IMPLEMENTATION OF A FINANCIAL MANAGEMENT SYSTEM FOR A WELFARE SCHEME

BY

IDUWEH TREASURE EKPEMI

(PSC2010399)

THE DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF PHYSICAL SCIENCE

UNIVERSITY OF BENIN

BENIN CITY, EDO STATE.

FEBRUARY, 2025

IMPLEMENTATION OF A FINANCIAL SYSTEM FOR A WELFARE SCHEME

BY

IDUWEH TREASURE EKPEMI

(PSC2010399)

A PROJECT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF PHYSICAL SCIENCES, UNIVERSITY OF BENIN, BENIN CITY, EDO STATE NIGERIA. IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF BACHELOR OF SCIENCE(B.Sc.) DEGREE IN COMPUTER SCIENCE.

FEBRUARY, 2025

CERTIFICATION

This is to certify that this project work was carried out by IDUWEH TREASURE EKPEMI with Matriculation Number PSC2010399 under my supervision and is adequate in scope and content for the award of Bachelor of Science (B.Sc.) DEGREE in Computer Science of the University of Benin.

_____ _____

Mr .S.O.P. OLIOMOGBE DATE

PROJECT SUPERVISOR

APPROVAL

This project is hereby approved by the department of Computer Science in partial fulfillment of the requirements for the award of Bachelor of Science (B.Sc.) degree in Computer Science of University of Benin.

_____ _____

PROF GOD'SPOWER .O. EKUOBASE DATE

HEAD OF DEPARTMENT

DEDICATION

I dedicate this work to God for making this journey possible. This work is also dedicated to my Mother, Mrs Iduweh Comfort Odufa (Ibatuemoli), for her guidance and love throughout this Journey.

And to my Project supervisor for his patient guidance.

ACKNOWLEDGEMENT

My profound gratitude goes to the almighty God who granted me wisdom and grace to complete this course.

My utmost thanks to my Mother, **Mrs Comfort Odufa** for being my source of inspiration and also my sisters, Sis Osas, Sis Debbie and Favour. Pst MICHAEL OLUMESE, MR ABIODUN OSARETIN IGORI, MR & MRS AGBADUA, Thank you all for your financial, moral and emotional support throughout my years in the University of Benin.

I also wish to acknowledge my project supervisor **MR (S.O.P.) OLIOMOGBE** for his keen supervision and direction. I am immensely grateful.

Special gratitude goes to the Head of the DEPARTMENT **PROF. GODSPOWER .0. EKUOBASE.** And all other lectures and staff of the Department of Computer Science for their impact in one way or the other

To all my friends and family, I say thank you for your support.

ATTESTATION

I, **IDUWEH TREASURE EKPEMI** with Matriculation Number **PSC2010399** attest that this project work titled **"IMPLEMENTATION OF A FINANCIAL MANAGEMENT SYSTEM FOR A WELFARE SCHEME"** is an original work and was carried out by me. And it has not been used elsewhere for any degree or any other certificate.

_____ _____

IDUWEH TREASURE EKPEMI **DATE**

PROJECT STUDENT

TABLE OF CONTENT

COVER

PAGE……i

TITLE PAGE…….ii

CERTIFICATION……iii

APPROVAL…….iv

DEDICATION…….v

ACKNOWLEDGEMENT……..vi

ATTESTATION…….vii

ABSTRACT……..x

CHAPTER ONE

INTRODUCTION

1.0 Background Study…… 1

1.1 Statement of problem……... 2

1.2 Aim and Objectives of the Study…….2

1.3 Significance of Study……...4

1.4 Motivation ……5

1.5 Scope of the Study……5

1.6 Limitations of the Study……...7

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction ……9

2.1 Financial Management Systems ……...12

2.2 Welfare Scheme Operations ……15

2.3 Technology Adoption in Financial Systems ……20

2.4 Existing Systems in Auchi Polytechnic or Similar Institutions ……25

2.5 Summary of Literature Review ……..30

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.1 Introduction ……32

3.2 System Overview ……32

3.3 Functional and Non-Functional Requirements ……...…33

3.3.1 Functional Requirements ……34

3.3.2 Functional Requirements ……....35

3.4 System Architecture ……..….36

3.5 Database Design ……37

3.5.1 Entity-Relationship Diagram …….....37

3.5.2 Database Schema ……39

3.6 User Interaction Flow ……..…41

3.6.1 User registration and Login ……41

3.6.2 Contribution Management …….…….43.

3.6.3 View Contribution History …….....44

3.6.4 Loan Application and Approval ……46

3.6.5 Loan Repayment ……....47

3.7 Security Considerations……..49

3.8 Technology Stack Justification …….50

3.9 Conclusion ……51

CHAPTER FOUR

IMPLEMENTATION

4.1 Introduction ……52

4.2 Development Tools and Technologies Used …….52

4.3 System Modules and Features ……...55

4.4 Backend Implementation ……..58

4.5 Frontend Implementation ……..60

4.6 Database Integration……...61

4.7 Security Measures Implemented ……...62

4.8 Testing and Debugging Process…….64

4.9 Challenges Faced and Solutions ……65

4.10 Conclusion ……....……66

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Summary……...67

5.2 Limitations and Challenges ……..68

5.3 Recommendations ……70

5.4 Conclusion ……71

REFERENCES ……73

APPENDIX ……..74

ABSTRACT

Welfare schemes provide essential financial support to their members, offering services like low-interest loans, retirement benefits, and end-of-year incentives. However, many of these schemes struggle with inefficiencies caused by outdated processes, leading to delays, errors, and frustration for both members and administrators. This project aims to develop a user-friendly financial management system to address these challenges. The system will allow members to easily track their contributions, loans, and benefits while providing administrators with tools to manage resources more effectively. By simplifying operations and improving transparency, this solution seeks to enhance the experience for all users, fostering trust and ensuring the welfare scheme runs smoothly for years to come.

CHAPTER ONE

INTRODUCTION

1.0 BACKGROUND STUDY

Managing finances within a welfare scheme can be complex, especially when it involves handling contributions, loans, and benefits for a large group of people. For this welfare scheme, **THE AUCHI POLYTECHNIC REGISTRY WELFARE SCHEME.** Members contribute monthly, access loans at minimal interest rates, and receive end-of-year incentives. Retired members can claim their total contributions with a 40% interest rate, while funds are also set aside to support the families of deceased members.

While the welfare scheme has been beneficial, its current financial processes lack the efficiency and clarity needed to keep up with the needs of its members. Many still rely on manual methods, which can lead to delays, errors, and confusion in tracking contributions, loan repayments, and benefits.

By implementing a financial management system, the welfare scheme can offer members a seamless experience, allowing them to monitor their finances, apply for loans, and stay informed about their benefits with ease. This system will not only simplify processes but also promote transparency and strengthen trust between the scheme and its members.

1.1 STATEMENT OF PROBLEM

Managing finances within the Auchi Polytechnic Registry Welfare Scheme has become increasingly challenging due to its reliance on manual processes. Members often face delays when trying to access information about their contributions, loan balances, or retirement benefits. It's difficult for them to track their financial standing, leaving many uncertain about their status within the scheme.

On the administrative side, handling records manually increases the chances of errors and inefficiencies, making it harder to provide accurate and timely updates to members. This lack of clarity and accessibility can lead to frustration and mistrust among members who rely on the scheme for financial security.

Without a proper system in place, the welfare scheme risks falling short of its potential to provide seamless and transparent services. A digital financial management system is needed to address these issues, making it easier for members to stay informed and for administrators to manage the scheme effectively.

1.2 Aim and Objectives of the Study

The goal of this study is to create a user-friendly financial management system for the Auchi Polytechnic Registry Welfare Scheme. This system will help simplify processes, improve transparency, and make it easier for members and administrators to manage financial activities.

To achieve this, the study focuses on the following objectives:

1. Develop tools to help members track their monthly contributions, loans, and expenses effortlessly.

2. Set up a loan management portal where members can explore loan options, understand interest rates, apply for loans, and monitor repayments.

