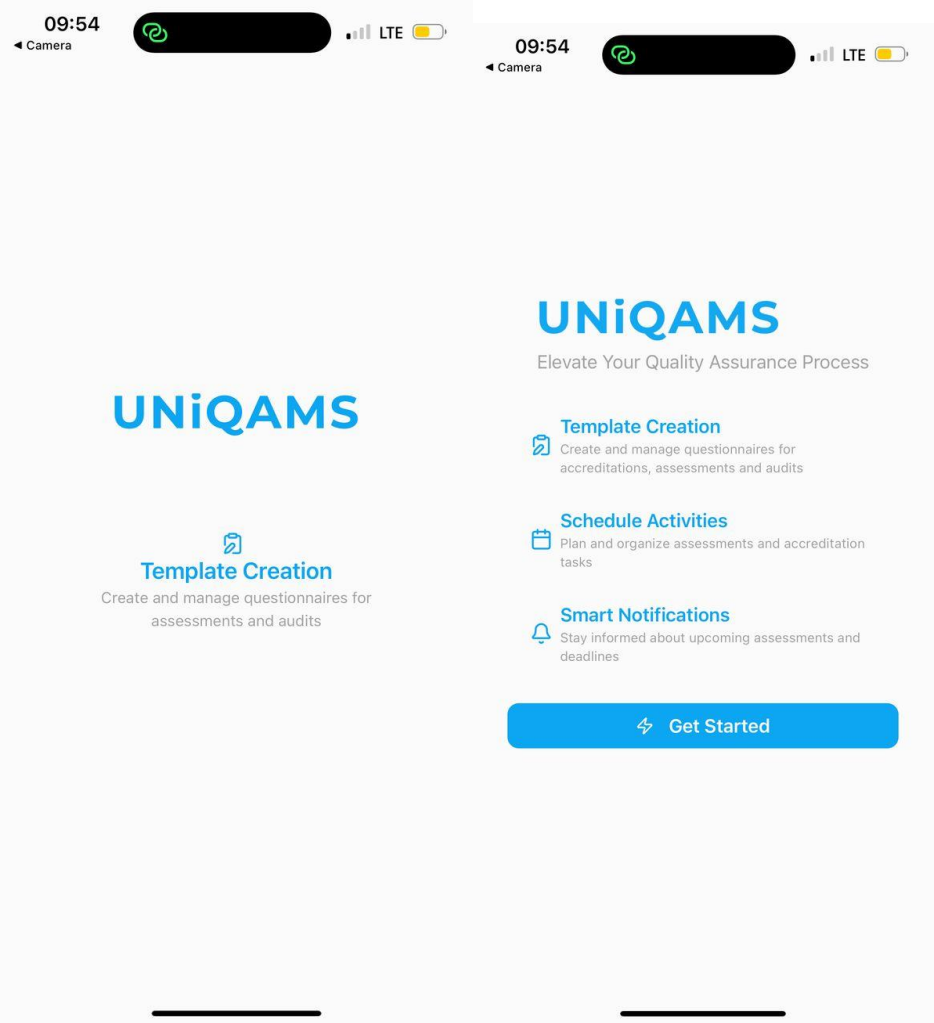


Figure 7.1

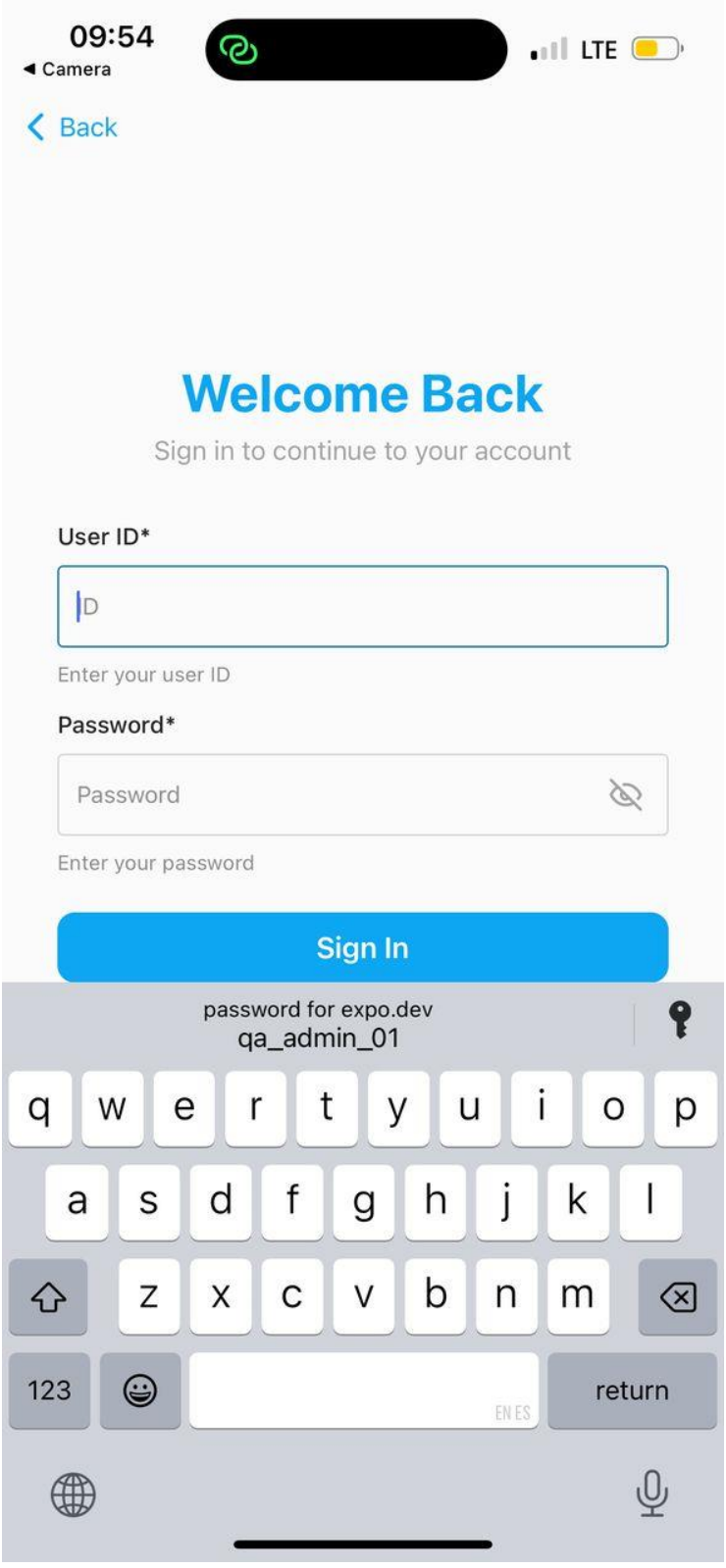

09:54
Camera
LTE
Back
Welcome Back
Sign in to continue to your account
User ID*
D
Enter your user ID
Password*
Password
Enter your password
Sign In
password for expo.dev
qa_admin_01
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
123
EN ES
return

