



UNIVERSITY OF ZIMBABWE

**FACULTY OF COMPUTER ENGINEERING, INFORMATICS AND
COMMUNICATIONS**

DEPARTMENT OF COMPUTER SCIENCE

TITLE: CENTRALIZED UNIVERSITY APPLICATION PLATFORM

NAME OF STUDENT: KANDEYA GODFREY T

REG NUMBER: R219420M

SUPERVISOR: MR. DEVE

**THIS CAPSTONE PROJECT IS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE BACHELOR HONORS IN COMPUTER SCIENCE**

DECEMBER 2024

Table of Contents

Declaration.....	i
Dedication.....	ii
Acknowledgments.....	iii
Abstract.....	iv
Chapter 1. Introduction.....	1
1.1 Introduction Centralized University Application System.....	1
1.2 Context and Background.....	1
1.3 Problem Statement.....	1
1.4 Aim.....	1
1.5 Purpose and specific objectives.....	2
1.6 Justification and Relevance.....	2
Chapter 2. Literature Review.....	3
2.1.1 Introduction.....	3
2.1.2 Theoretical Framework.....	3
2.1.3 Research Methodology.....	4
2.1.4 Methodological Approach.....	4
2.1.5 Data Structure.....	4
2.1.6 Description of Methods of Data Collection.....	5
2.1.7 Description of Methods of Analysis of Data.....	5
2.1.8 Evaluation & Justification of Methodological Choices.....	6
2.1.9 Resource Requirements.....	6
2.2.0 Software Requirements.....	6
2.2.1 Hardware Requirements.....	9
2.2.2 Functional and Network Requirements.....	9
2.2.3 Functional Requirements.....	9
2.2.4 Network Requirements.....	10
2.2.5 Non-Functional Requirements.....	10
2.2.6 Modelling Diagrams.....	11
2.2.7 Software Projects.....	13
2.2.8 Historical Context.....	13
2.2.9 Key Studies and Their Findings.....	14
2.3.0 Methodological Approaches.....	15
2.3.1 Controversies and Debates.....	17
2.3.2 Synthesis and Evaluation.....	17
2.3.3 Project Plan.....	18
2.3.3.1 Gantt Chart.....	18
2.3.4 Conclusion.....	18
Chapter 3: Methodology.....	19

3.1 Class Diagram.....	19
3.2 Object Diagram.....	20
3.3 Sequence Diagram.....	21
3.4 Communication Diagram.....	22
3.5 State Chart Diagram.....	23
3.6 Process Design Diagram.....	24
3.7 Network Diagram.....	25
Chapter 4: Results and Discussion.....	26
4.1 Introduction to Results and Discussion.....	26
4.2 Hardware and Software Specifications.....	26
4.2.1 Hardware Specifications.....	26
4.2.2 Software Specifications.....	27
4.3 Test Plan and Strategy.....	28
4.3.1 Integration Testing.....	29
4.3.2 Testing of System.....	29
Table 4.15 Applicant Search.....	35
4.4 Implementation Plan.....	44
Table 4.11: Implementation Plan.....	44
4.5 Evaluation.....	44
4.6 Findings and Interpretation.....	45
Chapter 5: Discussion and Conclusions.....	46
5.1 Discussion of Results.....	46
5.2 Conclusions.....	48
5.3 Recommendations.....	49
References.....	50

Declaration

I, KANDEYA GODFREY T (Registration Number: R219420M) do hereby declare that this capstone project entitled "Centralized University Application Platform" represents my own original work. This dissertation in whole or in part has never previously been submitted to any other university or institution for the award of a degree or diploma.

All ideas, information and data as well as any sources from which they were obtained that have been used in this dissertation have all been fully acknowledged and cited using the Harvard Referencing style. I declare that this research was done with an adherence to the principles of academic integrity and complies with all codes of ethical conduct.

Signature: _____

Date: December 2024

Dedication

This dissertation is humbly dedicated to my parents, whose unconditional support, immense encouragement and extreme patience have given the foundation to the arduous journey of this academic pursuit. It was through their belief in my capacities themselves that I drew my inspiration whenever I had to go through an uphill battle.

This is also dedicated to the countless future university hopefuls for whom having a transparent, speedier and fairer university admission procedure is not merely a convenience but more of a fundamental human right. May this research help bring their dreams of an education into reality.

Acknowledgments

I remain in debt of gratitude to all those persons and institutions that, directly or indirectly, assisted in the completion of this capstone project.

Foremost, I tender my sincere gratitude to my worthy supervisor, Mr. Deve, for his keen guidance, critical comments, and unfaltering encouragement during the entire process of the research and development. His profound insight and attention to detail immensely influenced the direction and quality of this dissertation.

My deepest thanks are extended to the Faculty of Computer Engineering, Informatics and Communications, as well as the Department of Computer Science at the University of Zimbabwe, for this undertaking in that it fortified the academic environment, resources and intellectual stimulation I so craved.

A heartfelt thank you is extended to the anonymous individuals who participated in the survey and focus groups. Their insightful feedback and perspectives not only affirmed the system's relevance but also highlighted opportunities for enhancement. They truly served as the empirical foundation for the platform.

Special mention is made here of my peers and colleagues, who through their spirit of cooperation, stimulating conversations, and common tribulations, created an atmosphere conducive and inspiring to learning.

Lastly, my family and friends, your sacrifices, understanding, and unwavering emotional sustenance shall remain the cornerstones of my perseverance. Without your constant encouragement, I would in all probability never have achieved this.

Abstract

This Capstone Project focuses on fixing the pervasive inefficiencies and lack of transparent information in traditional university admission systems. Such systems are typically hindered by scattered inflexible workflows that rely on outdated manual methods overflowing with inconsistent logic chains, creating bottlenecks at every step. For applicants who must navigate a complex multi-institutional application, these systems slow down processing time across a multitude of universities. Increased operational costs and overwhelmed administrators due to constant workload surges is yet another issue for struggling universities.

Adopting an Agile development framework with mixed-methods research resulted in this project's goal to create purpose-built solutions through implementing novel online centralized platforms, guiding users seamlessly while using them, regardless of their backgrounds or previous experiences. A central aspiration was to enable streamlined multi-university applications through a single interface alongside automated evaluations and tailored smart suggestions based on applicant profiles and preferences. Equally essential is the implemented AI assistant chatbot feature which ensures prompt access to reliable support around the clock by accessing the entire system database containing institution, program and application information enabling smooth navigation within its ecosystem. The system also includes advanced modules capable of sending real-time email and SMS notifications thus keeping applicants updated on status changes instantaneously guaranteeing seamless tracking throughout each stage.

Survey responses, focus group interactions, and performance system logs show a clear improvement in both the effectiveness of the processes as well as user satisfaction. For example, a notable reduction in the administrative burden, processing errors, and streamlined communication through automated notifications were all identified. Users especially appreciated the efficiency and accuracy with which information was dispensed by the AI Assistant Chatbot.

As stated above, there are no doubts that the Centralized University Application Platform is a sustainable infrastructure that adheres to the guidelines of DDDM , Artificial Intelligence in Education (AIED) , as well as E-Government Theory. They not only automated an educational

entry process but also advanced equity, access, transparency increasing responsiveness for today's diverse learners.

Chapter 1. Introduction.

1.1 Introduction Centralized University Application System

1.2 Context and Background

The increasing demand for higher education has made a strong case for a better streamlined, efficient process for university applications. Has there ever been any traditional system of university admission, which involved such fragmented, manual systems that simply required applicants to submit different applications for various institutions?.Such applications were all individual and separate, so that any student interested in applying for several universities would have to fill out several applications, one each for the institutions where he or she wants to be admitted.

Inevitably, this translation into inefficiency brings delays, mismanagement of applications, and errors in the handling of documents. Consider examples such as the Universities and Colleges Admissions Service (UCAS) in the UK: delivering on promises of centralized platforms having subsumed applications submission, document verification, and recommendation systems (Smith & Watson, 2019). This project seeks to develop an online centralized platform for the same, customized according to the needs of the particular university admissions.

1.3 Problem Statement

The inefficiency with which the present university admission system runs, along with the time that is consumed and human errors that are likely to occur when these processes are done manually, affects the applicants directly, who have difficulty applying for more than one university. It results in a lack of transparency and fairness as well. On the other hand, institutions when going through this again find themselves with an overload of administrative work that results in late processing and, of course, higher operational costs. Automated systems have been proposed to ameliorate and address those weaknesses and ensure that there is fairness to the whole process and, if possible, reduce manual labor (Brown et al., 2020).

1.4 Aim

Development of an AI powered centralized university application platform for applications, automated document verification, and qualification checks. This will boost efficiency, equity, and access towards students and relieve the universities from administrative burdens.

1.5 Purpose and specific objectives

This is a project whose intention is to revolutionize the application process to universities by:

- Establishing a single interface for students to apply to several universities at once.
- Automated systems of document verification and qualification assessments.
- Smart recommendations to applicants based on their profiles and preferences using AI.
- Reduced operational costs borne by universities through automation.
- Data security and privacy are to be ensured through encrypted storage and strong access control (SMITH & WATSON, 2019; JOHNSON, 2021)

1.6 Justification and Relevance

A single unified application platform will vastly enhance application processes today, resulting in a more economical and fair access to applications. The establishment of UCAS shows how centralized portals can actually increase transparency and efficiency while at the same time ensuring the highest standards of data safety with examples from other similar orders (UCAS Annual Report, 2022). Moreover, this will provide equal and fair access for all students, especially in rural or deprived areas, towards achieving diverse participation in higher education opportunities so as to fulfill global great goals towards achieving inclusive education (United Nations Sustainable Development Goals, 2015).

Chapter 2. Literature Review

2.1.1 Introduction

Presently, the current processes of admission to universities are mainly manual in nature and hence do not take less time and perform less number of functions. The same under deffical centers of students has to apply to each deffical institution which can instead end up causing acceptance to different universities that do not meet any qualifications. Also, where it manual verification of documents, it unnecessarily increases administrative overhead burdens as well as human errors. One of the solutions proposed to these problems is a centralized university application platform which makes it possible to eliminate any mediators and use fully automated systems of document verification, qualification checks as well as smart application recommendations. Effectively such systems not only speed up the process of admissions but also make it more fair and easy to access. Similar cases of other systems like the UCAS of UK shows that centralized systems are capable of enhancing transparency and efficiency while cutting down costs for the institutions and applicants in the process (Smith & Watson, 2019).

2.1.2 Theoretical Framework

The system architecture that has been proposed is grounded in three significant theoretical frameworks Data-Driven Decision Making (DDDM), Artificial Intelligence in Education (AIED), and E-Government Theory.

1. Data-Driven Decision Making (DDDM)

Data-informed decision-making is a major tenet of this theory. The suggested system will utilize applicant information to evaluate qualifications and use AI algorithms to suggest the best possible placements. Managing large volumes of information makes it possible to support evidence-based decision making in university admissions processes (Mandinach & Gummer, 2016).