3. Provide an easy way for members to view their total contributions, including any interest accumulated, especially for retirees.

4. Create a dedicated section to guide retired members on how to claim their contributions with the 40% interest rate.

5. Highlight additional benefits, such as end-of-year contributions and the incentives members can expect.

6. Build a platform for announcements and updates, so members can stay informed about changes and upcoming benefits.

7. Establish a reliable contact section for members to reach administrators quickly for support or questions.

This system is designed to empower members with clear and accessible financial information while easing the administrative burden of managing the scheme.

1.3 SIGNIFICANCE OF THE STUDY

The implementation of a financial management system for the Auchi Polytechnic Registry Welfare Scheme is essential for addressing the current challenges faced by members and administrators. This study is significant because it will transform the way the welfare scheme operates, ensuring that financial processes are streamlined and easily accessible to all stakeholders.

For members, the system will provide a transparent and reliable platform to track their contributions, loans, and benefits. Retired members will benefit from clear and accessible information about their accrued contributions and interest, while active members will find it easier to apply for loans, monitor repayments, and stay updated on incentives.

For administrators, the system will simplify record-keeping, reduce errors, and enable efficient management of funds and member accounts. It will also improve communication by providing a centralized platform for updates, announcements, and member support.

Overall, this study is significant because it addresses the need for modernized operations within the welfare scheme. By improving transparency and efficiency, the system will foster trust, enhance user satisfaction, and ensure the scheme's long-term success.

1.4 MOTIVATION

My motivation for this study is deeply personal. Having seen firsthand the challenges faced by Welfare Scheme's through my mother, who serves as the secretary, I have a clear understanding of the issues that arise in managing the scheme's finances. She often talks about how difficult it is to ensure accurate record-keeping, process loans efficiently, and keep members informed about their contributions and benefits.

I've also seen the impact on members—people who rely on the scheme for financial support but sometimes experience delays or confusion due to the manual systems in place. These challenges make it harder for the welfare scheme to fulfill its purpose of providing timely and reliable support to its members.

This study is motivated by a desire to help ease these struggles. By implementing a financial management system, I hope to create a solution that not only simplifies my mother's work but also improves the experience for all members of the welfare scheme.

1.5 SCOPE OF STUDY

This study will focus on developing a comprehensive financial management system for the Auchi Polytechnic Registry Welfare Scheme, specifically targeting the areas that impact members and administrators the most. The system will cover the following key aspects:

1. Contribution Tracking: Members will be able to easily view and track their monthly contributions, including any accrued interest for retirees.

2. Loan Management: The system will provide details on available loans, interest rates, and the application process, allowing members to apply for loans and monitor repayments.

3. Retirement Benefits: Retired members will have access to information about claiming their contributions with the 40% interest rate.

4. End-of-Year Incentives: Members will be able to see how end-of-year contributions are distributed and track their eligibility for related benefits.

5. Communication and Updates: The system will provide members with a centralized space for announcements, updates, and changes to the welfare scheme, as well as a contact option for support.

The scope will also include system testing, feedback collection from members and administrators, and refining the system to meet the needs of all stakeholders. However, the study will not delve into the broader financial practices of the institution outside of the welfare scheme itself.

The focus is on ensuring that the system is easy to use, secure, and accessible, with the goal of improving overall management and member satisfaction.

1.6 LIMITATIONS OF THE STUDY

While this study aims to create a practical and efficient financial management system for the Auchi Polytechnic Registry Welfare Scheme, there are a few challenges that might limit its full implementation.

1. Technology Limitations: The system's effectiveness could be impacted by the existing technological infrastructure within the institution. If the hardware, software, or internet connectivity is not up to standard, it could affect the performance of the system.

2. Budget Constraints: Developing and maintaining a digital system requires funding for software, licenses, and potentially additional technical support. If the available budget falls short, it could limit the system's features or scalability.

3. Training and Adoption: For the system to be successful, both members and administrators need to be trained on how to use it. Some individuals may find it difficult to adapt to a digital system, especially if they are not familiar with using technology, which could slow down the implementation process.

4. Data Privacy and Security: Managing sensitive financial information requires a high level of security. While the system will be designed with security in mind, ensuring that it meets the

necessary data protection standards could present challenges, particularly with limited resources for advanced security measures.

5. Resistance to Change: Some members or administrators may be hesitant to transition from a manual system to a digital one. This resistance could delay the implementation and affect the overall success of the project.

Despite these challenges, the study aims to find solutions to overcome them and create a system that makes managing finances for the welfare scheme more efficient and accessible for everyone involved.

CHAPTER TWO

2.0 INTRODUCTION

A financial management system is an important tool for any organization in streamlining operations and offering efficient services to its members. Various studies and implementations on such systems over the years have shown how they could enhance transparency, improve record-keeping, and ensure user accessibility. This section introduces existing knowledge about financial management systems and welfare schemes, highlighting the interplay between technology and efficient financial administration.

Welfare schemes are generally supposed to deal with manifold financial activities like members' contribution, loan disbursement and repayment, retirement benefits, and incentive schemes. Most of the manual systems practiced are often found burdened with inefficiency, errors, and delays. In addition, there is a felt need to digitalize the operations with an ever-increasing scope and scale of welfare schemes.

The concept of financial management systems has, however, been explored by different researchers in other contexts, like microfinance institutions, cooperatives, and employee benefit schemes. These works have shown how well such a system is capable of guaranteeing financial accountability and operational efficiency. However, gaps still exist, particularly on how such

systems can suit small-scale welfare schemes like the Auchi Polytechnic Registry Welfare Scheme.

In the last couple of years, technology integration in financial management has shown promising results. Systems that can allow members to track contributions, apply for loans, and access benefits through user-friendly interfaces are becoming common. However, resistance to change, lack of technical infrastructure, and concerns about data privacy remain significant barriers to adoption.

This paper now sets a platform to understand the relevance of financial management systems to welfare schemes and presents challenges unique in welfare schemes. This thus sets a baseline for the next sections that will narrow down to specific aspects of financial management systems, operations of welfare schemes, and the use of technology in transforming such systems.

Financial management systems are thus very important in organizations that receive contributions, make disbursements, and allocate benefits. These systems automate some of the complex financial processes that may be prone to human error and ensure accuracy in keeping records. In the context of welfare schemes, which are supposed to serve as financial support networks for their members, the need for such systems is even more critical. Welfare schemes usually entail many transactions like monthly contributions, loan repayments, retirement benefits, and incentives, which need effective management. (The International Labour Organization (ILO)Ansar et al. 2023) emphasize the importance of digital solutions in addressing inefficiencies and enhancing transparency, particularly in resource-constrained environments.

Welfare schemes have conventionally used manual methods to manage these financial operations. While these methods may be effective in small-scale systems, they become inefficient and prone to errors as the membership base increases. Some of the frequent problems that arise include

errors in record-keeping, delays in processing loans, and miscommunication concerning member benefits. These may lead to a loss of confidence between members and administrators and, in turn, affect the effectiveness of the welfare scheme.

It is here that research has indicated the immense benefits that financial management systems provide as a solution to these challenges. For instance, one study concerning cooperatives and microfinance organizations revealed that the adaptation of digital systems increased transparency and accelerated processes, which bettered member satisfaction. Financial systems implemented in employee benefit programs have equally demonstrated how automation can simplify complex calculations, such as interest on contributions and loan repayment schedules, hence reducing administrative workloads.

Yet, with all such developments, welfare schemes normally face challenges unique to the nature of welfare schemes not usually addressed by generic financial management systems. In contrast with other traditional financial institutions, welfare schemes are member-owned and most often very resource constrained. Any system proposed has to be affordable, user-friendly, and suited for the specific needs of the scheme. In the case of the Auchi Polytechnic Registry Welfare Scheme, the diverse financial services provided—including loans, retirement benefits, and incentives—require a system that integrates these functions seamlessly while remaining accessible to all members.