Figure 7.1

7.2 Dashboard

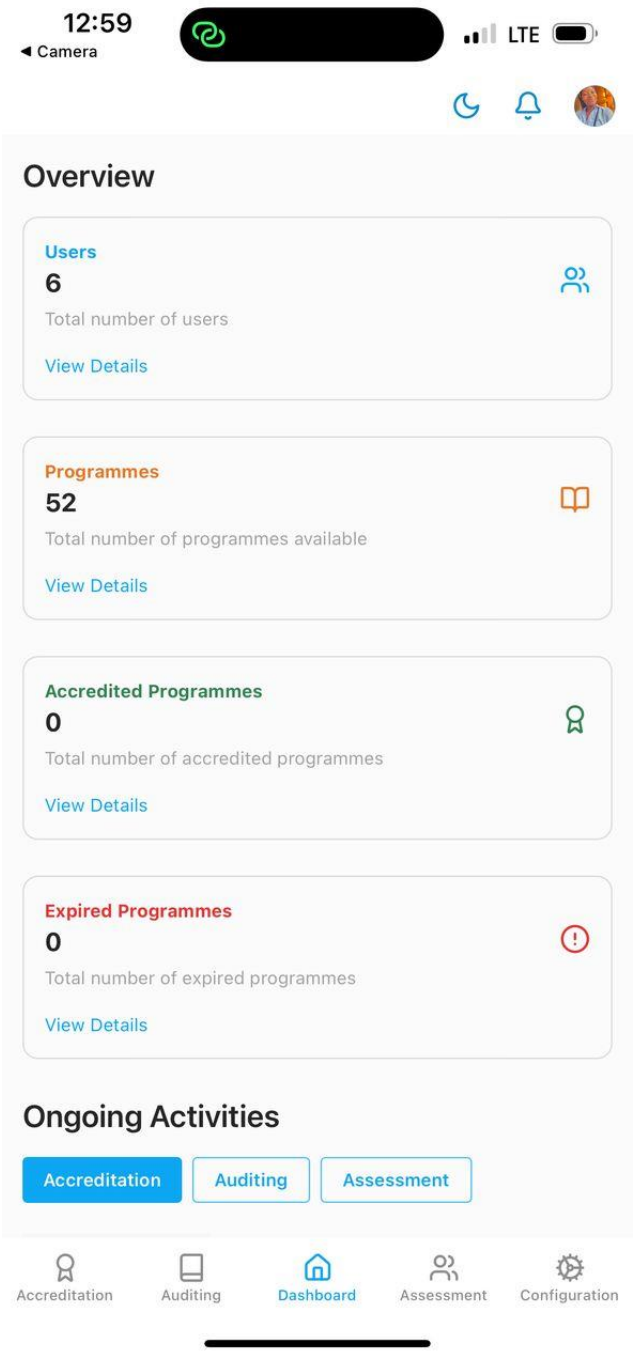


Figure 7.2

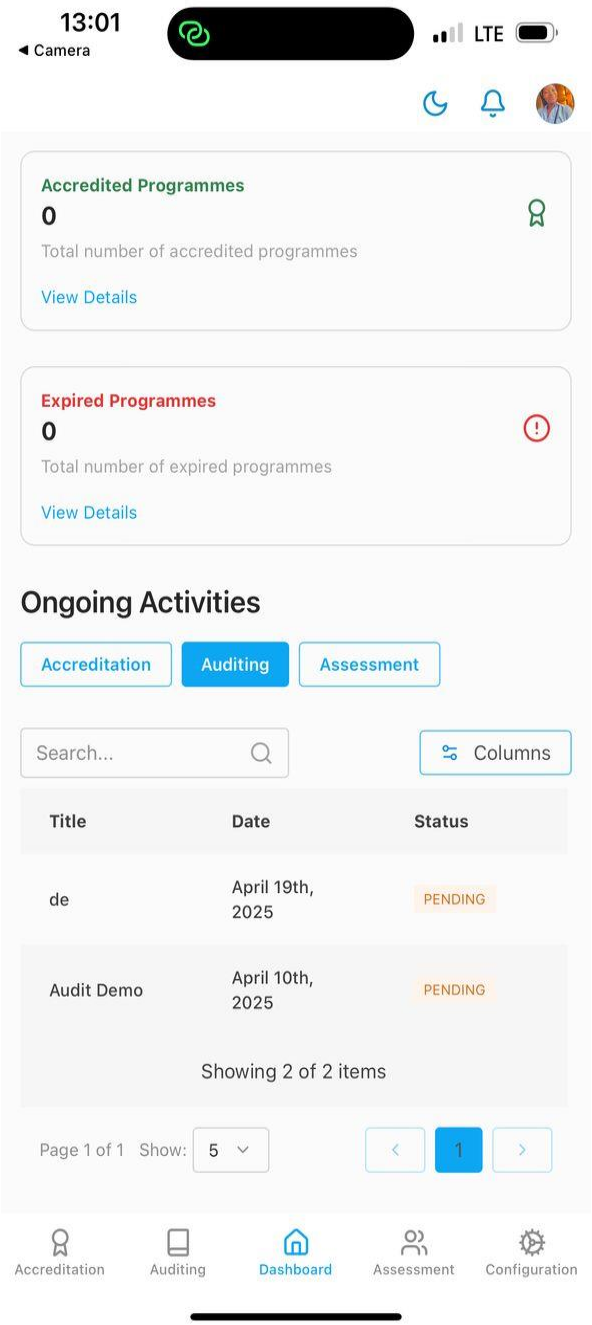


Figure 7.2.1

7.3 Admin Configuration Page

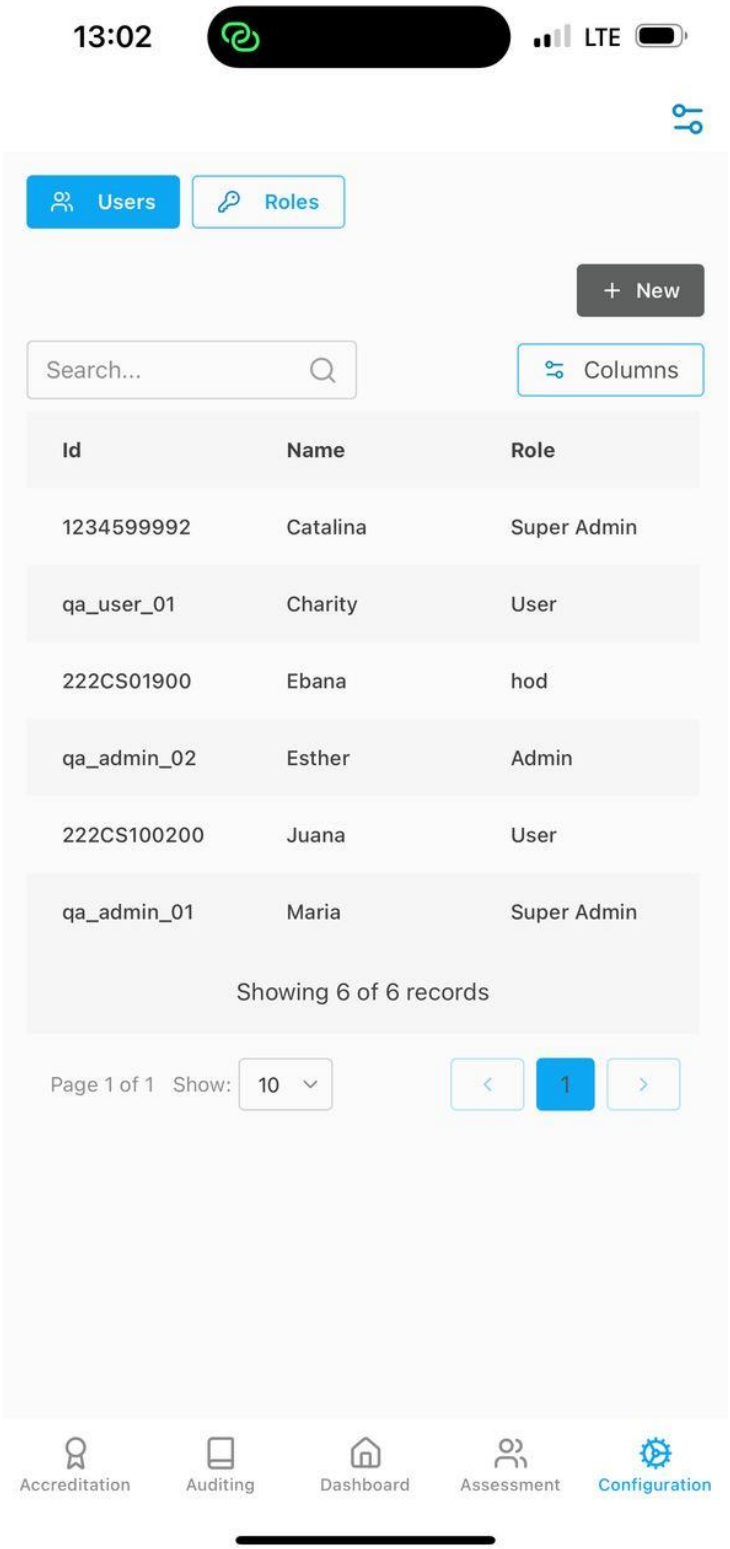


Figure 7.3

7.4 Admin Adding Users

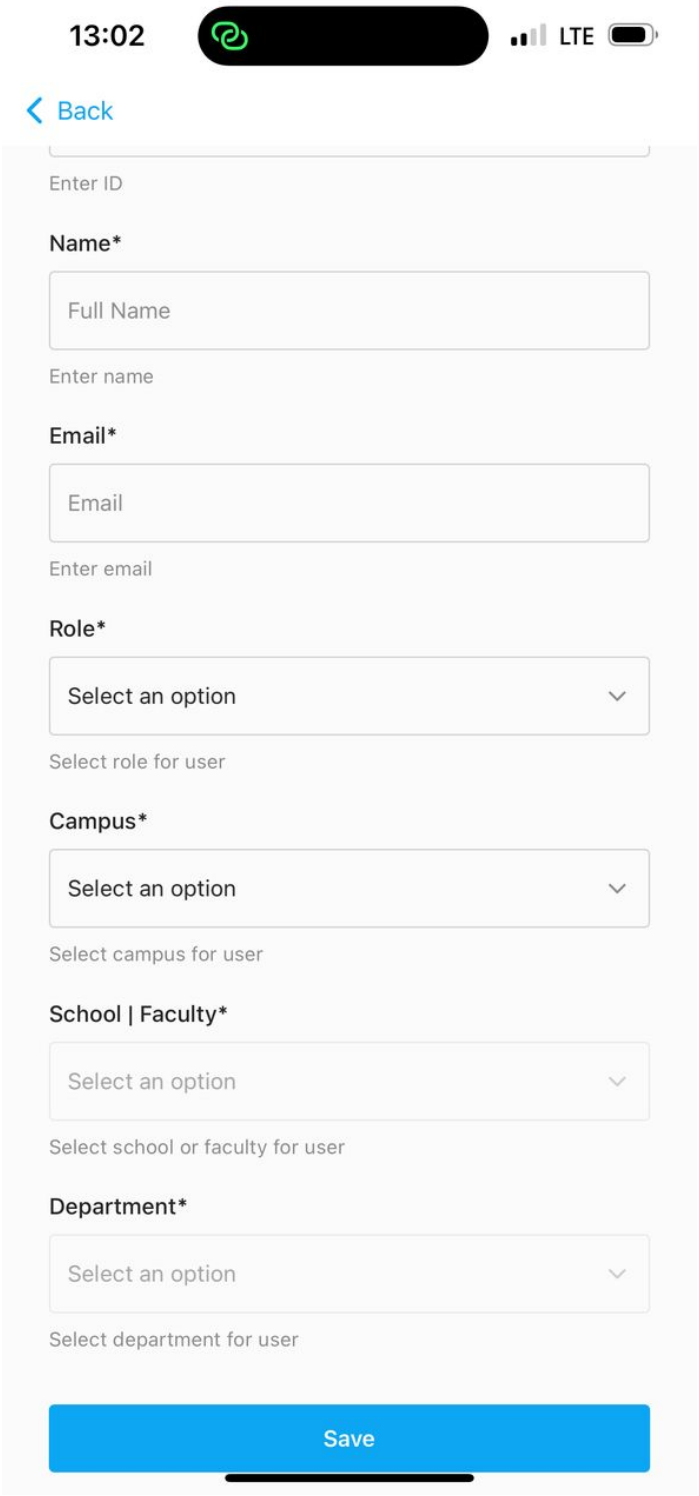


Figure 7.3.1

7.5 Admin Setting User’s Roles