2. Artificial Intelligence in Education (AIED)

The integration of AI in education has significantly increased over the years with the provision of solutions such as personalized learning and predictive analytics. This theoretical structure forms the basis for the recommendation engine of the proposed system which matches students to appropriate programs according to their preferences and qualifications to achieve the best results for both students and institutions (Luckin et al., 2016).

3. E-Government Theory

Derived from the discipline of public management, this theory aims to enhance service delivery by the use of information communication technologies. In the context of university admissions, this assists in ensuring that all people, including those from remote and marginalised areas of the country, can be included, reach out to and enhanced with the system at a higher scale (Heeks, 2006). This approach also enhances the design so that people with different backgrounds or levels of education / training are able to make use of it (Heeks, 2006).

2.1.3 Research Methodology

Research approach integrates both qualitative and quantitative approaches to assess the user effectiveness and efficiency of the platform. For such an investigation a combination of survey and focus group, together with system-generated performance evaluation will be utilized. This approach guarantees strong feedback mechanisms designed for iterative development (Creswell & Creswell, 2018).

2.1.4 Methodological Approach

The process of system development is implemented within the Agile framework, which is ideal for projects where actual users' feedback is compulsory and subjective improvement is carried out. The iterative cycles of Agile permit fast modification of the product following user feedback to improve functionality and efficiency in the course of development (Beck et al., 2001). This method goes further by allowing in-flight coping mechanisms for complications that arise in the processing of large volumes of data and diversity of users.

2.1.5 Data Structure

The system will rely on three main types of data

1. User Information

This is obtained through application forms and surveys within the system. It consists of student's profiles, their academic records, and choices.

2. Performance Data

Logs created by the system on parameters like server capacity, how long an application takes to be processed, and the number of failures recorded will benefit system tuning.

3. Application and Qualification Contextual Data

This encompasses information like criteria for programs, minimum requirements needed, and statuses of application processes for effective matching and suggesting policies.

2.1.6 Description of Methods of Data Collection

The data collection methods have been tailored to obtain more view on the system performance and the user experience

1. Surveys

They were carried out among the users and the institution staffs for the evaluation of the system's recommendations in terms of usability, efficiency and accuracy. These Surveys aim at determining the shortfalls and the scope of betterment (Dillman et al., 2014).

2. System Logs

These are the automatic logs generated for the purpose of gauging the latencies, throughput, error rate and other operational indicators. These logs will be of essence in maintaining the systems, diagnosing troub-shooting the systems and doing systems reliability enhancements.

2.1.7 Description of Methods of Analysis of Data

Three analytical methods will be used to analyze the data that has been collected

1. Statistical Analysis

Descriptive statistics will be used to analyze the survey responses, especially focusing on user satisfaction levels and the effectiveness of application results.

2. Performance Analysis

System logs will be analyzed to determine the performance of the system and determine how it can be improved by examining factors such as the responsiveness of the system and scalability.

3. Validation of Recommendations

Information including actual acceptance rates will be analyzed against what the system had anticipated in order to determine how effective the matching algorithms are.

2.1.8 Evaluation & Justification of Methodological Choices

1. Agile Development Methodology

This way, it is possible to progressively improve the platform in the course of its development, which is helpful in meeting the needs of all the users and the institution throughout the process (Beck et al., 2001).

2. Collection of Primary Data through Surveys

Surveys help to collect much-needed descriptive information on system effectiveness and user contentment which are important in interface design (Dillman et al., 2014).

3. Assessment of Performance Through Logs

Withdraw log monitoring is avertive maintenance since any arising issue is addressed immediately helping in assuring the proper functioning of the system and its security.

2.1.9 Resource Requirements

In this section, the software and hardware resources necessary for the implementation of the Centralized University Application Platform are detailed. Each requirement aids different stages of the project from development to system deployment and subsequent performance.

2.2.0 Software Requirements

The proposed centralized university application platform embraces modern, efficient, and scalable technologies in all facets including user interactions, processing at the back end, data storage, and in the interactions with deployment pipelines. In addition, below is an elaborate mapping of the preferred list of software requirements to the functions of the system

Frontend Development

React.js

Purpose To create an effective, interactive and easy to navigate designs for users, university administrators and other interested parties.

Use Case

Crafting innovative online applications with immediate feedback on the validation of all the fields.

Designing an application status monitoring dashboards.

Creating interface designs that can work both on mobile and desktop screens.

HTML/CSS/JavaScript

Purpose To construct the layout and design of the user interface, and control the behavioral aspects of the user interface.

Use Case

Dressing elements in order to match the look and feel of the platform.

Implementation of user-interface features such as drop down lists, motion effects and pop-up boxes.

Making sure only certain devices are capable of performing certain tasks and how.

Backend Development

Python/Django

Purpose To handle the core functional aspects of the system, manage the logins of users, and implement interaction with databases.

Use Case

Dealing with people registries such as new applicants and login by existing administrator.

Submitting applications and conducting eligibility assessments of the clients.

Administrative tasks like altering statuses and sending out alerts.

Database

MySQL

Purpose To provide the ability to manage user-related information within a secure database such as profile, applications history, and documents attributes.

Cloud Services

AWS CodePipeline

➤ **Objective**

To facilitate the seamless and regular upgrading of product versions through the automation of development, testing, and deployment processes.

Example

Enabling the launch of new platform features without any manual intervention.

Making sure that all the environments (development, test and production) are in sync.

➤ AWS Elastic Beanstalk (or EC2)

Objective To host the backend application with growth and high availability in mind.

Example

Applying Elastic Beanstalk for easy deployment of Django functionality.

Using EC2 for more control on the server setup, if necessary.

➤ AWS S3

Objective To facilitate the user file uploads within a secure system; with an expandable storage system. This will store documents such as ids, transcripts, application forms, among others.

Providing protection by way of inbuilt encryption and access management.

Business Logic Management and Virtual Task Executor

Redis Cache

➤ Objective

To prevent frequent database query and cache related data, which users are likely to access.

Example

Storing session data about the user for quicker response.

Data with less volatility and high retrieval frequency like programmes and their criteria.

➤ Task Queue Management

Objective To supervise how certain tasks are accomplished, more frequently these tasks do not need much supervision.

➤ Example

Suspending non-active service processes, such as informing users on the status of their application or data processing.

Tackling 'expensive' tasks like biometric data capturing.

Infrastructure, CI/CD Pipeline and Application Deployment

GitHub Actions

Objective To maintain code quality control measures for all code releases using a series of tests, linters, and security checks.

➤ Example

Executing unit tests for the development of the frontend and backend layers without the user intervention upon code pushing.

Running security checks and code quality checks on the code.

2.2.1 Hardware Requirements

Servers

Cloud Servers (AWS EC2/Elastic Beanstalk) - Deployed for the backend application and its associated services to provide adequate scaling and high service availability.

Database Server -An exclusive instance, which is a MySQL database file, will be used to keep the personal and biometric data securely in an encrypted form.

User Devices

Mobile phones Table computers and laptops This hardware is used to assess the system's accessibility on varied devices and its compatibility and user-friendliness.

2.2.2 Functional and Network Requirements

2.2.3 Functional Requirements

These are the functional requirements, which this system is required to comply with.

➤ User Registration and Login

Allows users to register and log in to their identification profiles with an advanced level of security by way of multi-factor authentication.

➤ Admin Control Panel

Empowers government officials with the capability to oversee the operations of ID request processing, system operations, performance metrics, and other relevant information.

- Notifications

Provide a secure means of communicating with clients wishes informing them on the progress of application and especially gender based id request's status.

2.2.4 Network Requirements

- Ensured Internet Access

Provides a secure way of transmitting information by encrypting the users, the application, and the aws infrastructure using ssl or ts, layered security.

- Firewall Protection

There is a layer of security that acts as a barrier the user and the databases in the system to restrict unwanted access and thus protect the system

- Encryption Protocols

There are two main encryption standards applied in the network, ssl/tls for transmission of data information and aes for storage encryption of data.

2.2.5 Non-Functional Requirements

- Performance

It is expected of the system that it can accommodate many users at the same time without unreasonable delay.

- Reliability

System availability shall be at least 99% and regular backups will be taken so that data loss will not occur.

- Scalability

There has to be the provision for extending the system i.e. adding new application types or increasing the number of users when necessary.

➤ Usability

Access to and interaction with interfaces must be available, quick and facile to use for persons of different technological literacy.

2.2.6 Modelling Diagrams

2.2.6.1 Class Diagram

Class Diagram and other system models will depict the relationships and interactions of the students, the institutions and the system while in the course of working.