Another significant aspect of implementing financial management systems in welfare schemes is the need to balance functionality with accessibility. Many members may be unfamiliar with digital platforms, especially since some have worked with a manual system for several years. Thus, designing a user-friendly system and offering proper training are among the key factors to

successful implementation. Research has reinforced that when it comes to financial systems, user-oriented design ensures ease of use translates to user engagement and satisfaction.

Infrastructural and organizational problems also make the adoption of financial management systems difficult. The use of such systems is highly restricted in institutions like Auchi Polytechnic due to poor internet connectivity, old hardware, and lack of IT knowledge on the part of administrators. In sum, addressing these barriers requires a holistic approach comprising infrastructure development, capacity building, and ongoing technical support.

This review of existing knowledge underlines the increasing prominence of financial management systems within welfare schemes. While clear benefits of automation, transparency, and efficiency exist, successfully applying such systems requires due attention to the particular needs and challenges of each scheme. The subsequent sections will develop the specific functions of the financial management system, impacts on welfare scheme operations, and the role of technology in bridging gaps in financial management.

2.1 FINANCIAL MANAGEMENT SYSTEMS

A financial management system is a system that is meant to automate and streamline the financial processes in an organization. It is a single platform where transactions are handled, financial data is tracked, and reports are prepared. In welfare schemes, where many financial activities-both routine and ad hoc-are performed, like contributions, loan disbursements, and benefits processing, an FMS becomes very important to ensure accuracy, transparency, and efficiency.

Importance of Financial Management Systems

The main objective of the financial management system is to enhance operational efficiency. In welfare schemes, these activities include the simplification of tasks such as member contributions, interest on savings, and loan repayments. In this way, an FMS automates these functions and reduces the administrative burden on the staff, thereby minimizing human errors and keeping financial records current and accessible.

To the member, the FMS presents an avenue of transparency and ease. A member can keep track of his or her financial activities with great ease without having to refer back to the manual record books or continually bothering the administrator for follow-ups. In this case, such transparency instills a sense of accountability and trust among the members towards the scheme.

Features of Financial Management Systems

Modern financial management systems are designed with a variety of features to meet the needs of welfare schemes, including:

1. Contribution Tracking: In an FMS, members can track their contributions over time and the accrued interest for retirees. This feature is helpful in those schemes that provide retirement benefits, as it gives transparency into financial growth and entitlement.

2. Loan Management: A fully-fledged FMS will have a loan management module, which can handle loan applications, calculate interest, and track repayments. This ensures that loans are efficiently disbursed and repayments are accurately monitored.

3. Budgeting and Reporting: An FMS is able to provide detailed financial reports that will help administrators analyze income, expenses, and overall financial health. This aids in budgeting and long-term planning for the welfare scheme.

4. Member Portal: Many of the systems have a very user-friendly portal through which the member can view all financial data, apply for loans, and view scheme notifications.

5. Notification and Alerts: Members receive due date payment notifications, loan approval notifications, and future benefit notifications without relying on physical reminders.

The Journal of Financial Literacy and Wellbeing discusses how digital platforms make processes like loan applications easier, track contributions, and enhance transparency, thus becoming vital for any modern welfare scheme. (Bushra, 2024)

Likewise, financial management systems have been embraced in large organizations for employee benefit schemes to facilitate retirement savings plans, loan programs, and other financial benefits. Such systems have been proved to enhance transparency and cut down on administrative costs, thereby being an asset to both the employees and administrators.

Challenges and Considerations

The challenges associated with implementing an FMS, however, cannot be overruled. Developing or purchasing such a system and maintaining it may pose too much cost for certain small welfare schemes that belong to a tight budget.

Data security is one of the major concerns in managing sensitive financial information. An effective FMS should, therefore, be equipped with appropriate security measures to protect member data from unauthorized access and other cyber threats. Ensuring compliance with data protection regulations is also important in terms of trust and avoidance of legal issues.

Relevance to Auchi Polytechnic Registry Welfare Scheme

The tailored FMS will help the Auchi Polytechnic Registry Welfare Scheme solve a lot of its problems. The system will provide tools for tracking contributions, managing loans, and calculating retirement benefits, hence enhancing the transparency and efficiency of operations. Besides, it provides features such as member portals and automated notifications that increase member engagement and satisfaction.

This subsection explores the role of financial management systems in welfare schemes, featuring their features, applications, and challenges. These will provide the background understanding for how such a system would transform the Auchi Polytechnic Registry Welfare Scheme.

2.2 OPERATIONS OF WELFARE SCHEMES

Welfare schemes provide various forms of financial benefits and security for their members. Many of them are established for specific groups, which could be in the form of employees,

members of cooperatives, or even retirees, providing loans, retirement savings, and other incentives. Generally, welfare scheme operations involve maintaining member records and contributions, disbursements, tracking of benefits, and financial records. While these activities are crucial to the running of these, they can be complicated and ridden with inefficiencies when managed manually.

Core Functions of Welfare Schemes

1. Member Contributions:

The major source of funds for welfare schemes comes from the regular contributions made by their members. The usually monthly collected contributions may even accrue interest to be distributed to members upon their retirement or withdrawal from the scheme. For this purpose, a correct record of these contributions is of prime importance, which ensures fairness and transparency.

2. Loan Disbursement and Management:

Many welfare schemes provide low-interest loans to their members, serving as a reliable financial safety net. The loan disbursement process involves evaluating applications, approving loans based on eligibility criteria, and setting repayment schedules. Managing these loans effectively requires careful monitoring to prevent defaults and maintain the financial stability of the scheme.

3. Retirement Benefits:

One of the important objectives of welfare schemes is to give financial support to members after retirement. It mostly involves returning the contribution made by the members with interest, so that they have a financial cushion for their post-retirement life. The calculation of these benefits and ensuring timely disbursement becomes an important part of the operations of the scheme.

4. Incentives and Support for Families:

Welfare schemes sometimes have additional advantages such as end-of-year incentives and financial support in case of member deaths to their families. Such activities strengthen the ties between the scheme and the members by building trust and loyalty.

5. Record-Keeping and Reporting:

A welfare scheme cannot function properly unless proper records of all the monetary transactions, such as contributions, loans, and benefits, are appropriately maintained. Administrators need to prepare periodical reports for the sake of accountability and transparency among the membership.

Challenges in Operations of Welfare Schemes

There are a number of challenges that hinder the effective operability of welfare schemes.

1. Manual Processes:

Most of the schemes are still at a stage where contribution tracking, loan processing, and record management are done manually. This frequently causes errors, delays, and inefficiencies that undermine members' trust and satisfaction in the scheme. Studies from Nigerian cooperatives underline delays and inaccuracies in manual systems, leading to inefficiencies in benefit disbursement and member trust. ▌Yusha'u .L. Angu, Science world Journal 2024)

2. Limited Resources:

In welfare schemes, the budget size is usually small; it's hard to get financing for modern technologies or seek experts in specialized skills. This will limit their capabilities of scaling up operations or best practice in financial management. The ILO (International Labour Organization) highlights barriers such as poor infrastructure and the high costs of transitioning to automated systems when it comes to transitioning into automated systems (Michael Gallo, 2021)

3. Issues on Transparency and Communications:

In addition, members often have difficulties obtaining information about their contributions, loans, or benefits. This may also create some misunderstandings or disputes between the members and the administrator.

4. Data Security Risks:

The sensitivity of financial data requires strict security measures to ensure the protection against unauthorized access and data breaches. A welfare scheme that does not provide a secure system can expose itself to fraud or critical data loss.

5. Adaptation to Changing Needs:

As the membership increases or the regulatory requirements change, the welfare scheme has to update its operations, which may be quite demanding without flexible systems that are easily updated or modified.

Need for Digital Transformation

The challenges being faced by the welfare schemes are a call for the need to embrace digital transformation. In this way, a financial management system can be adopted to automate tasks that are routine, improve accuracy, and enhance transparency. For instance, it can offer members real-time access to their financial data, make loan applications online, and automate the calculation of retirement benefits.