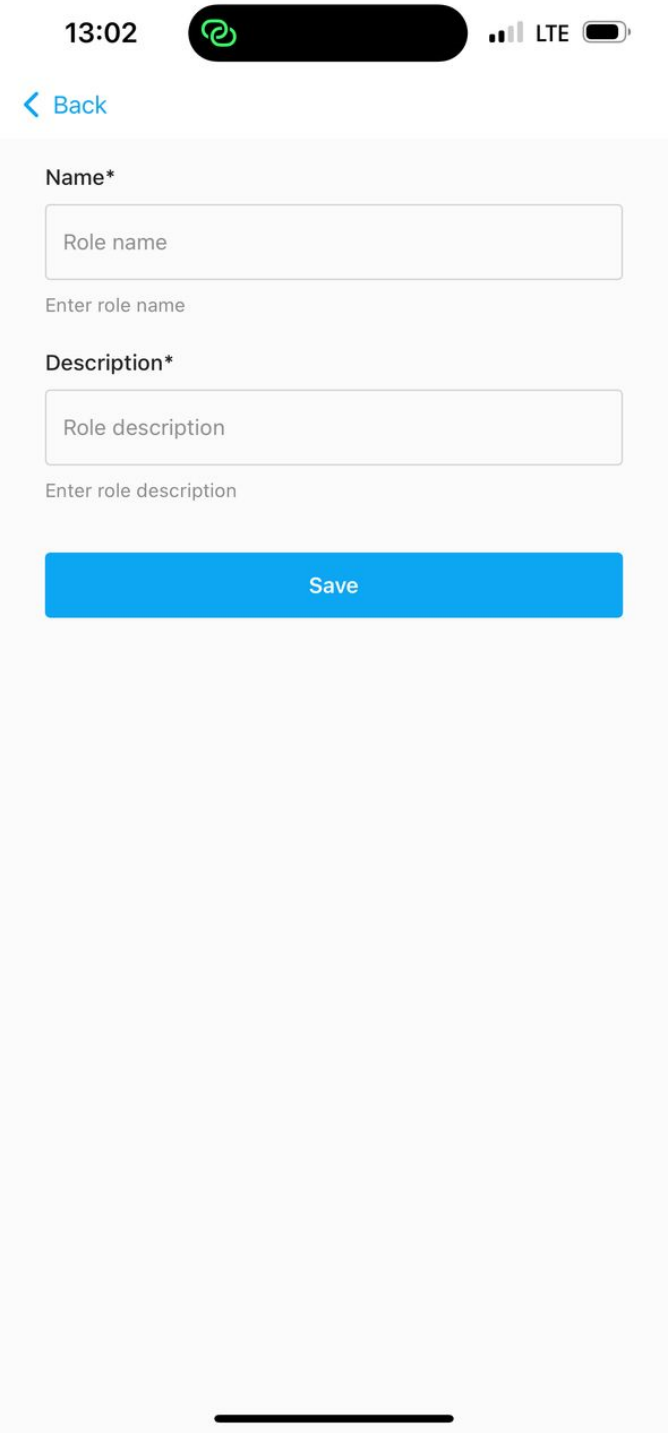


Figure 7.3.2

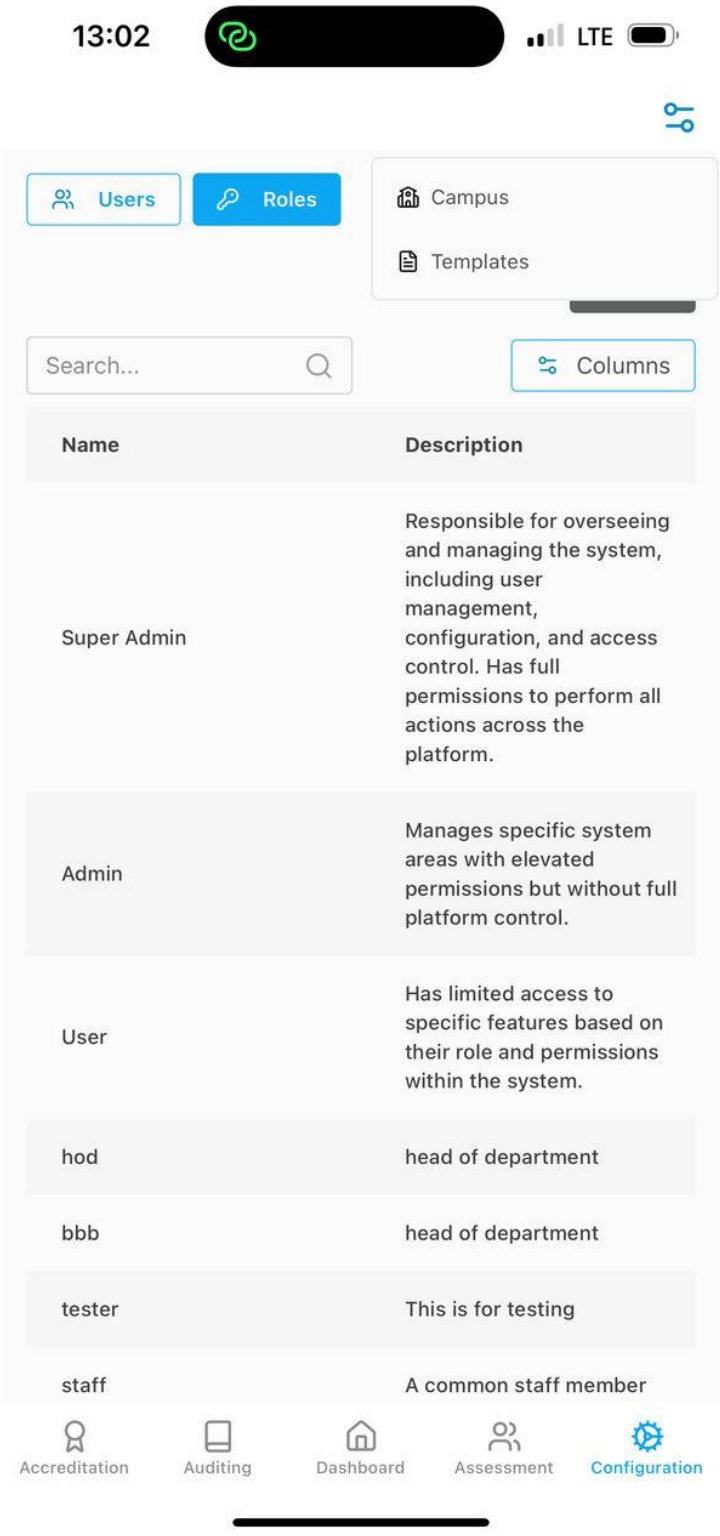

13:02
LTE
Users
Roles
Campus
Templates
Search...
Columns
Name
Description
Super Admin
Responsible for overseeing and managing the system, including user management, configuration, and access control. Has full permissions to perform all actions across the platform.