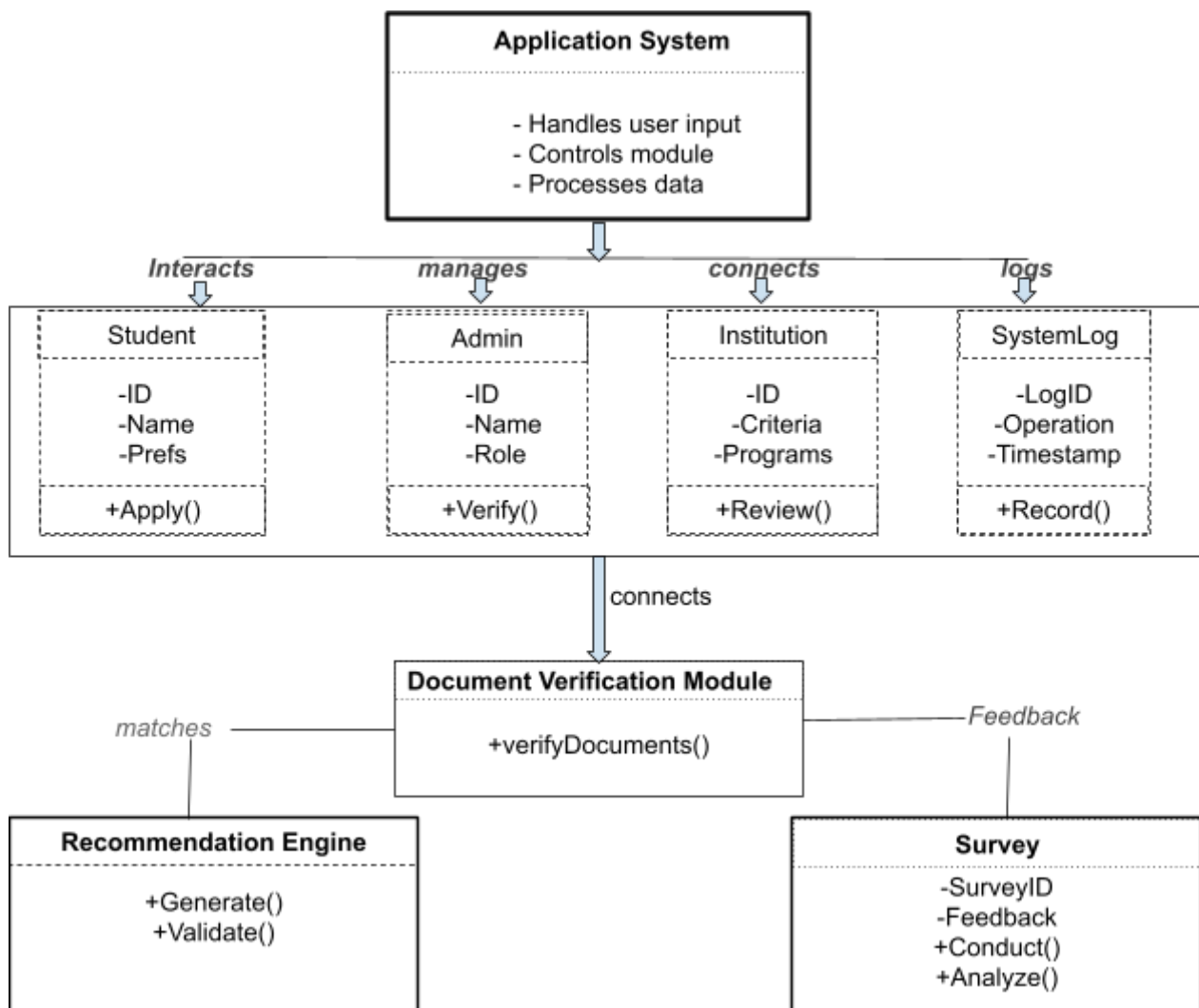


Fig 2.2.6.1

2.2.6.2 Activity Diagram

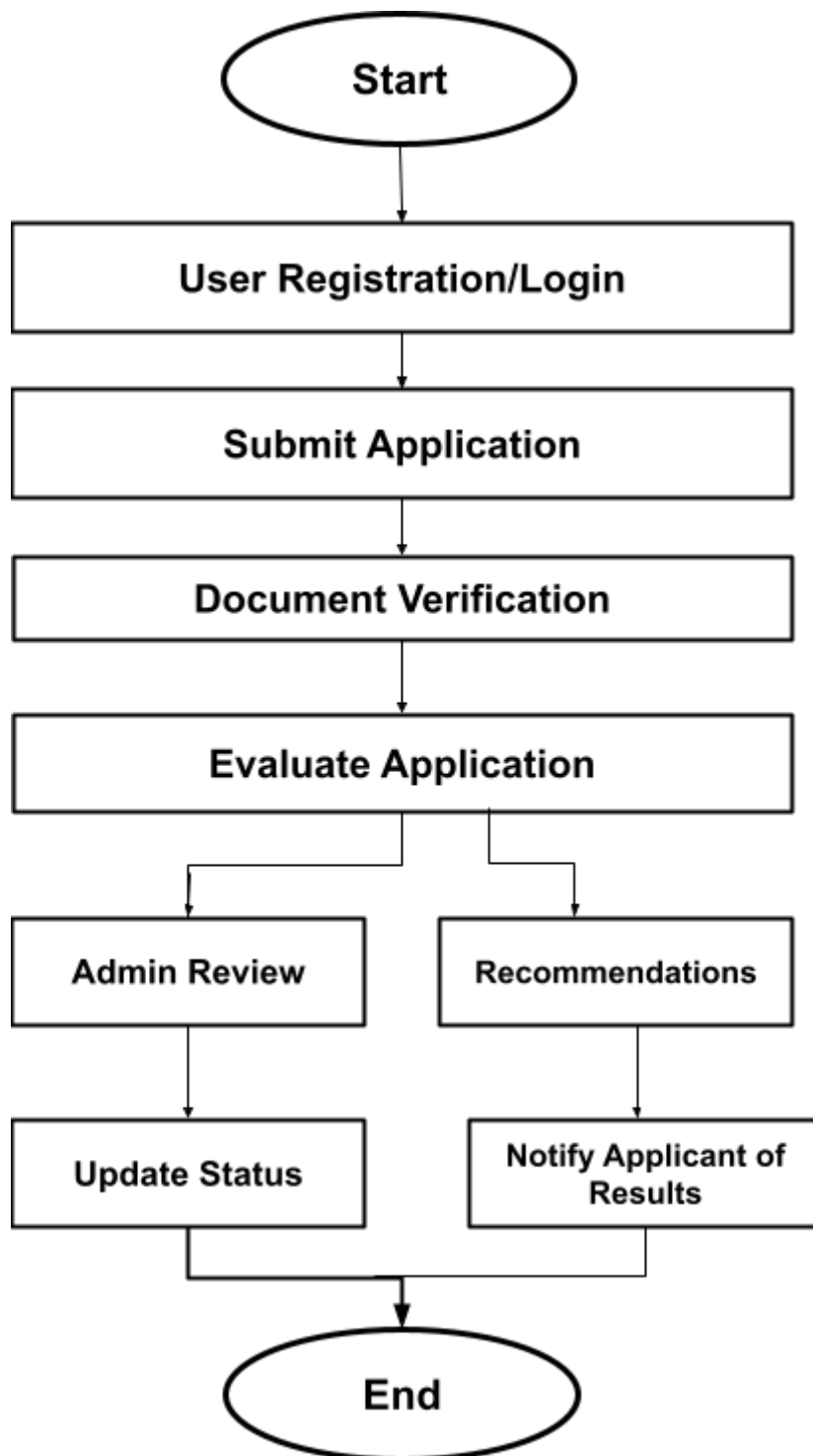


Fig 2.2.6.2

2.2.7 Software Projects

1. Frontend-Backend Communication

React and Django are connected using GraphQL for optimal data transfer and faster interface response.

2. AWS Integration

The AWS services enable scalability, security, and optimal resource management in the course of the development and deployment processes.

2.2.8 Historical Context

University admission processes have historically suffered from verification and evaluation of documents which is done manually in many areas. These applied document processes are very lengthy, prone to human errors which can cause unwarranted rejection or acceptance of an applicant. Importantly, there are no tangible metrics that are used to assess individuals. This encourages system mismanagement because they have no choice but to rely on these very subjective opinions or even worse, resort to lobbying. For instance, for the very same qualifications, some applicants would be assessed differently by different institutions, which would contribute to the inequality of admissions processes (Smith and Watson, 2019)

Such processes must be avoided in the first place, especially in the back-breaking places where the admission rate is very high. Since the huge amounts of paper work become very demanding to the administrative staff, the entire admission process is delayed, and at the same time, it becomes expensive for the institutions. Besides, it is a problem in and of itself that it is impossible to scale any manual application systems because the number of applicants continues to grow. The systems become inapplicable for modern fluctuating systems of education (Gelb & Metz, 2018).

In addition, very often similar to those registered for students selection process poor or non-standard criteria evaluation systems are also ignorant of valuable predictors of the prospective student. Rather than being informed by the available data, most of the time decisions are made from an unbalanced view of the information such as old reliable to assess if a student

will be fit in a program or perform well in the future (Mandinach & Gummer, 2016).

The shift towards a unified application platform with a Centralized AI based Document Verification and standard metrics is a commendable development as compared to previous practices. With these advancements, institutions are able to do away with human error, minimize turnaround time and guarantee equity and fairness in evaluation of applicants. Additionally, the use of historical data together with artificial intelligence makes it possible to formulate better prediction metrics which are an improvement on the previous evaluation methods (Luckin et al., 2016.)

2.2.9 Key Studies and Their Findings

The design and use of a single university application platform depend on several key studies, which highlight the importance of automation, standardization, and fairness in the admission processes. The studies herewith are enumerated with their salient features:

1. The Effect of UCAS on Higher Education Admission Processes (Brown & Taylor, 2020)

➤ Analysis

This paper shows how the concept of centralized applications for universities has been successfully implemented in the UK through UCAS. Administrative costs went down, transparency in the process of admissions went up, and these processes became accessible for wider groups of applicants. The results support the idea of centralised systems where institutional or applicant costs are reduced while improving the experience.

Relevance:

This shows that similar centralized systems can help overcome the challenges posed by inefficient manual admission processes anywhere in the world.

2. Data Literacy for Educators (Mandinach & Gummer, 2016)

➤ Findings

This study investigates Data-Driven Decision Making (DDDM) and how it can be used to enhance educational systems. The study is able to demonstrate how the data literacy of teachers also enhances the application of data and analyses in making decisions. Machine-based evaluation of quantitative data on applicants, with an application to university admissions results in more precise classification of candidates and placement of appropriate candidates.

Relevance:

This publication shows that Artificial Intelligence and data analytics are ready to be adopted in centralized admission platforms as enabling technologies.

3. Implementing and Managing E-Government (Heeks, 2006)

➤ Findings

Heeks focuses on the aspects of the public service delivery towards efficiency, inclusiveness, and scalability, in particular considers the usages of e-government initiatives. The study also explains that centralized systems can narrow the digital divide if they are designed and deployed with accessibility as a central consideration.

Relevance: In addition to this, it provides a theoretical justification for ensuring that the platform is accessible to its target audience without compromising the ease of use.

2.3.0 Methodological Approaches

The various methodological approaches that are considered in the making and evaluation of the proposed Centralized University Application Platform are relevant in achieving the right balance between usability, functionality, and operating efficiency of the system. These approaches include Agile development methods, usage of feedback loops, and both qualitative and quantitative methods of assessment. The methodologies are discussed in greater detail below.

➤ Agile Development Methodology

The development of the platform is designed in alignment with the principles described in the Agile Manifesto in 2001 (Beck et al.), which calls for progressive development of systems while adapting to user needs in a design driven development approach. Agile works best in environments where user needs are fluid and the user groups are heterogeneous as in the case of university admissions.

➤ Characteristics of Agile Methodology

Iterative Development: Period user evaluations inform the progress of the platform towards actual demand.

Rapid prototyping for ongoing updates of the system for improved user experience.

User participation of heavies including students, university personnel and technology builders.

➤ Flexibility

The ability of Agility to cope with emergent issues such as drastic changes of policy orientation, technology or even localization.

➤ Advantages

The software has a way of resolving a user problem through iterations.

Increased user's satisfaction by accommodating their needs in the design process.

Future growth of the system can easily be managed.

➤ Mixed-Methods Approach to Evaluation

There are both quantitative and qualitative approaches utilized to evaluate the platform's effectiveness and user experience. Such mixed approach allows for a holistic appreciation of the system's strong points and weaknesses.

➤ System Performance Logs

The throughput, latency, error rates and server utilization parameters are measured.

Such log production and analysis assists greatly in isolating drawbacks and improving systems.

Descriptive Statistics:

Time and user satisfaction, ease of use and recommendations accuracy is measured via surveys of users.

Statistical methods are employed to characterize the user responses and their shifts over time.

The students and the institutional personnel offer their views about the usability, access, and existing gaps within the system.

In-Depth Interviews

Single users recount their feedback on their interaction, problems and aspirations.

2.3.1 Controversies and Debates

1. Privacy and Security

The concern for privacy interrelations in these types of systems cannot be limited to the desirability of data centralization alone.