Studies have established that welfare schemes adopting digital solutions record improved efficiencies and member satisfaction. A case study on an employee welfare scheme in South Africa, for example, established that the introduction of a digital management system reduced the administrative workload by 40% and increased member engagement by 60%. (Regina .W. M. , 2016)

Relevance to Auchi Polytechnic Registry Welfare Scheme

The Auchi Polytechnic Registry Welfare Scheme is one of the examples of organizations that will greatly benefit by modernizing its operations. An FMS would help them solve the problems

they faced in the course of carrying out their operations, including tracking members' contributions and loans and providing timely statements to members. The scheme will also provide transparency, as well as effective communication needed for the long-term solvency of the scheme.

This section brings forth the critical operations of welfare schemes, challenges in implementation, and how far the digital solution can be addressed for these challenges. The next section shall discuss the different ways in which technology adoption will transform financial systems for better efficiency in the overall welfare scheme.

2.3 TECHNOLOGY ADOPTION IN FINANCIAL SYSTEMS

The adoption of technology in financial systems has transformed the way organizations manage and process financial data. Digital transformation, from traditional banking systems to small-scale welfare schemes, has improved efficiency, accuracy, and transparency in financial operations. For welfare schemes in particular, where contributions, loans, and benefits need to be carefully tracked and managed, technology adoption provides several advantages.

Importance of Technology in Financial Systems

1. Automation of Financial Processes:

Technology automates repetitive tasks like data entry, calculations, and record-keeping, reducing the risk of human error. In welfare schemes, this means correct tracking of member contributions, real-time updates on loan repayments, and automatic calculation of interest.

2. Accessibility and Convenience:

Digital financial systems allow members to access their financial information from any location through online platforms or mobile applications. This accessibility fosters transparency and improves member satisfaction. Members can view their contribution history, loan status, and benefits in real time without necessarily having to visit the welfare office physically. The ILO reinforces that data automation and secured systems reduce errors and increase adherence to regulatory compliance. The adoption of mobile money systems in Sub-Saharan Africa brought

about increased accessibility and enhanced transparency, with 31% citing ease of use as the major advantage. (Amare .T. Alemu, 2024)

3. Improved Record Management:

Digital systems store financial records securely and ensure easy retrieval. Unlike manual systems, which are prone to loss or damage, electronic records are backed up and protected, ensuring data integrity over time.

4. Data Analysis and Reporting:

Technology allows for advanced data analysis, thus enabling administrators to create comprehensive financial reports. These insights are helpful in budgeting, forecasting, and assessing the financial health of the scheme. For instance, an administrator can analyze the trend of loan applications or member contributions in order to make informed decisions.

Benefits of Technology Adoption in Welfare Schemes

1. Efficiency in Operations:

Technology automates core operations like contribution collection, loan disbursement, and benefit calculations. This reduces delays and administrative workload, thus allowing welfare schemes to serve their members more effectively.

2. Enhanced Transparency:

Digital systems allow members to access their financial data directly, which increases transparency. Members can verify their contributions, monitor loan repayments, and calculate expected benefits, reducing the chances of disputes.

3. Cost-Effectiveness:

While the initial investment in technology may be huge, digital systems often reduce long-term operational costs. For example, automating processes reduces the need for extensive manual labor, and electronic communication minimizes paperwork.

4. Security and Compliance:

Modern financial systems boast a number of security features that protect sensitive member data, including encryption and multi-factor authentication. Besides, the system can also be made to be compliant with regulatory standards to ensure that the welfare scheme is operating within legal frameworks.

Challenges of Technology Adoption

Despite its many advantages, the adoption of technology in financial systems is not without challenges:

1. High Initial Costs:

The implementation of a digitalized financial management system involves huge expenses on software, hardware, and training. This becomes a limitation for small-scale welfare schemes, as their budget is relatively low.

2. Resistance to Change:

Administrators and members who are accustomed to manual systems may be reluctant to change to new technologies. Effective change management and training programs will be needed to overcome such resistance.

3. Infrastructure Limitations:

Poor internet connectivity, outdated hardware, and a lack of technical support are major barriers in implementing and utilizing digital systems, especially in developing regions.

4. Data Privacy Concerns:

The handling of sensitive financial information online presents chances for data breaches and hacking. Ensuring tight security measures and compliance with data protection regulations would be crucial in this regard.

Case Studies on Technology Adoption

Research studies and real-world examples prove the transformative potential of technology in financial systems. For example, in a study conducted on the cooperative societies of India, it was

found that the adoption of digital financial platforms increased operational efficiency by 70% and reduced member grievances by 50% (Anjan .A. Kaikini, 2020).

In another instance, a welfare scheme in Nigeria implemented a web-based management system to track contributions and loans in real time, which considerably enhanced transparency and member trust. (Dauda Rilwan, 2021)

Relevance to Auchi Polytechnic Registry Welfare Scheme

Technology adoption for the Auchi Polytechnic Registry Welfare Scheme will provide ample opportunity to surmount most of the operational challenges. A customized financial management system will automate processes such as contribution tracking, loan management, and benefit calculations, reducing administrative workload and improving accuracy. Additionally, features such as a member portal and automated notifications can enhance communication and transparency, fostering trust among members.

By addressing the barrier of resistance to change and infrastructure limitations, technology will help the scheme towards the aim of an efficient, secure, and member-centric system for financial management.

This subsection has discussed how technology has transformed financial systems, keeping in view the relevance to welfare schemes. The following section considers the review of existing systems at Auchi Polytechnic or similar institutions, identifying deficiencies the proposed financial management system would aim to address.

2.4 EXISTING SYSTEMS IN AUCHI POLYTECHNIC OR SIMILAR INSTITUTIONS

The existing financial systems in institutions like Auchi Polytechnic are a mix of both manual and simple digital tools. While these may work to some extent for daily operations, they are usually not sufficient to address the increasing complexities of modern financial management, especially in welfare schemes. This section reviews the state of the financial system in the Auchi Polytechnic Registry Welfare Scheme and other similar institutions, with regard to their limitations and ways of improvement.

Current Financial Management Practices

Presently, the financial management practices of the Auchi Polytechnic Registry Welfare Scheme are highly manual, with physical records and spreadsheets used to track contributions, loans, and benefits. Administrators manually record monthly contributions from members, process loan applications, and calculate retirement benefits. While these methods may be sufficient for small-scale operations, they become inefficient and error-prone as the membership base grows and financial activities expand.

Manual systems often lead to some hitches like:

Data Entry Errors: Human errors in recording the actual contributions or in calculation of interest create discrepancies leading to dissatisfaction and disputes from members.

Delayed Processes: Manual handling of loan applications and benefit calculations leads to sluggish service delivery and keeps the members frustrated.

Lack of Transparency: Members hardly or do not have any insight into their financial records, creating no trust in the way the scheme is being conducted.

Besides, there are no centralized platforms where members can access their financial information or communicate with the administrators. The members have to physically go to the welfare office or depend on verbal updates, which is time-consuming and not convenient.

Case Study of Similar Institutions

A review of related welfare schemes that have been in operation from other educational institutions and employee cooperatives likewise indicates the same operational problem. For example, researchers studied a staff cooperative based in Lagos, Nigeria; the study revealed that where there is a reliance on manual systems, inefficiency is experienced in the forms of "delayed disbursement, contribution records, and inconsistencies among others" (Stephen Oluwatobi, 2017). The study recommended introducing digital solutions to enhance operational efficiencies and member satisfaction.

Some have tried to introduce simple spreadsheet-based or light accounting software for financial management. While such tools better organize the data, they lack the enterprise-level functionality of a state-of-the-art financial management system-real-time updates, member

portals, and notifications. In the end, most of them still experience various inefficiencies and low member interaction.