Admin
Manages specific system areas with elevated permissions but without full platform control.
User
Has limited access to specific features based on their role and permissions within the system.
hod
head of department
bbb
head of department
tester
This is for testing
staff
A common staff member
Accreditation
Auditing
Dashboard
Assessment
Configuration

Figure 7.3.3

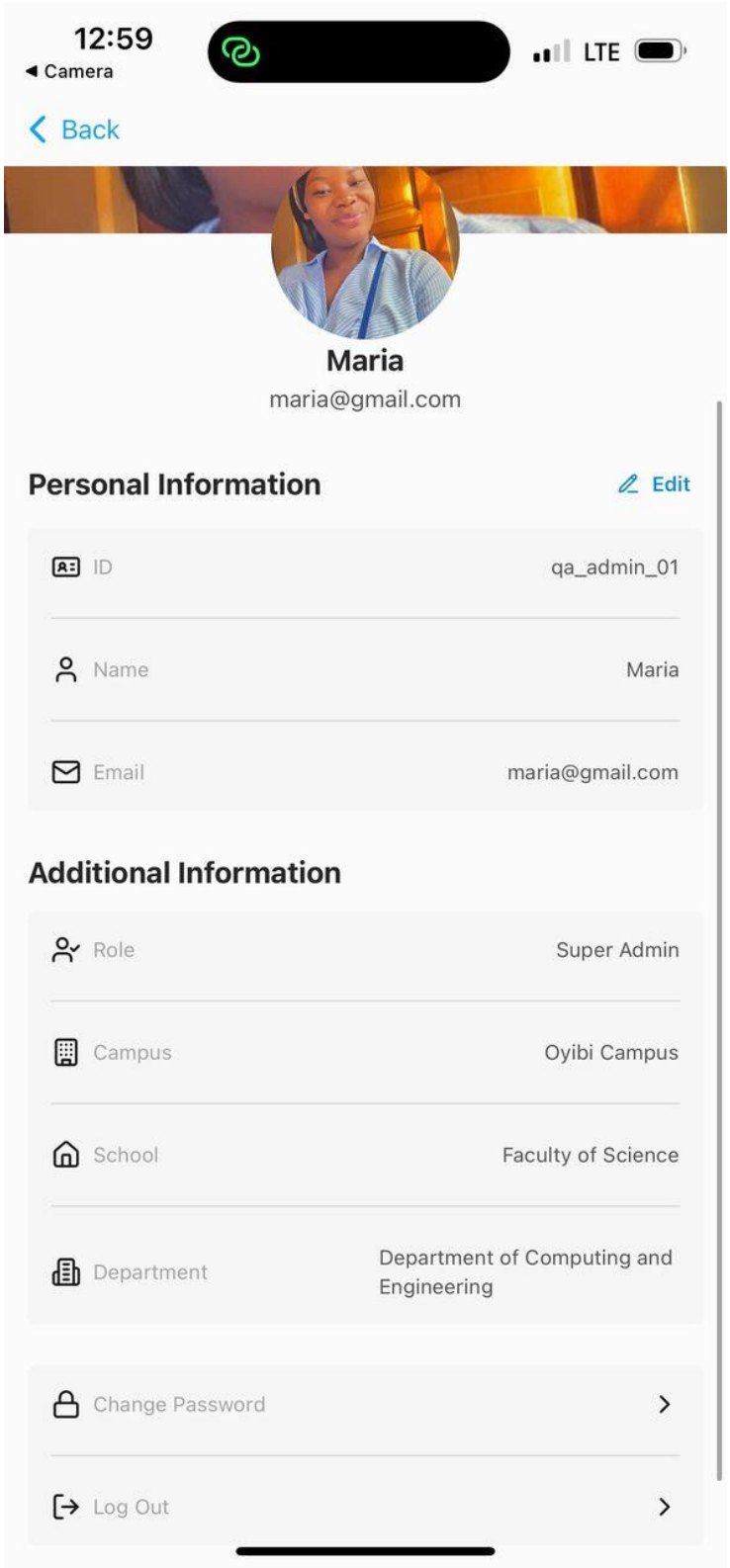

12:59
Camera
LTE
Back
Maria
maria@gmail.com
Personal Information
Edit
ID
qa_admin_01
Name
Maria
Email
maria@gmail.com
Additional Information
Role
Super Admin
Campus
Oyibi Campus
School
Faculty of Science
Department
Department of Computing and Engineering
Change Password
Log Out