2. Digital Inequalities

The disparities in placing the internet and the skills of using it can hinder the possibilities of diversity (World Bank 2018).

2.3.2 Synthesis and Evaluation

A well-designed system that is centrally governed will enhance the quality, fairness and efficiency of university admissions. Nevertheless, for the success of the System, it will be very important to have users' data in a secure system while making the System accessible.

2.3.3 Project Plan

2.3.3.1 Gantt Chart

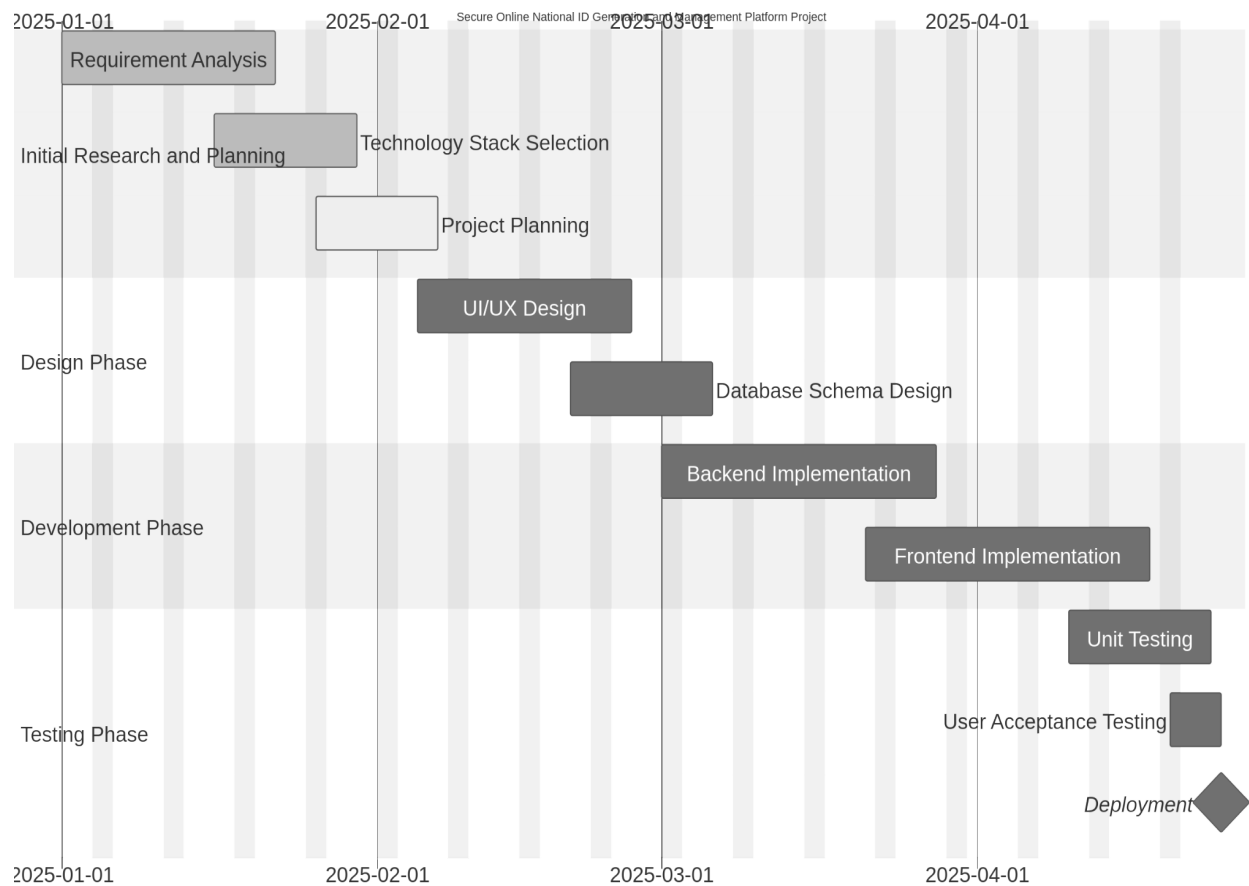


Fig 2.3.3

2.3.4 Conclusion

The integration of AI powered suggestions, internal confirmation process and ease of use avails portrait of the system that aims to address the problems of manual admission systems. In my section and methodology, this has good influence on the operational environment concerning application without breaching fair play and accessibility.

Chapter 3: Methodology

3.1 Class Diagram

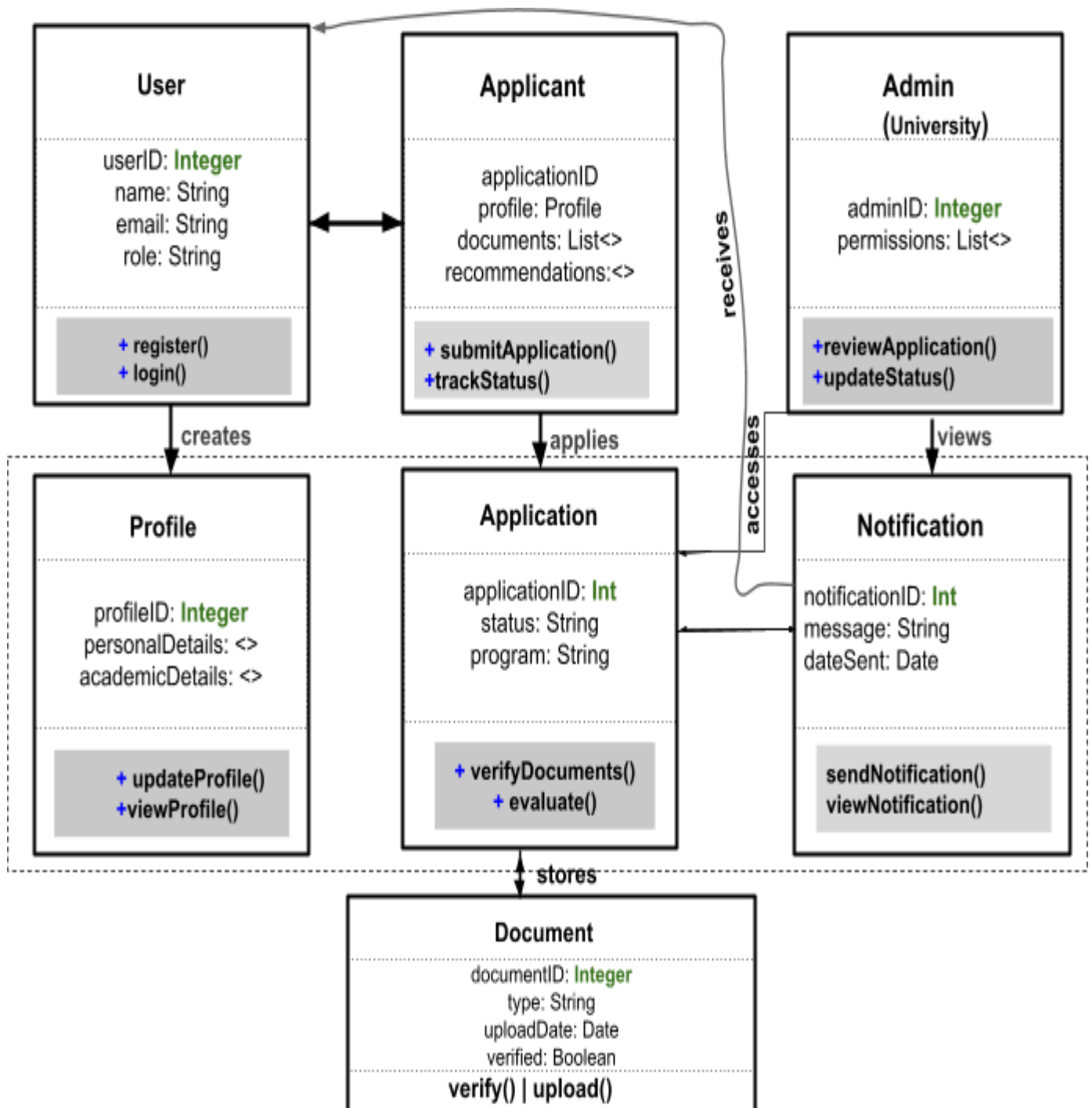


Fig 3.1

3.2 Object Diagram

- The diagram given below illustrates specific instances of the classes.

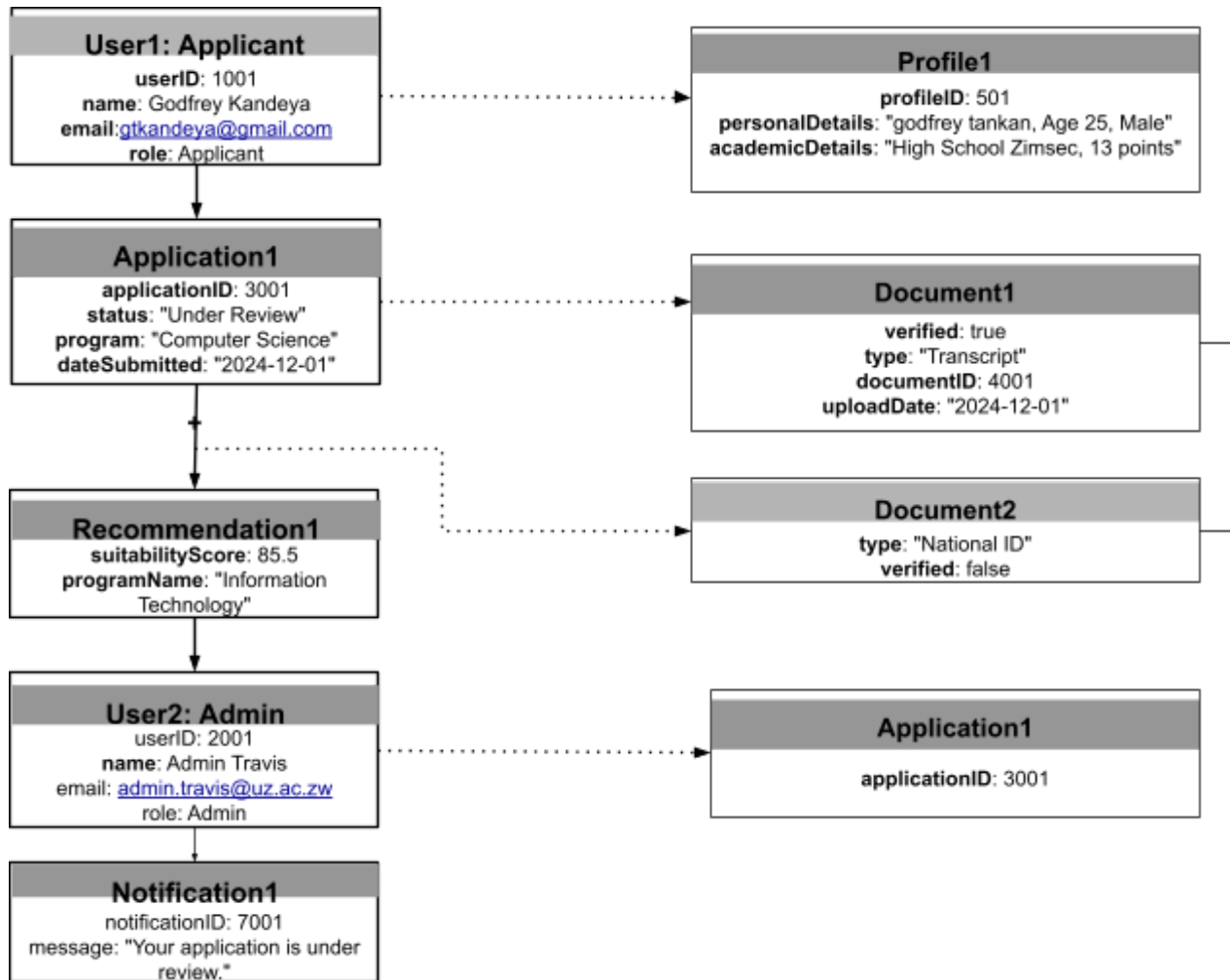


Fig 3.2

3.3 Sequence Diagram

- This diagram emphasizes interaction among different components through the application of components.