Gaps in the Existing Systems

The inadequacies of current financial systems in Auchi Polytechnic and similar institutions can be summarized under the following areas:

1. Automation:

Manual operations have dominated the operations, bringing along inefficiencies and delays. There is no automation to track contributions, loan processing, or interest calculation.

2. Member Accessibility:

No access is given to the members to independently go through their financial records or perform their transactions online. This negates convenience and transparency, having to rely on physical interaction.

3. Data Security:

Manual records are susceptible to loss, damage, and access by unauthorized persons. Even the digital tools, such as spreadsheets, generally do not have strong security features to protect sensitive financial data.

4. Scalability:

As the membership grows, it becomes cumbersome and difficult to manage a manual system. Inadequate scalable solutions make it difficult for welfare schemes to efficiently handle higher volumes of transactions.

5. Reporting and Analysis:

Current systems lack the facilities for advanced reporting and data analysis that are crucial in financial planning and decision-making. This always leaves the administrators with the only option of using tedious manual methods of compiling reports or reviewing the financial status of the scheme.

The gaps in existing systems highlight the pressing need for a dedicated financial management system tailored to the specific needs of welfare schemes like the Auchi Polytechnic Registry Welfare Scheme. Such a system would address key operational challenges by:

- Automating routine tasks to improve efficiency and reduce errors.
- Providing members with online access to their financial information, enhancing transparency and convenience.
- Strengthening data security through encryption and access controls.
- Support scalability for increasing membership and financial activities.
- Advanced reporting capabilities to aid in better financial planning and management.

Lessons from Successful Implementations

Institutions that have adopted digital financial management systems demonstrate the transformative potential of such solutions. For instance, a staff cooperative in South Africa implemented a cloud-based platform that allowed members to view their contribution history, apply for loans online, and receive automated notifications about due payments. This system reduced administrative workload by 50% and significantly improved member satisfaction.

The Kenyan welfare scheme introduced a mobile application through which members could track their savings and benefits in real time. The introduction of this facility raised the level of member participation by 70% and reduced disputes over contribution records by 40% (Gabriel Thegeya, 2012).

The examples above demonstrate how tailored financial management systems can meet the peculiar needs of welfare schemes, enhancing operational efficiency and member experience.

Relevance to Auchi Polytechnic Registry Welfare Scheme

The lessons from other institutions therefore provide useful insights into the Auchi Polytechnic Registry Welfare Scheme. A financial management system would overcome the present operational challenges and ensure a more efficient, transparent, and member-centric operation. The following section explores the proposed solutions in detail with regard to the design and implementation of a financial management system for the scheme.

2.5 SUMMARY OF LITERATURE REVIEW

In this chapter, the different aspects of the financial management system implementation for Auchi Polytechnic Registry Welfare Scheme were discussed; best practices and challenges that surround financial management and the adoption of technology.

Summary of Findings

1. Financial Management in Welfare Schemes:

Financial management within welfare schemes encompasses the management of member contributions, loans, and benefits-all of which need to be kept track of and updated. Most welfare schemes still rely on a manual system, which usually brings about inefficiencies, errors, and a lack of transparency, hence diminishing members' trust and satisfaction. The adoption of digital financial systems has been shown to improve operational efficiency, accuracy, and transparency, benefiting both administrators and members.

2. Technology Adoption in Financial Systems:

The adoption of technology in financial systems, especially in welfare schemes, has many advantages, such as automation of financial processes, real-time access to financial data, and enhanced transparency. Though resistance to change, infrastructure limitations, and high initial costs are some of the challenges that may impede the adoption of technology, successful case studies from similar institutions prove the transformational potential of digital solutions.

3. Related Systems in Auchi Polytechnic and Institutions of Similar Standing

Most of the financial systems currently in use in institutions, including Auchi Polytechnic, are manual and very limited in scope. This may also be attributed to various efficiencies and inaccuracies. Such systems are poorly suited for a modern welfare scheme since they cannot effectively scale with increasing membership and financial activities. These existing system gaps were further reviewed, as well as related work in similar institutions, thus highlighting the need for a tailored financial management solution.

Conclusion

The literature review thus formed the basis on which a design and implementation of an FMS were realized that suit the particular needs of the Auchi Polytechnic Registry Welfare Scheme. Addressing significant challenges in this area, such as manual entry, lack of transparency, and inefficiency, will ensure a digitized system greatly improves member contribution management, loan, and benefit management. The insights gathered from different successful implementations from other institutions will inform and help create a system that is friendly to users, secure, and scalable, benefitting the administrators and members of the association.

With the framework now designed, the succeeding steps involved the design details and details of implementation; hence, the design aimed at having an efficient system, highly accessible, with the possibility of adapting to the growing welfare scheme.

CHAPTER THREE

3.1 INTRODUCTION

This chapter provides a comprehensive analysis and design blueprint for the Auchi Polytechnic Registry Welfare Scheme Web Application. This system aims to streamline financial management, contribution tracking, and loan processing for scheme members, while also providing a robust administrative dashboard for efficient management of users, contributions, and loans. The chapter details the system's requirements, architectural design, database structure, user interaction flows, and critical security considerations to ensure a secure, scalable, and user-friendly implementation. This detailed design will serve as the foundation for the subsequent implementation phase.

3.2 SYSTEM OVERVIEW

The Auchi Polytechnic Registry Welfare Scheme Web Application empowers members to seamlessly manage their financial interactions with the scheme. It enables members to register, securely log in, conveniently make financial contributions, apply for loans with transparent terms, and effortlessly track their financial status within the scheme. The system caters to two primary user roles:

- **Users (Members):** Members can register an account, securely log in, view their complete financial records (contributions, loan history, etc.), make contributions conveniently, apply for loans with clear information about interest rates and repayment schedules, track the status of their loan applications, and view their loan repayment status.

- **Administrators:** Administrators have comprehensive control over the system. They can manage user accounts (create, update, deactivate), approve or reject loan applications based on predefined criteria, monitor all financial transactions, generate reports on contributions and loan activity, manage scheme settings, and oversee the overall health and security of the system.

The system is designed as a modern, responsive web-based application. The frontend will be developed using HTML, CSS, and JavaScript to provide a rich and interactive user experience. The backend will be powered by Node.js, leveraging the Express.js framework for robust API development and efficient request handling. MongoDB will serve as the database, providing a flexible and scalable solution for storing user data, financial records, and transaction logs. Security is paramount, and the system will incorporate robust authentication mechanisms, including bcrypt for secure password hashing, and input validation to mitigate common web vulnerabilities.

3.3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

3.3.1 **Functional Requirements**

- **User Management:**
  - User registration with validation (e.g., email format, password complexity).
  - Secure user authentication (login/logout).
  - Password reset functionality.
  - User profile management (update contact details, etc.).
  - Administrative user management (create, update, deactivate user accounts).
- **Contribution Management:**
  - Recording and tracking member contributions.
  - Viewing contribution history.
  - Generating contribution reports (for members and administrators).
  - Handling different contribution types (if applicable).
- **Loan Management:**
  - Loan application process with clear information on loan terms and interest rates.
  - Loan application review and approval/rejection workflow for administrators.
  - Loan repayment tracking.
  - Loan repayment reminders (optional).
  - Calculation of interest and total repayment amount.
  - Loan status updates and notifications.
- **Reporting and Analytics:**
  - Generating reports on contributions, loans, and overall scheme financials.

    - Data visualization (charts and graphs) for better understanding of financial trends.

- **Notification System:**
    - Email notifications for loan approvals, status updates, and payment reminders.
    - In-app notifications (optional).
- **Administrative Panel:**
    - Dashboard for administrators to view key system metrics.
    - Tools for managing users, contributions, and loans.
    - Report generation and export.
    - Scheme settings management.