Figure 7.3.4

7.6 Selection of questionnaires according to the program

Figure 7.5

7.7 Selection of questionnaires according to the school

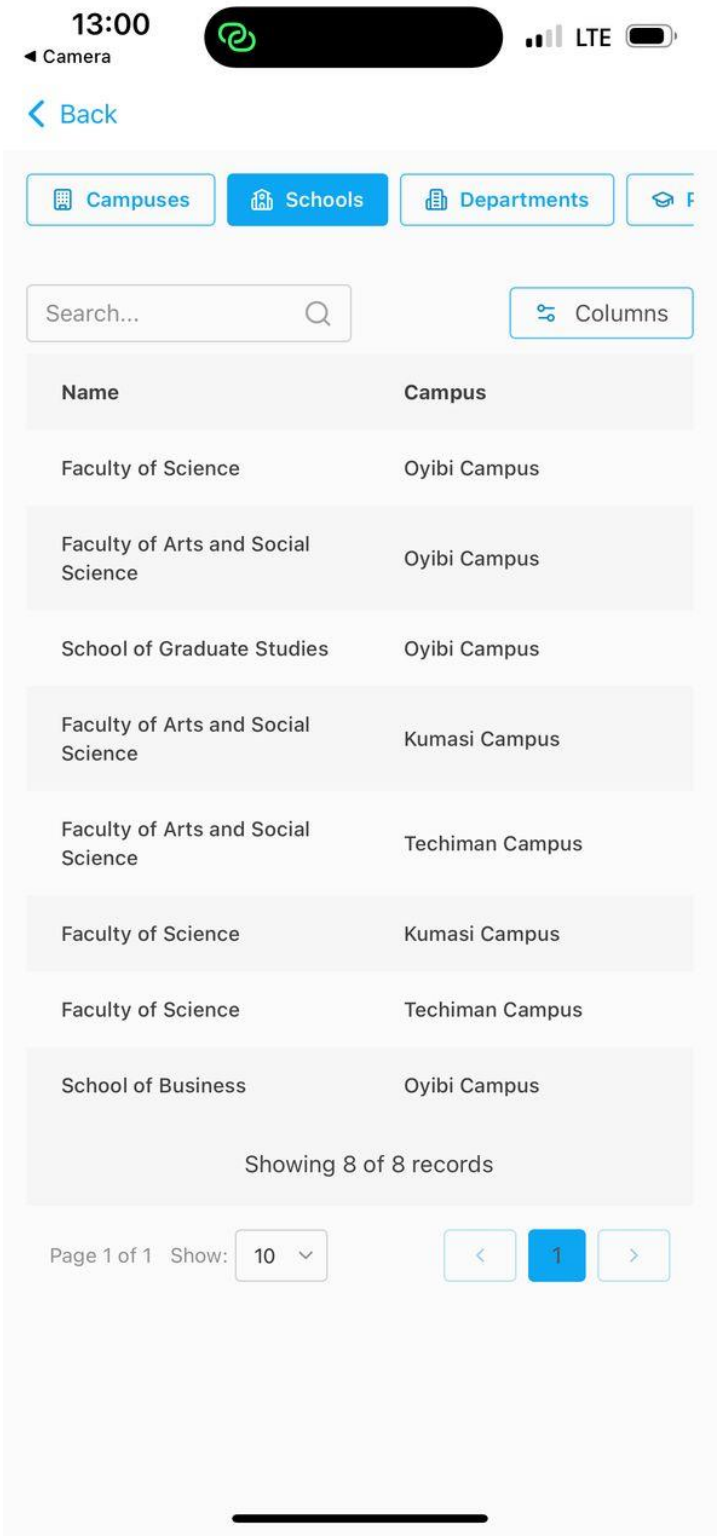


Figure 7.6

7.8 Selection of questionnaires according to the department

Figure 7.7

7.9 Selection of questionnaires according to the campus

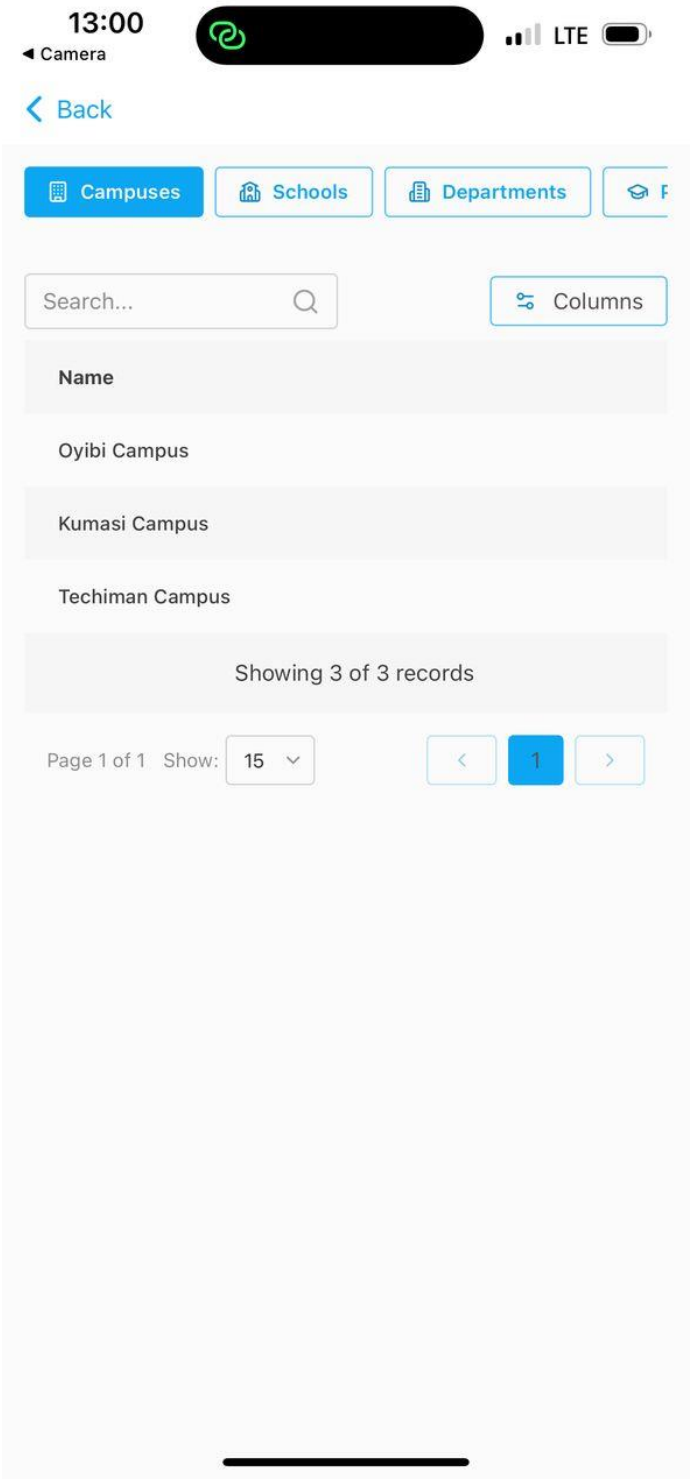
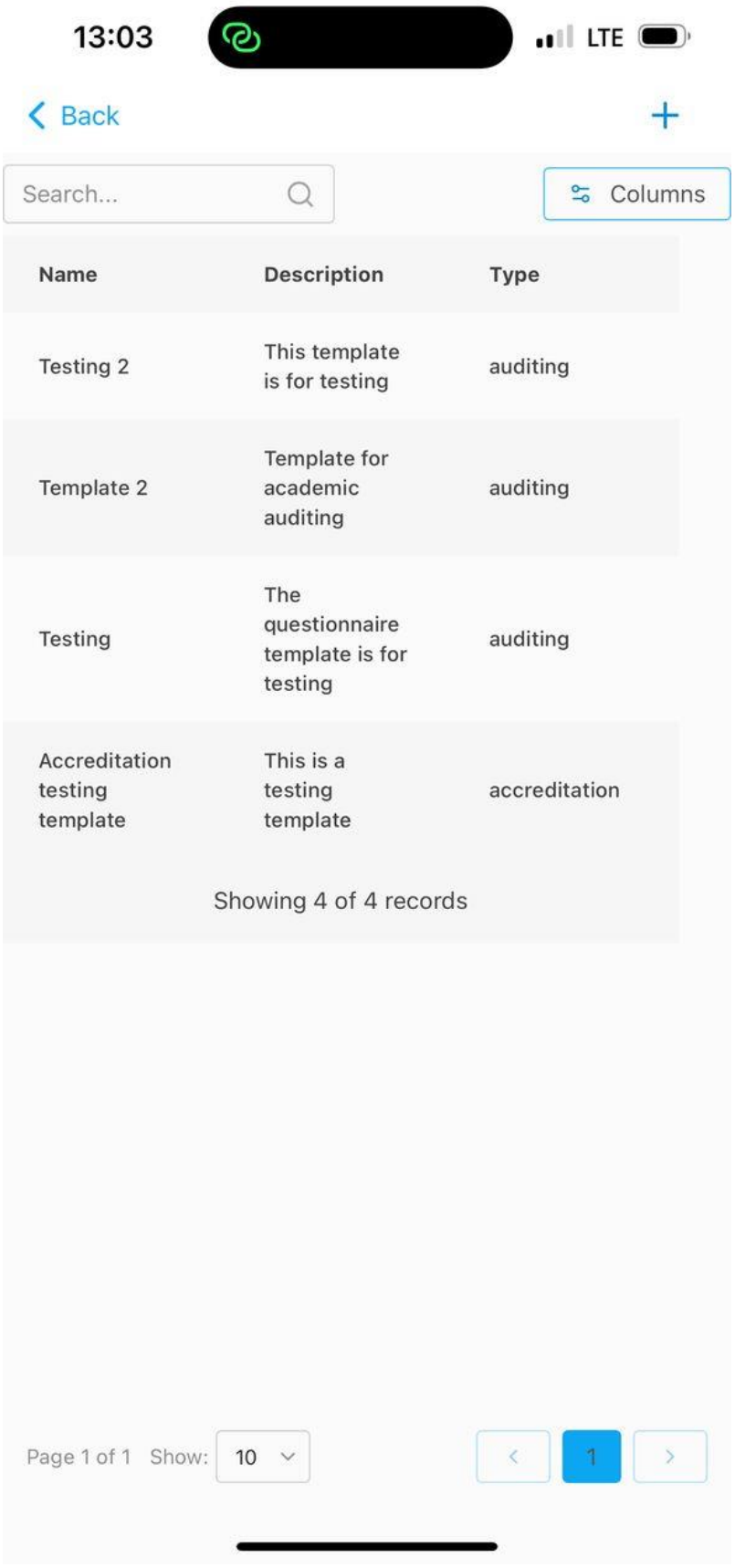


Figure 7.8


13:03
LTE
Back
Search...
Columns
Name
Description
Type
Testing 2
This template is for testing
auditing
Template 2
Template for academic auditing
auditing
Testing
The questionnaire template is for testing
auditing
Accreditation testing template
This is a testing template
accreditation
Showing 4 of 4 records
Page 1 of 1
Show:
10
1

CHAPTER EIGHT

CONCLUSION AND RECOMMENDATION

CONCLUSION

In conclusion, the development of UniQAMS for Valley View University aims to modernize and streamline quality assurance processes, including accreditation, assessments, and auditing. By automating these processes, the system will eliminate inefficiencies and delays, improve document management, communication, and ensure compliance with accreditation standards. The project addresses key challenges like data management and system integration, with a focus on user-friendly features. UniQAMS will enhance the efficiency of the QA department and support continuous improvement, positioning the university to meet the growing demands for quality education and academic excellence.

RECOMMENDATION