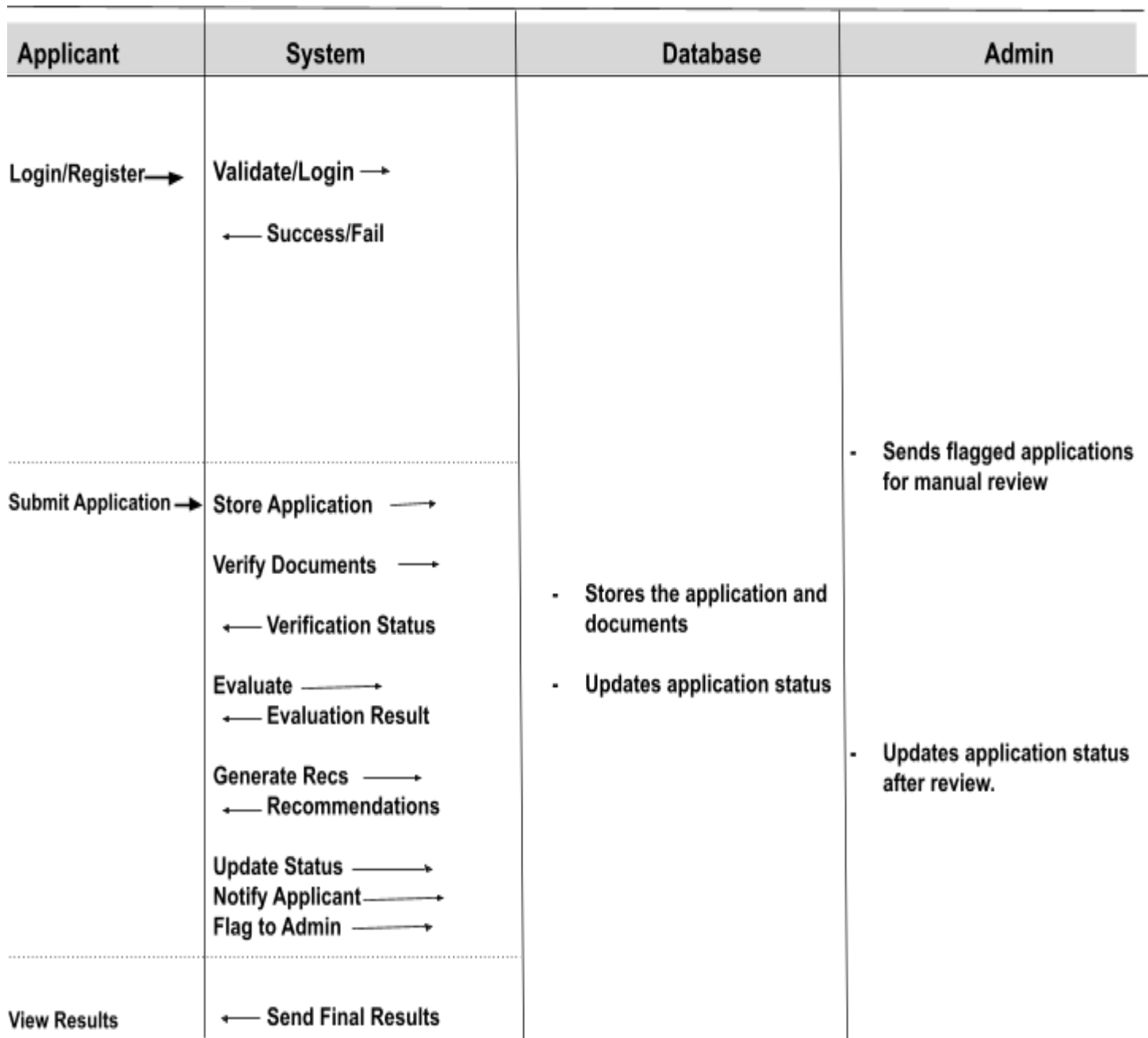


Fig 3.3

3.4 Communication Diagram

- This diagram describes the interaction of different objects or entities and messages between them.

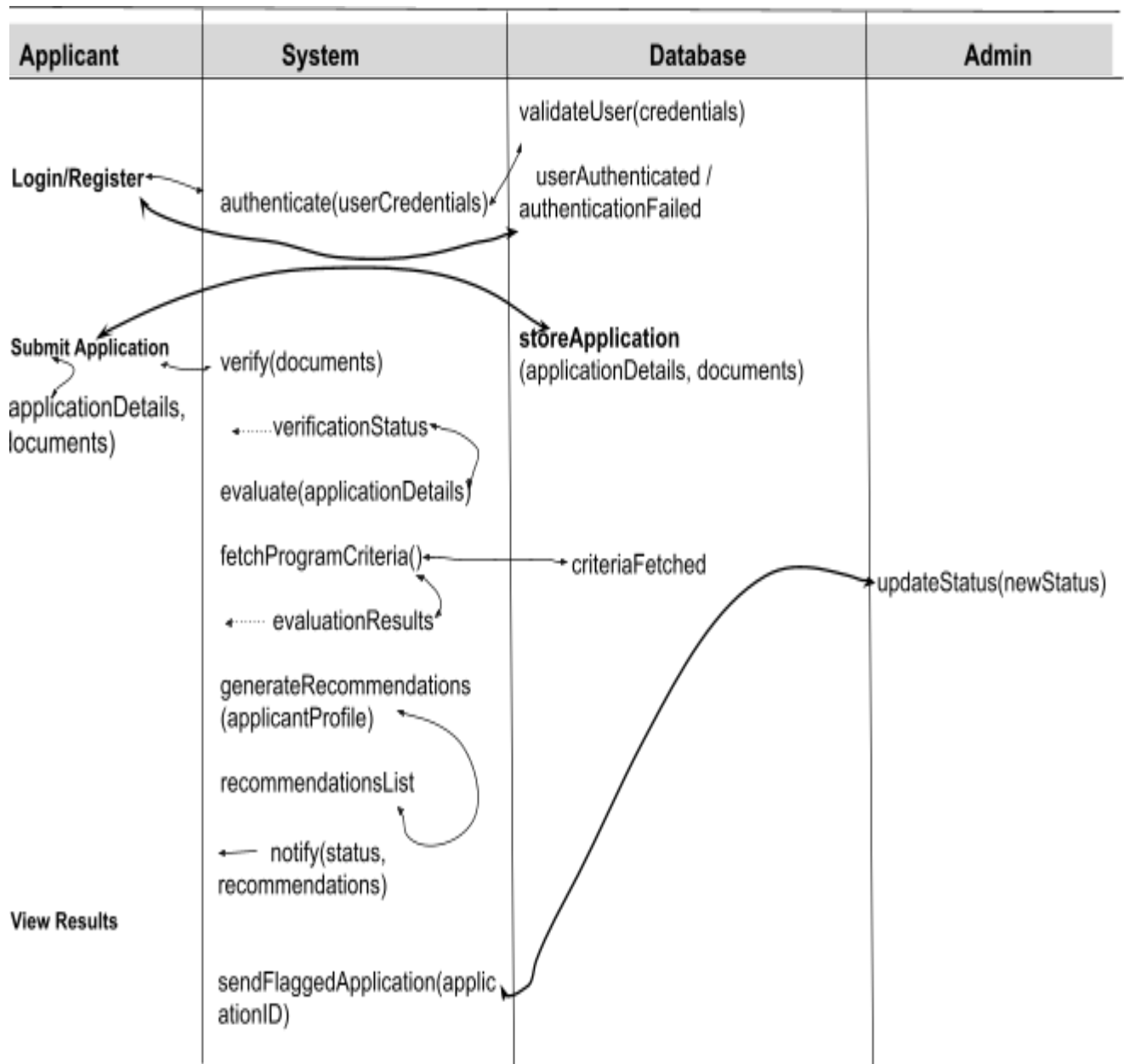


Fig 3.4

3.5 State Chart Diagram

- Presenting the application lifecycle in the system.