3.3.2 Non-Functional Requirements

- **Scalability:** The system must be able to handle a large number of users and transactions concurrently without performance degradation.
- **Security:** Robust security measures are essential to protect sensitive financial data. This includes secure authentication, authorization, data encryption, and protection against common web vulnerabilities.
- **Performance:** The system should provide fast response times for user interactions and transactions.
- **Usability:** The system should have a user-friendly interface that is easy to navigate and understand, even for users with limited technical skills. A responsive design is crucial to ensure usability across different devices (desktops, tablets, and smartphones).

- **Reliability:** The system must be reliable and available, ensuring data integrity and minimal downtime.
- **Maintainability:** The system should be designed for easy maintenance and updates. Well-structured code and clear documentation are essential.
- **Portability:** The system should be portable across different environments (e.g., development, testing, production)

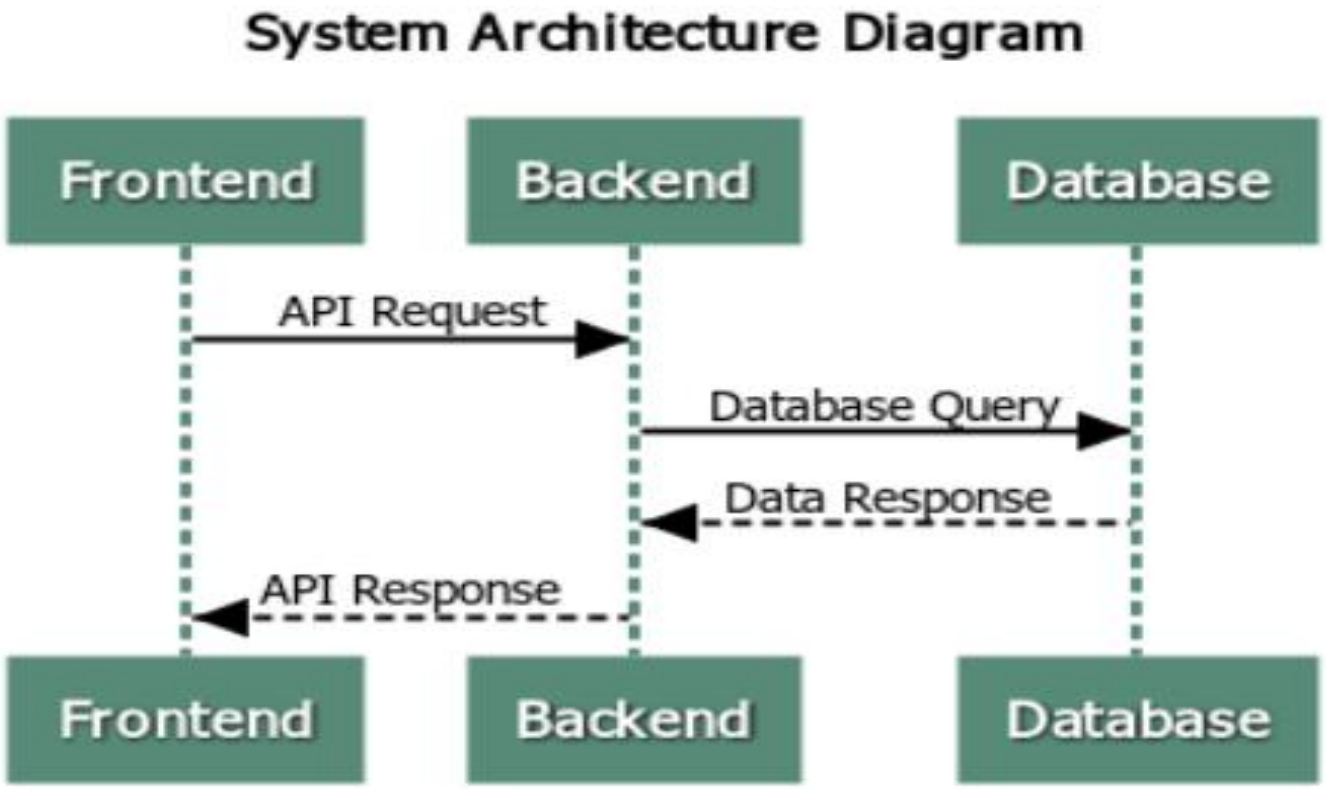
3.4 SYSTEM ARCHITECTURE

The application will adhere to a three-tier architecture, a well-established and robust approach for building scalable and maintainable web applications:

- **Presentation Tier (Frontend):** This tier is responsible for presenting the user interface to the user. It will be built using HTML, CSS, and JavaScript, providing a dynamic and interactive user experience. A modern JavaScript framework (e.g., React, Vue, or Angular – choice to be made during implementation) may be considered for enhanced interactivity and maintainability.
- **Application Tier (Backend):** This tier houses the core business logic of the application. It will be implemented using Node.js and the Express.js framework. This tier handles API requests from the frontend, interacts with the database, enforces security rules, and manages user sessions.

- **Data Tier (Database):** This tier is responsible for storing and retrieving data. MongoDB, a NoSQL database, will be used for its flexibility and scalability, making it suitable for handling the diverse data requirements of the welfare scheme.

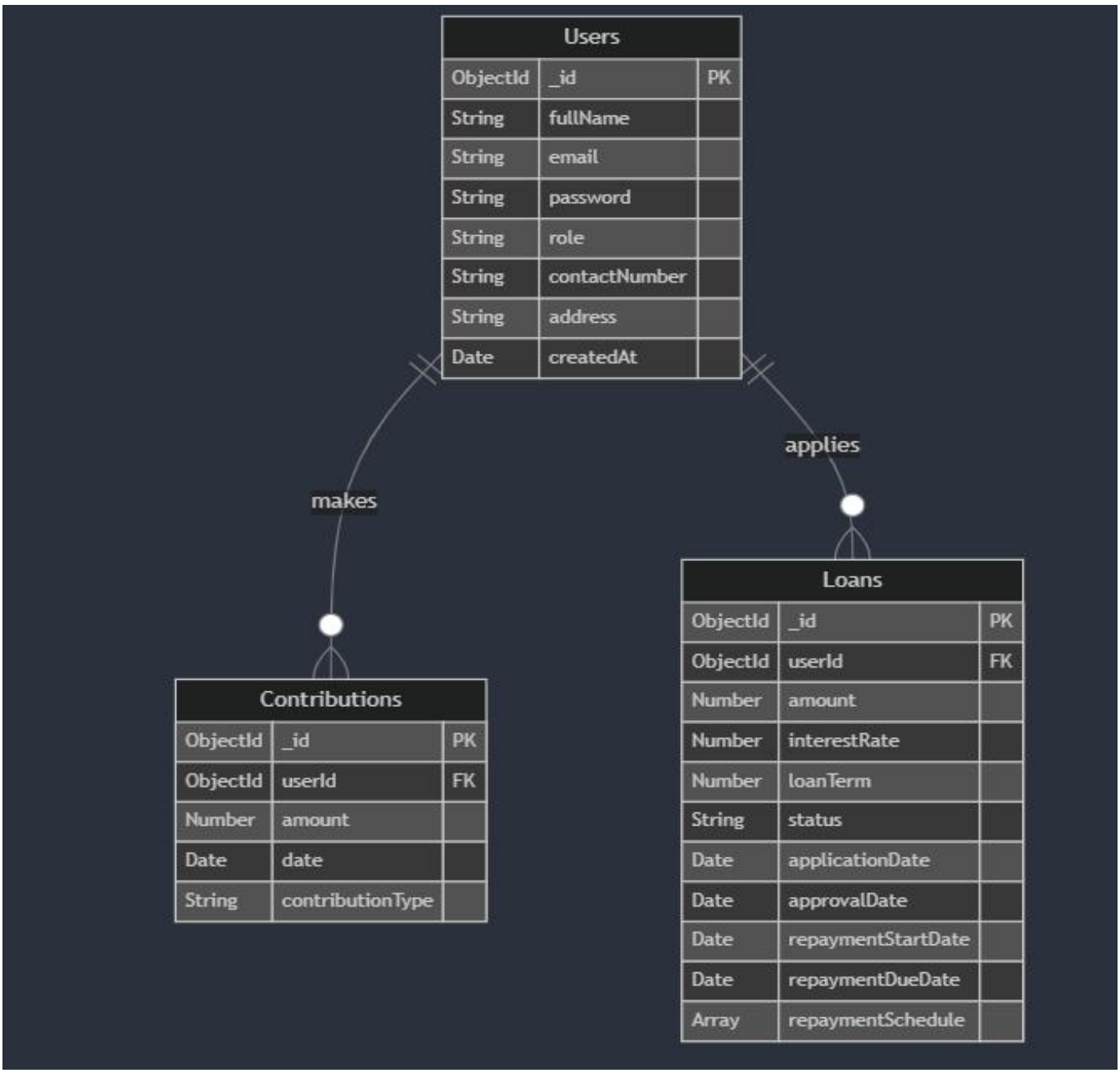
3.4.1 System Architecture Diagram



3.5 Database Design

The system will utilize MongoDB as its primary database. MongoDB's NoSQL nature offers flexibility in schema design and scalability, which are crucial for this application.