While the current implementation of the University Quality Assurance Management System (UniQAMS) provides a strong foundation for improving the quality assurance processes at Valley View University, there are additional functionalities that should be considered for future development. One of the most essential components that remains to be fully integrated is the assessment feature. Although assessment plays a critical role in evaluating academic quality and institutional performance, the current system does not yet include a comprehensive module for managing assessment activities. Including this functionality will enhance the ability to monitor, evaluate, and improve academic delivery in real time.

In addition, the system would benefit from the integration of graphical data collection and reporting tools. Visual elements such as graphs and charts can offer immediate and intuitive insights into key performance indicators, audit results, and accreditation statuses. This will support data-driven decision-making and make reporting more efficient and engaging for stakeholders across departments.

Furthermore, to support the university's diverse community of users, implementing multi-language functionality is strongly recommended. Providing the option for users to interact with the application in different languages will make the system more

inclusive and accessible, especially for international staff, students, and external stakeholders involved in quality assurance processes.

These enhancements will contribute significantly to the continued evolution of UniQAMS, ensuring it remains scalable, user-friendly, and aligned with the growing demands of a modern academic environment.

REERENCES

Aburizaizah, S. j. (2022). The role of quality assurance in Saudi higher education institutions. International Journal of Educational Research Open 3(4):100127, 20-22.