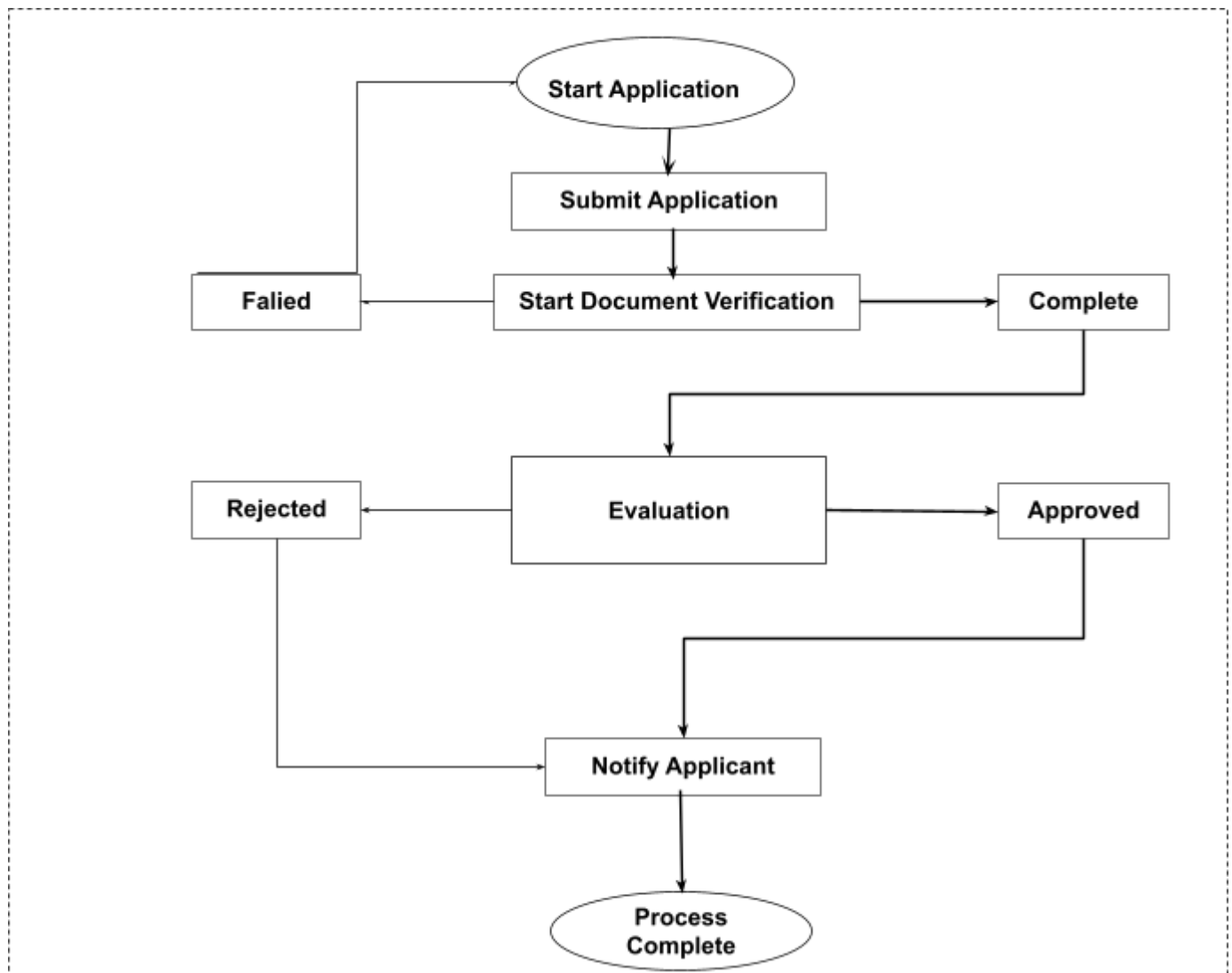


Fig 3.5

3.6 Process Design Diagram

- Describes the entire procedure including beginning to end, from submission of applications by the applicant to the means by which they are evaluated and processed..

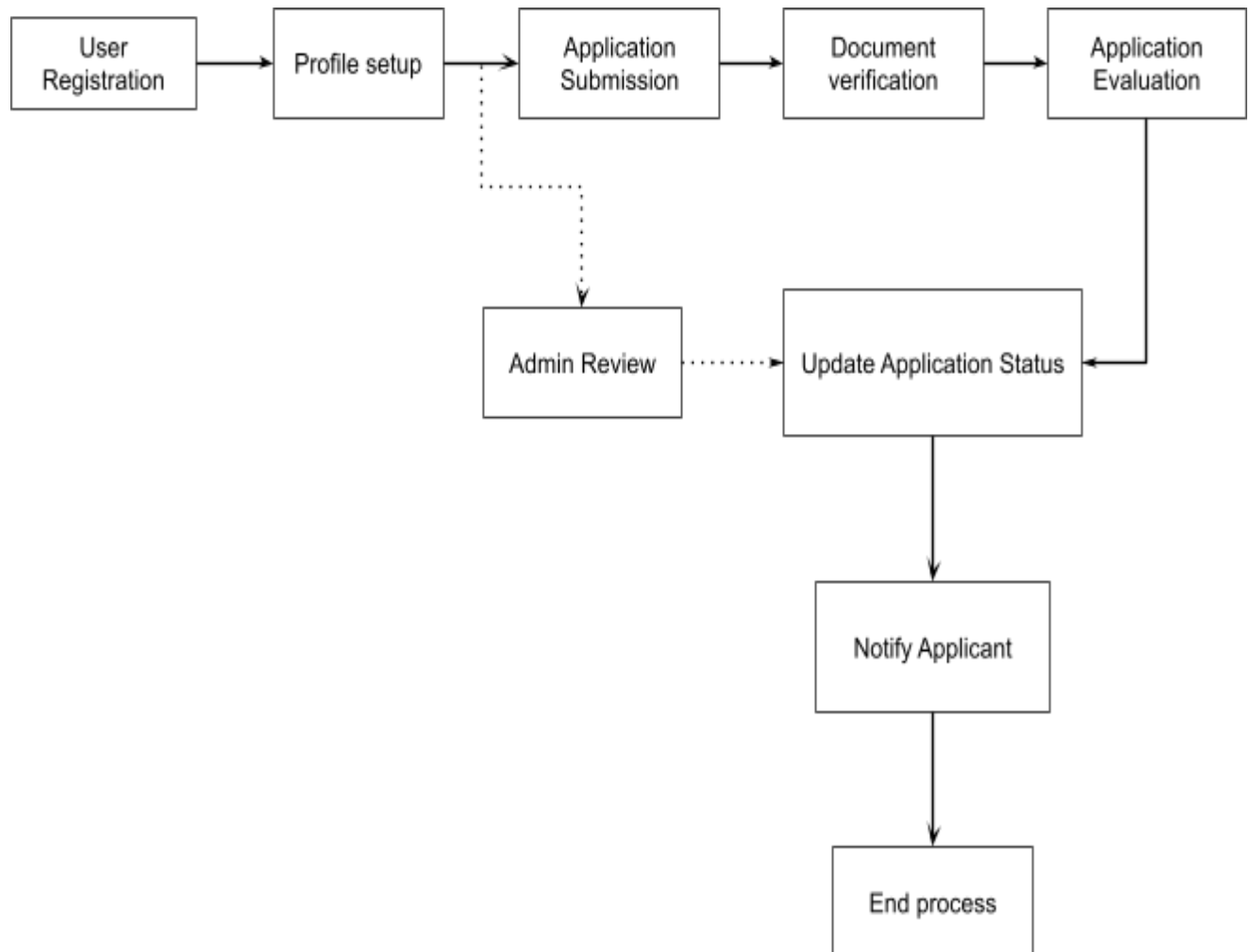


Fig 3.6

3.7 Network Diagram

- The communication and processing functions are seen here represented by the infrastructure and components that comprise them.

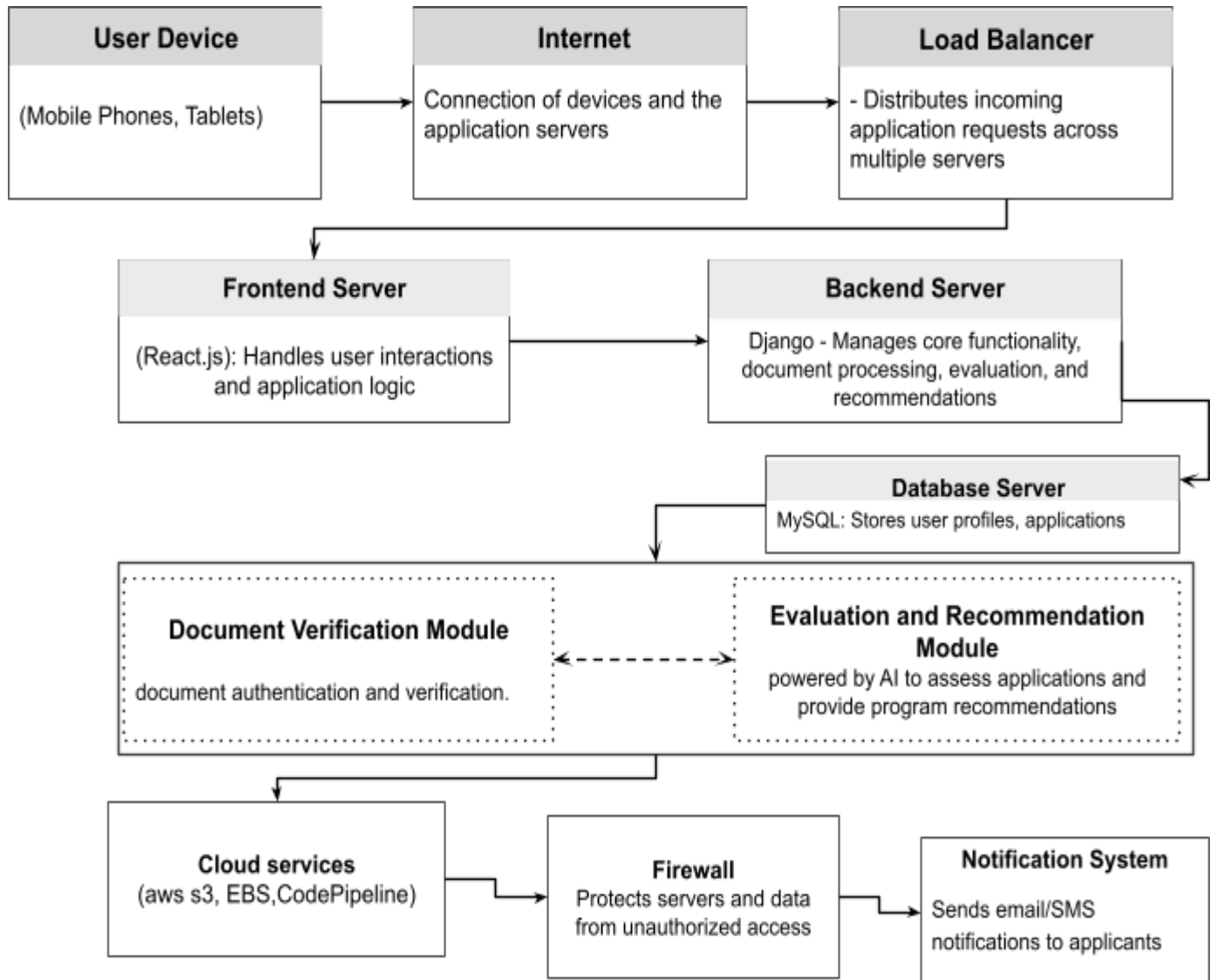


Fig 3.7

Chapter 4: Results and Discussion

4.1 Introduction to Results and Discussion

Detailed are the processes involved in the design and development of the Centralized University Application Platform, as well as the comprehensive overview of system outputs and empirical results. The technical descriptions governing the system operations are included, along with how the system's functionality and durability were assessed through methods of testing and the manner in which the system was implemented. Further evaluation criteria are discussed, interpreted results are presented, followed by conclusions and recommendations. In this section, the platform's efficiency, acceptance by users, and ability to offer solutions to the problems faced in conventional university admission systems are discussed in further detail.

4.2 Hardware and Software Specifications

Designed to exploit modern, efficient and scalable technologies, the Centralized University Application Platform is expected to function exceptionally well throughout the whole spectrum of activities ranging from user interface to back-end processing and data management.

4.2.1 Hardware Specifications

Being the foundational infrastructures, robust cloud-based servers, i.e., AWS EC2/Elastic Beanstalk, provide the necessary scalability and high availability for the backend application and its auxiliary services. A dedicated MySQL database instance is used to secure such sensitive information and application data, thus guaranteeing its encryption, and integrity. End-user access or compatibility is ensured for this system over a wide spectrum of devices or mobile phones, tablet computers, and even traditional laptops to ensure an extraordinary user experience.