3.5.1 **Entity-Relationship Diagram (ERD)**


Users
ObjectId _id PK
String fullName
String email
String password
String role
String contactNumber
String address
Date createdAt
makes
applies
Contributions
ObjectId _id PK
ObjectId userId FK
Number amount
Date date
String contributionType
Loans
ObjectId _id PK
ObjectId userId FK
Number amount
Number interestRate
Number loanTerm
String status
Date applicationDate
Date approvalDate
Date repaymentStartDate
Date repaymentDueDate
Array repaymentSchedule

3.5.2 **DATABASE SCHEMA**

Users Collection:

```
json
[redacted]
{
  "_id": ObjectId,
  "fullName": String,
  "email": String,
  "password": String (hashed using bcrypt),
  "role": String ("user" or "admin"),
  "contactNumber": String,
  "address": String,
  "createdAt": Date
}
```

Contributions Collection:

```
json
[redacted]
{
  "_id": ObjectId,
  "userId": ObjectId (references Users),
  "amount": Number,
  "date": Date,
```

```
  "contributionType": String // e.g., "monthly", "special"
}
```

Loans Collection:

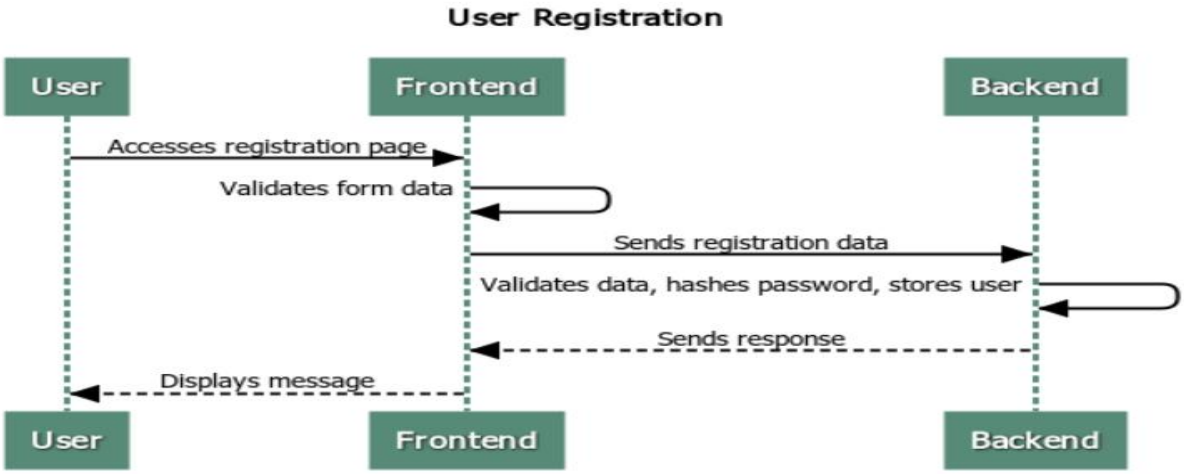
```
json
[redacted]
{
  "_id": ObjectId,
  "userId": ObjectId (references Users),
  "amount": Number,
  "interestRate": Number, // Added interest rate
  "loanTerm": Number,   // Added loan term (e.g., in months)
  "status": String ("Pending", "Approved", "Rejected", "Disbursed", "Repaid"),
  "applicationDate": Date, // Added application date
  "approvalDate": Date,    // Added approval date (if applicable)
  "repaymentStartDate": Date, // Added repayment start date
  "repaymentDueDate": Date,   // Added repayment due date
  "repaymentSchedule": [ // Array of repayment installments (optional)
    {
      "dueDate": Date,
      "amountDue": Number,
      "amountPaid": Number
    }
  ]
```

}

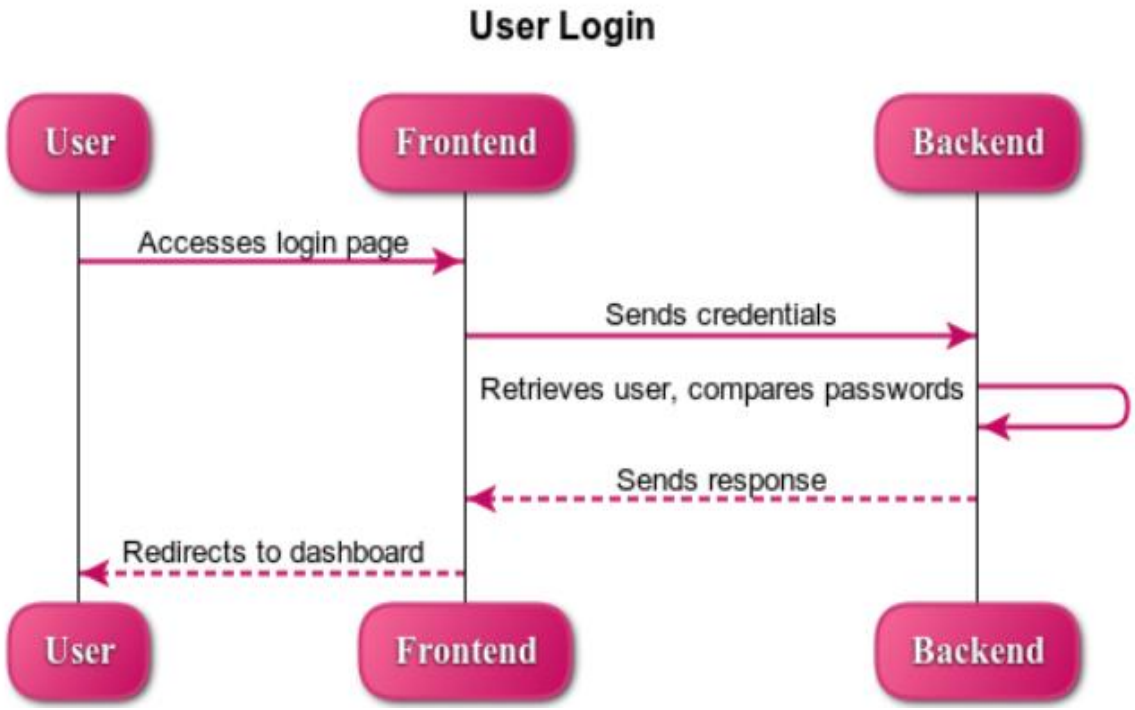
3.6 USER INTERACTION FLOW

3.6.1 User Registration And Login

1. User accesses the registration page.
2. User fills out the registration form with their details.
3. The frontend validates the form data (e.g., email format, password strength).
4. The frontend sends the data to the backend.
5. The backend validates the data, hashes the password using bcrypt, and stores the user in the database.
6. The backend sends a success/error response to the frontend.
7. The frontend displays a confirmation message or error message.