Alexander V.Rechinskiy, Liudmila V.Chernenkaya, Vladimir E.Mager. (2020). System for Quality Assurance of Study Programs in the Area of Engineering Education. International Conference on Information Technologies in Engineering Education (Inforino) (pp. 1-16). Moscow, Russia: IEEE.

Allee, S. (2015). EQAA Accreditation Services. Retrieved from eqaa.eu: https://eqaa.eu/en//accreditation/accreditation-services/

Arifudin and Jesry. (2022).

ASME. (2006). ASME Nuclear Quality Assurance (NQA-1) Certification. Retrieved from asme.org.

Belmont. (2020). System Certification and Quality Assurance City of Belmont. Retrieved from belmont.wa.gov.au: https://www.belmont.wa.gov.au/about-us/our-city/our-organisation/system-certification-and-quality-assurance#

Brilliant. (2019). Quality Assurance & Accreditation. Retrieved from touchnetix.com: https://www.touchnetix.com/quality-assurance-accreditation/

Bunyamin , Alamsyah and Fadhli. (2020).

Chalaris, M. (n.d.).

Chalaris, M. (2011). Developing an Information System for Quality Assurance in Higher Education using the Balanced Scorecard Technique - The Case Study of TEI-A.

CHEA. (2020). Quality Assurance Agency for Higher Education (QAA). Retrieved from chea.org: https://www.chea.org/international-directory/quality-assurance-agency-higher-education-0

Ebène. (2020, January 17). Quality Assurance Authority. Retrieved from qaa.ac.mu: https://qaa.ac.mu/qainmauritius

EFMD Global Network’s Quality Assurance Academy . (n.d.).

Eric Cornuel, Howard Thomas and Matthew Wood. (2006). EFMD Global. Retrieved from efmdglobal.org: https://www.efmdglobal.org/

George Ubachs, Piet Henderikx. (2023). Quality Assurance Systems for Digital Higher Education in Europe. Distance and Digital Education, 745-760.

George Ubachs, Piet Henderikx. (2023). Quality Assurance Systems for Digital Higher Education in Europe. Handbook of Open, Distance and Digital Education, 745.

Georges Vernez, Shelly Culbertson, Louay Constant, Rita Karam. (2016). Developing a School Quality Assurance System. Initiatives to Improve Quality of Education in the Kurdistan Region—Iraq: Administration, School Monitoring, Private School Policies, and Teacher Training (pp. 27-48). Santa Monica, California, United States: RAND Corporation.

Georges Vernez, Shelly Culbertson, Louay Constant, Rita Karam. (2016). Initiaves to improve Quality of Education in the Kurdistan Region-Iraq. RAND Corporation.

Hadzhikoleva, S. (n.d.).

Hadzhikoleva, S. (2020). Model of a Centralized System for Quality Assurance in Higher Education .

Haskova, A. (2016). Information systems as a component part of tertiary education quality assurance. International Conference on Application of Information and COmmunication Technologies (AICT) (pp. 9-20). Baku, Azerbaijan: IEEE.

Haskova, A. (2017). Information Systems as a Component Part of Tertiary Education Quality Assurance.

Konstantin Holl, Frank Elberzhager . (2017). Guiding Quality Assurance for Mobile Application with FIT4Apps-A Two-Step Evaluation. Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 435-439). Vienna, Austria : IEEE.

Konstantin Holl, Frank Elberzhager. (2014). A Mobile-specific Failure Classification and its Usage to Focus Quality Assurance. Conferencce Rngineering and Advanced APplications (pp. 385-388). Verona, Italy : IEEE.

Kyvik, S. (2009). The Dynamics of Change in Higher Education: Expansion and Contraction in an Organization Field.

Kyvik, S. (2009). The Dynamics of Change in Higher Education: Expansion and Contraction in an Organizational Field.

Larisa Petrova, Vyacheslav Prinkhodko, Alexander Solovyev. (2015). Integration of the Study Programmes' Quality Assurance to the internal quality management system in Russian Universities. International Conference on Interactive Collaborative Learning (ICL) (pp. 434-436). Firenze, Italy: IEEE.

Monolis Chalaris, Anastasios Tsolakidis, Cleo Sgouropoulou, Ioannis Chalaris . (2011). Developing an Information System for Quality Assurance in Higher Education using the Balanced Scorecard

Technique - The case study of TEI-A. Panhellenic Conference on Informatics (pp. 373-377). Kastoria, Greece: IEEE.

Ni Ketut Dewi Ari Jayanti and Ni Luh Ayu Kartika Yuniastari Sarja. (2019). IT as a Tool: Quality Assurance System in Higher Education .

Ni Ketut Dewi Ari Jayanti, Ni Luh Ayu Kartika Yuniastari Sarja. (2019). IT as a Tools: Quality Assurance System in Higher Education . International Conference oon Cybernestics and Intelligent System (ICORIS) (pp. 279-282). Denpasar, Indonesia: IEEE.

Nina N.Aniskina, Ekaterina V.Lunina. (2017). Integration of quality assurance models for education on the basis of comparative analysis. International Conference "Quality Management, Transport and Information Security, Information Technologies"(IT&QM&IS (pp. 616-620). St. Petersburg, Russia: IEEE.

Nurdin Nurdin, Saggaf S. Pettalongi, Muhammad Nur Ahsan. (2019). Implementation of Teaching Quality Assessment System Using Android. International Conference on Science and Technology(ICST) (pp. 1-34). Yogyakarta, Indonesia: IEEE.

Qualitest. (2018). Qualitest Group in healthcare. Retrieved from qualitestgroup.com: https://www.qualitestgroup.com/industry/healthcare/

Quality Assurance Agency-Mauritius. (n.d.).

RANZCR. (2023, September 9). Mammography Quality Assurance Program. Retrieved from ranzcr.com: https://www.ranzcr.com/fellows/clinical-radiology/quality-assurance-and-accreditation/mqap

RESNET. (2021). RESNET Quality Assurance (QA) App. Retrieved from resnet.us: https://www.resnet.us/about/qa/resnet-qa-app/

Stanka Hadzhikoleva, Daniela Orozova, Emil Hadzhikolev, Nedelcho Andonov. (2020). Model of a Centralized System for Quality Assurance in Higher Education. International Conference on Intelligent Systems (IS) (pp. 87-92). Varna, Bulgaria: IEEE.

TouchNetix-Quality Assurance Accreditation . (n.d.).

Vernez, G. (2016). Developing a School Quality Assurance System .