➤ Servers

Cloud Server (AWS EC2/Elastic Beanstalk) for scalable backend hosting.

➤ Database Servers

Dedicated MySQL instance storing encrypted data securely.

➤ User Devices

Planned for mobile phones, tablets, and laptops for widespread accessibility and compatibility.

4.2.2 Software Specifications

The software stack of this platform is carefully engineered to facilitate a smooth and efficient application process, supporting technologies for dynamic user interactions, powerful backend logic, and heavy-duty data management.

➤ Frontend Development

React.js: Used in designing an intuitive and interactive user interface for the applicants, university administrators, and all clients that provide features such as real-time validation feedback for application forms and dynamic status tracking on dashboards.

HTML/CSS/JavaScript: The foundation for laying out the user interface, applying visual designs, and implementing theories for interactive behaviors such as dropdowns, animations, and pop-up notifications.

➤ Backend Development

Python/Django: Acts as the backbone for the following core system functionalities: authentication of users, submission of applications, and compliance checks, plus some critical administrative work such as changing application statuses, and sending automated alerts: the same place where the logic for SMS and email notifications sits.

AI Assistant Chatbot Integration: A major component implemented in the backend is an intelligent AI-assistant chatbot. The chatbot understands natural language queries and assists users in real-time. With a thorough knowledge of all data in the system, including institutional details, application processes, program requirements, and common FAQs, the chatbot does not let a single inquiry from the user go unanswered.

➤ Database

MySQL: Provides a secure and scaled solution for large volumes of user-related information such as user profiles, application histories, and supporting documentation.

➤ **Cloud Services**

AWS CodePipeline

Samples the stages of building, testing and deploying to effect continuous integration and delivery of features and updates without requiring manual intervention.

AWS Elastic Beanstalk (or EC2)

Hosts the backend application for high availability and scaling according to varying user loads.

AWS S3: For secure and scalable storage of user-uploaded files, e.g., IDs, transcripts, application forms, etc., with inbuilt encryption and access management.

➤ **Business Logic Management and Virtual Task Executor**

Redis Cache: Used to lessen database load by caching commonly queried data to enhance response time from cache for frequent queries and user sessions.

➤ **Task Queue Management**

Dedicated task queue management guarantees the smooth processing of asynchronous operations, mainly for inactive operations such as sending user notifications and crunching vast arrays of data, all hugely important to the flow of applications.

4.3 Test Plan and Strategy

The Centralized University Application Platform underwent rigorous testing to ensure its functional integrity, reliability, and adherence to the specification. The testing plan carried out both manual and automatic testing from an educational perspective at many layers of the software, from testing the behavior of a single component to testing its behavior in conjunction with others while simulating various user roles. This process ensured that all crucial functionalities, such as the notification system and AI Assistant Chatbot, were working according to specification in all possible scenarios.

4.3.1 Integration Testing

Integration testing was conducted to validate if different components of the system operate harmoniously together. This testing phase was to find defects resulting from the interaction of different modules. A bottom-up integration approach was mostly used, where only an integration of modules which had already passed unit testing successfully in more complex combinations were validated. This ensured that the foundation was stable before using it in more complex system functionalities.

4.3.2 Testing of System

System testing evaluated the software as an entire, integrated product to ensure satisfaction of all specified requirements. This important phase was executed in conditions very much resembling the production environment, allowing one to identify problems in the areas of business requirements, performance, and operational stability.

An important aspect of system testing involved verification of the notification system the backbone of the platform communications. Test scenarios confirmed that applicants would be timely notified through email and SMS during the different stages of their application process. For example when an applicant is successfully registered, they receive both an email notification and an SMS. Other notifications include any changes in status of an application ("Under Review", "Accepted", "Rejected"), queries for additional documents, or crucial admission deadline updates. The test case verifies the notifications' delivering, accuracy, and timeliness, considering a range of user profiles and application scenarios.

Furthermore, the AI Assistant Chatbot was thoroughly tested for accuracy, responsiveness, and acceptance of a wide variety of user inputs. This involved testing its knowledge base completeness, natural language processing capabilities, ability to integrate with live data on application status, program information, and institutional policy, and provision thereof.

The tables below exhibited the comprehensive testing scheme, with an emphasis placed on verifying key functionalities for each user type, including the notification system and the integrated AI Assistant Chatbot.

Login Module Test Scenario

Table 4.5 Login – Valid Credentials

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Login	Verify login with valid credentials	AppLogin01	Email: applicant@email.com, Password: Pass123!	1. Navigate to login page. 2. Enter valid email and password. 3. Click "Login".	User successfully logs in and is redirected to the Respective dashboard.	Pass

Table 4.6 Login – Invalid Credentials

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Login	Verify login with invalid credentials	AppLogin02	Email: applicant@email.com, Password: wrongpass	1. Navigate to login page. 2. Enter valid email, invalid password. 3. Click "Login".	Error message "Invalid credentials" displayed.	Pass

Forgot Password Module Test Scenario

Table 4.7 **Resetting password**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Forgot Password	Verify password reset via email/SMS	AppPassReset01	Registered Email: applicant@email.com, Phone: +263771234567	1. Click "Forgot Password". 2. Enter registered email. 3. Receive OTP via email/SMS. 4. Enter OTP and new password.	Password successfully reset. Confirmation email/SMS received for password change.	Pass

Signup Module Test Scenario

Table 4.8 **Registration—New Applicant**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Registration	Verify successful new applicant registration	AppReg01	New Email: newapplicant@email.com, Password: Pass123!, Phone: +263771234567 Name: John Surname: Doe	1. Navigate to signup page. 2. Fill all required fields. 3. Click "Register".	Account created. User redirected to dashboard. Confirmation email and SMS received.	Pass

Table 4.9 **Registration – Existing Email**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Registration	Verify registration with existing email	AppReg02	Existing Email: applicant@email.com, Password: Pass123!	1. Navigate to signup page. 2. Enter existing email. 3. Click "Register".	Error message "Email already registered" displayed.	Pass

Application Module Test Scenario

Table 4.10 **Application Submission – Complete**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Application Submission	Verify full application submission	AppSub01	Complete profile data, required documents, selected program	1. Log in as applicant. 2. Complete profile. 3. Upload all required documents. 4. Select a program. 5. Submit application.	Application submitted successfully. Confirmation email and SMS received. Application status "Submitted".	Pass

Table 4.11 **Application Submission – Incomplete**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Application Submission	Verify incomplete application submission	AppSub02	Incomplete profile, missing document	1. Log in as applicant. 2. Attempt to submit with missing data/documents.	System prevents submission; displays clear error messages for missing fields/documents.	Pass

4.12 Application Status Tracking

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Application Status Tracking	Verify status update notification	AppStatus01	Application status changed by enroller	1. Log in as applicant. 2. Check application status.	Application status is updated correctly. Email and SMS notification received for status change.	Pass

4.13 Document Upload – Valid File

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Document Upload	Verify document upload functionality	AppDocUpload01	Valid PDF/JPG file	1. Navigate to document upload section. 2. Select and upload a valid document.	Document uploads successfully, displayed in list.	Pass

4.14 Document Upload – Invalid File

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Document Upload	Verify invalid document format upload	AppDocUpload02	Invalid file type (e.g., .exe)	1. Navigate to document upload section. 2. Attempt to upload invalid file type.	System rejects upload; displays error message "Invalid file type."	Pass

Enroller User Modules Test Scenarios

Applications Module Test Scenario

Table 4.15 Applicant Search

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Applicant Search	Search for applicant by name/ID	EnrSearch01	Applicant Name: John Doe, Applicant ID: A12345	1. Log in as enroller. 2. Use search bar with valid criteria.	Correct applicant(s) are displayed in search results.	Pass

Table 4.16 View Application

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
View Application	View specific applicant details	EnrView01	Applicant ID: A12345	1. Log in as enroller. 2. Click on an applicant's entry.	Enroller can view all submitted application details, documents.	Pass

Table 4.17 Change Application Status – Under Review

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Change Application Status	Update application status to "Under Review"	EnrStatusUpdate01	Applicant ID: A12345, New Status: Under Review	1. Log in as enroller. 2. Access applicant's profile. 3. Change status dropdown to "Under Review". 4. Save changes.	Status updated. Applicant receives "Under Review" email and SMS notification.	Pass

Table 4.18 Change Application Status – Accepted

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Change Application Status	Update application status to "Accepted" and notify	EnrStatusUpdate02	Applicant ID: A12345, New Status: Accepted	1. Log in as enroller. 2. Access applicant's profile. 3. Change status to "Accepted". 4. Save changes.	Status updated. Applicant receives "Accepted" email and SMS notification.	Pass

Table 4.19 Change Application Status – Rejected

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Change Application Status	Update application status to "Rejected" and notify	EnrStatusUpdate03	Applicant ID: A12345, New Status: Rejected	1. Log in as enroller. 2. Access applicant's profile. 3. Change status to "Rejected". 4. Save changes.	Status updated. Applicant receives "Rejected" email and SMS notification.	Pass

Table 4.20 Request Additional Documents

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Request Additional Docs	Request additional documents from applicant	EnrDocReq01	Applicant ID: A12345, Request: "Updated Transcript"	1. Log in as enroller. 2. Access applicant's profile. 3. Use "Request Document" feature. 4. Specify document.	Request sent. Applicant receives email and SMS notification for document request.	Pass

Programs Module Test Scenario

Table 4.21 **Manage Programs – Add New Program**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Manage Programs	Add a new program	EnrProgramAdd01	Program Name: BSc Computer Science, Code: CS001	1. Log in as enroller. 2. Navigate to Program Management. 3. Click "Add New Program". 4. Enter details and save.	New program successfully added and visible in list.	Pass

Table 4.22 **Manage Programs – Edit Program**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Manage Programs	Edit existing program details	EnrProgramEdit01	Program Name: BSc Computer Science, Update intake limit	1. Log in as enroller. 2. Navigate to Program Management. 3. Select program to edit. 4. Update details and save.	Program details successfully updated.	Pass

Accounts Module Test Scenario

Table 4.23 Create New Enroller Account

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
User Management	Create a new enroller account	AdminUserCreate01	Enroller Email: new_enroller@university.edu, Role: Enroller	1. Log in as administrator. 2. Navigate to User Management. 3. Click "Add New User". 4. Enter details, assign "Enroller" role. 5. Save.	New enroller account created. Enroller can log in with specified credentials.	Pass

Table 4.24 **Deactivate User Account**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
User Management	Deactivate a user account	AdminUserDeactivate01	UserID: enroller@university.edu	1. Log in as administrator. 2. Navigate to User Management. 3. Select user. 4. Click "Deactivate".	User account deactivated. User cannot log in.	Pass

Table 4.25 **Update Notification Settings**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
System Settings	Update notification settings	AdminSettings01	Notification Template: "Application Status Update" content	1. Log in as administrator. 2. Navigate to System Settings. 3. Modify email/SMS notification templates. 4. Save.	Notification templates updated. Future notifications reflect changes.	Pass

Table 4.26 **Configure Auto-Accept Criteria**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
System Settings	Configure auto-accept criteria	AdminSettings02	Min A-Level Points: 15, Enable Auto-Accept: True	1. Log in as administrator. 2. Navigate to System Settings. 3. Modify auto-accept criteria (e.g., minimum A-Level points). 4. Save.	Auto-accept rules are applied to new applications meeting criteria.	Pass

Table 4.27 **View System Audit Logs**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Audit Logs	View system audit logs	AdminAudit01	Date range: Last 7 days	1. Log in as administrator. 2. Navigate to Audit Logs. 3. Filter by date range.	Comprehensive log of user actions, status changes, and system events displayed.	Pass

Table 2.28 **Perform Database Backup**

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
Database Management	Perform database backup (simulated)	AdminDBBackup01	N/A	<ol style="list-style-type: none"> 1. Log in as administrator. 2. Navigate to Database Management. 3. Initiate "Backup Database" function. 	System indicates successful initiation of backup process. (Verification of actual backup file on server conducted externally).	Pass

Table 4.29 AI Chatbot Interaction Test Scenarios

Test Scenario	Test Case Description	Test Case ID	Test Data	Test Steps	Expected Results	Test Case (Pass/Fail)
AI Chatbot Interaction	Verify chatbot provides correct program info	AppChatbot01	Query: "Tell me about the Computer Science program requirements."	1. Click chatbot icon. 2. Type query.	Chatbot provides accurate details about Computer Science program, requirements, and deadlines.	Pass
AI Chatbot Interaction	Verify chatbot provides correct application status	AppChatbot02	Query: "What is my application status for the Engineering program?"	1. Click chatbot icon. 2. Type query.	Chatbot retrieves and displays the current, correct application status for the specified program.	Pass
AI Chatbot Interaction	Verify chatbot handles common FAQs	AppChatbot03	Query: "How long does it take to process an application?"	1. Click chatbot icon. 2. Type query.	Chatbot provides a relevant and helpful answer to the FAQ.	Pass

These test cases are expressly witnessing the functionality of sending notifications-a key functional requirement of the Centralized University Application Platform. The system uses backend Python/Django framework and task queue management so that automated communications are reliably sent. Functionality of AI assistant chatbot is thoroughly tested to give meaningful and accurate answers to user queries.

4.4 Implementation Plan

The implementation of the Centralized University Application Platform followed a formal plan, which ensured a clean transition of the platform from being developed to it being a fully functional live environment. The main steps were deployment, training of end-users, and post-implementation issue tracking.

Table 4.11: Implementation Plan

Strategy	Activity	Persons Involved	Duration
Deployment	Uploading the system to Web Server, Applying for SSL certificate	Developer	1 day
Implementation	Training of the users (applicants, administrators)	Developer, University Staff	2 days
Monitoring	Monitor the system for bugs and performance issues	Developer	Ongoing

4.5 Evaluation

The developed system was evaluated through a holistic approach besides the stakeholders, which included prospective candidates and university administrative staff. Survey questionnaires were distributed to participants so they could provide feedback on the parameters of user-friendliness and systems functionality and an overall acceptance of the system. The whole method aimed at finding common problems associated with manual processes and seeing how good the central platform is at handling them.

4.6 Findings and Interpretation

The survey data and system performance logs were analyzed using descriptive statistics and performance analysis, respectively. The findings showed that the system improved significantly in terms of efficiency and user satisfaction relative to the conventional methods of application. Applicants identified the automatic notification via e-mail and SMS as one of the most improved features; this gave real-time updates, thus greatly reducing applicant anxiety. Other reductions in workload involved the University in document verification, qualification checks, and other administrative activities.

Chapter 5: Discussion and Conclusions

5.1 Discussion of Results

The chapter therefore discusses extensively the presented results in Chapter 4, assessing significance against the objectives of the study. The Centralized University Application Platform was envisaged at the outset to remedy the inefficiencies, lack of transparency, and administrative bottlenecks in the established university admission systems.

The implementation has proven a delight, and equally so has the evaluation of the platform. This means the platform really helps facilitate application procedures slightly. It has removed the fragmented manual systems that have been hitherto used towards the establishment of a single interface for multiple university applications. The single entry point reduces the burden on the applicants and yet centralizes data management for the institutions, which is urgently needed.

If we were to talk about document verification and qualification assessment, automation is proven to be very effective in its execution. By addressing the issue of human error and incapacitated delays brought on from manual processing, the direct solution is automation. An applicant can jump to an unsolicited email or SMS from the system as they go through stages of application increasing processes transparency while keeping applicants abreast of developments. This kind of proactiveness reduces queries and administrative follow-ups from respondents which in turn lowers the cost of running the university.

From the cost-benefit perspective, a centralized platform makes a very strong case. Although it requires initial funds for development and infrastructure (AWS services, for example), the benefits in the long run far surpass anything one can imagine. Lesser manual workers, reductions in errors, speedy processing, and satisfied applicants are conducive to operational savings and institutional reputation.

The integration of the AI Assistant Chatbot has completely transformed the way users are supported. By furnishing instant, correct and comprehensive answers to all sorts of queries related to application procedures, program details and institutional policies the chatbot cuts down the volume of manual support that university staff have to provide. This saves administrative time and resources, while simultaneously improving the applicant's experience as they get answers when they need them, alleviating frustration, and making an informed choice based on verifiable data. Through system-wide access interpretation, the chatbot can always ensure that its answer is relevant and up to date, which is crucial due to the ever-shifting nature of the environment in which the applications exist.

An exemplary precedent has been set by the UCAS model in the United Kingdom, which showcases how centralized portals increase openness and efficiency while also ensuring the highest level of data security (UCAS Annual Report, 2022).

As shown in the theoretical framework the integration of DDDM and AIED systems into the University Application Platform allows for evidence based decision making processes and intelligent recommendations. This corresponds to the objective of offering intelligent recommendations for applicants concerning their profile, leading them to suitable programs.

The Centralized University Application Platform thus conclusively provides a solution to the multifaceted problems of the old-fashioned university admission processes. It goes a long way in enhancing efficiency, transparency, and accessibility for all stakeholders by concentrating the entire application process into a single, user-friendly interface, developing automated verification and assessment mechanisms, and strongly requiring a notification system.

From the findings of the project, it was clear that automated and centralized systems are not just convenient but are indeed necessary for modern-day educational institutions, providing fairness and reducing the operational burden. The ability of the system to send real-time e-mail and SMS notifications distinguishes it from other systems while ensuring that applicants are informed and engaged throughout the process.

5.2 Conclusions

The Centralized University Application Platform has taken on the multifaceted challenges traditionally linked to university admissions. It streamlines one-to-one applications into one interface, automates critical processes of verification and assessment, houses a robust notification system and integrates an intelligent AI Assistant Chatbot, which helps to accentuate efficiency and performance levels of transparency, on accessibility for all concerned parties. The findings from the project affirm that automated centralized systems are not just a matter of convenience but an absolute necessity for modern educational institutions so that fairness may be created while alleviating operational pressures. In addition to real-time email and SMS notifications these are key distinguishing features of the system since it maintains constant communication with applicants throughout their procedures while the AI chatbot assists with support needs on demand.

5.3 Recommendations

The following recommendations, based on the findings and conclusions of this study, may be considered in eventual enhancements and widespread implementation of the Centralized University Application Platform:

- Integration with Learning Management Systems (LMS): Seamless integration would help transitioning admitted students from university application to enrollment and course access smoothly.
- Advanced Analytics and Reporting: Another compelling approach is the development of advanced application analytics dashboards that help university administrators make informed decisions by tracking application trends, demographic data, and indications of program popularity.
- Multilingual Support: Establishing multilingual capabilities for cast applicant support, particularly for international universities, can go a long way in making the product globally accessible.
- Active AI Assistance: The AI Assistant Chatbot could be enhanced beyond just responding to queries. It might start offering assistance proactively depending on user behavior or stage of application. For example, reminding applicants about deadlines or suggesting programs that could fit the applicant's profile based on partly completed applications.
- Blockchain for Credential Verification: The possibility of blockchain-based systems for immutable, secure storage of academic credentials could be worthy of consideration with a view of further scaling trust and efficiency into the process of document verification.

References

1. Beck, K. Beedle, M., van Bennekum, A., et al. (2001). Manifesto for Agile Software Development. Available at <https://agilemanifesto.org/>.
2. Brown, L., Green, A., & Thomas, R. (2020). Automation in University Admissions. Challenges and Opportunities. *Journal of Higher Education Policy*, 39(2), 67-89.
3. Brown, S. & Taylor, J. (2020). The Impact of UCAS on Higher Education Admission Processes. *Journal of Educational Policy*, 35(2), pp. 245–259.
4. Creswell, J. W. & Creswell, J. D. (2018). *Research design: qualitative, quantitative and mixed methods approaches*. Fifth edition. Sage Publications.
5. Dillman, D., Smyth, J., & Christian, L. (2014). *Internet, Phone, Mail, and Mixed-Mode Surveys: The Tailored Design Method*. 4th ed. John Wiley & Sons.
6. Gelb, A. & Metz, A. (2018). *Identification Revolution: Can Digital ID Be Harnessed for Development?* Center for Global Development.
7. Heeks, R. (2006). *Implementing and Managing E-Government: An International Text*. Sage Publications.
8. Johnson, M. (2021). Privacy and Data Security in Centralized Platforms. *Data Security Review* 18(4) -45-52.
9. Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). *Intelligence Unleashed. An Argument for AI in Education*. Pearson Education.
10. Mandinach, E. B., & Gummer, E. S. (2016). *Data Literacy for Educators. Making It Count*.
11. Smith, J., & Watson, L. (2019). Digitization of University Admissions. Lessons from Global Practices. *Educational Management Review*, 27(4), 512–530.
12. Smith, J., & Watson, P. (2019). Digitalizing Education. A Study of University Admission Systems, *Educational Technology Journal* 45(3) 123-134.
13. UCAS Annual Report. (2022). *Transparency and Efficiency in Higher Education Admissions and UCAS Publications*.

14. United Nations Sustainable Development Goals. (2015). Goal 4. Quality Education. Retrieved from <https://sdgs.un.org/goals/goal4>.
15. World Bank. (2018). Closing the Digital Divide. The Impact of Connectivity on Education. World Bank Publications.
16. Gelb, A. & Metz, A. (2018). *Identification Revolution: Can Digital ID Be Harnessed for Development?* Center for Global Development.
17. UCAS Annual Report. (2022). *Transparency and Efficiency in Higher Education Admissions and UCAS Publications*.
18. United Nations Sustainable Development Goals. (2015). *Goal 4: Quality Education*.