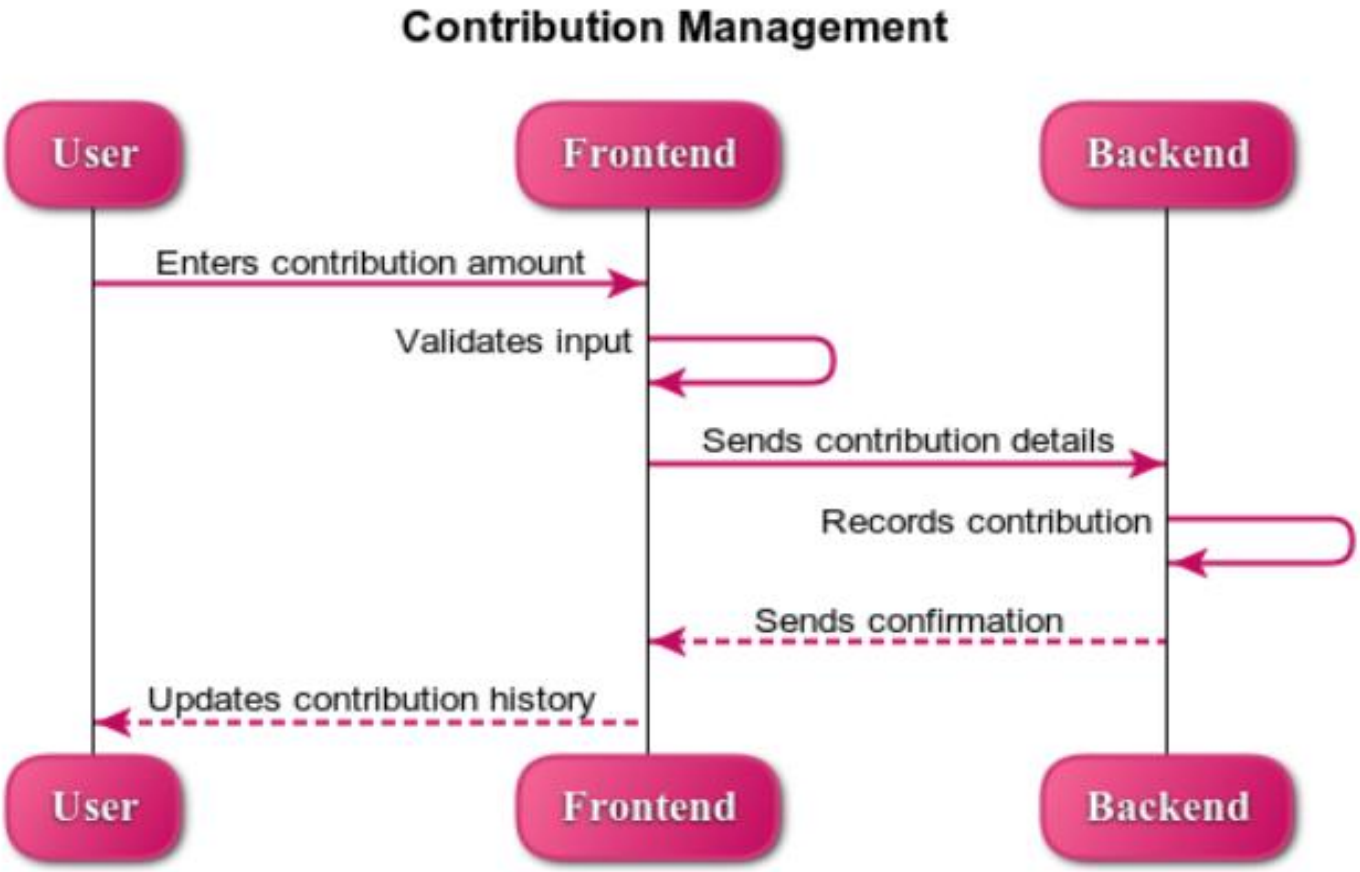


1. User accesses the login page.
2. User enters their username and password.
3. The frontend sends the credentials to the backend.
4. The backend retrieves the user from the database (based on username).
5. The backend compares the entered password with the stored hashed password.
6. If the passwords match, the backend creates a session.
7. The backend sends a success response to the frontend.
8. The frontend redirects the user to the appropriate page (e.g., dashboard).



3.6.2 **Contribution Management**

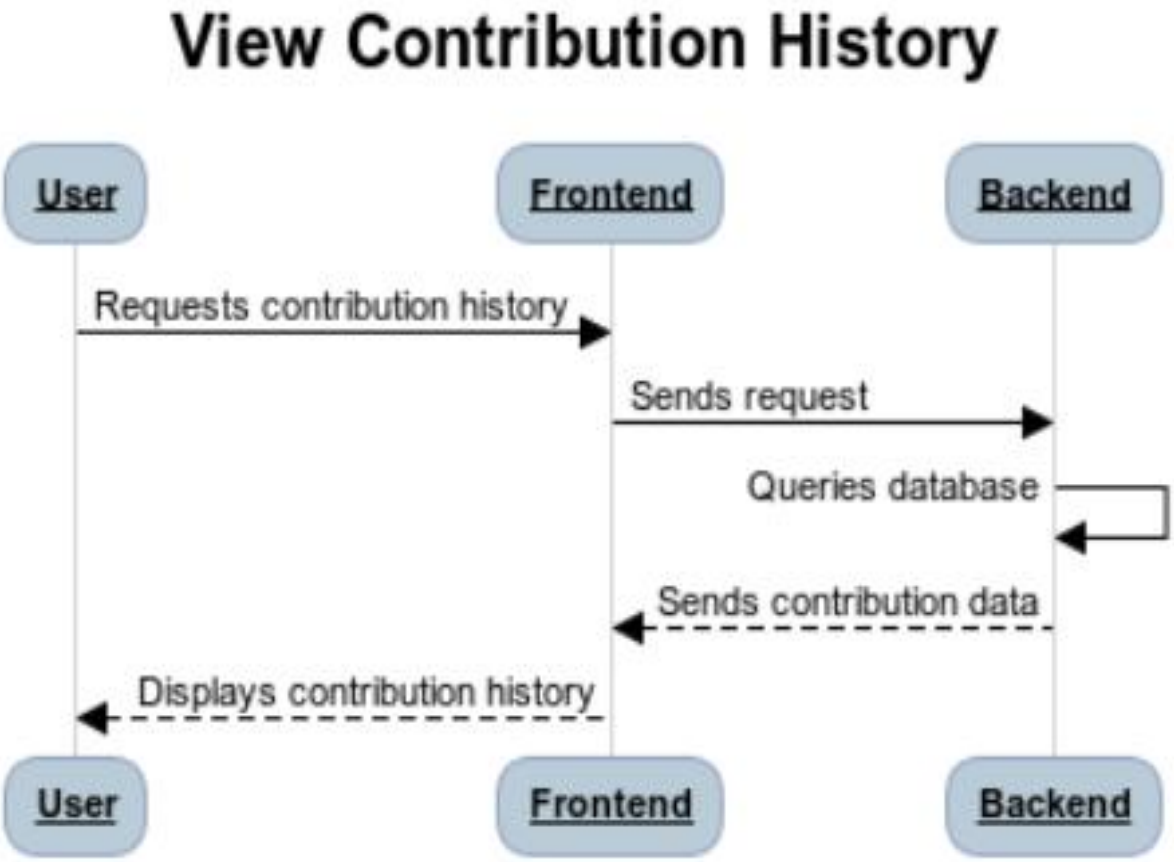
1. User logs in and navigates to the contribution section.
2. User enters the contribution amount.
3. The frontend validates the input (e.g., amount is a valid number).
4. The frontend sends the contribution details to the backend.
5. The backend records the contribution in the database, associating it with the user.
6. The backend sends a confirmation message to the frontend.
7. The frontend updates the user's contribution history.



3.6.3 **View Contribution History**

1. User logs in and navigates to the contribution history section.
2. The frontend sends a request to the backend to retrieve the user's contribution history.
3. The backend queries the database for the user's contributions.

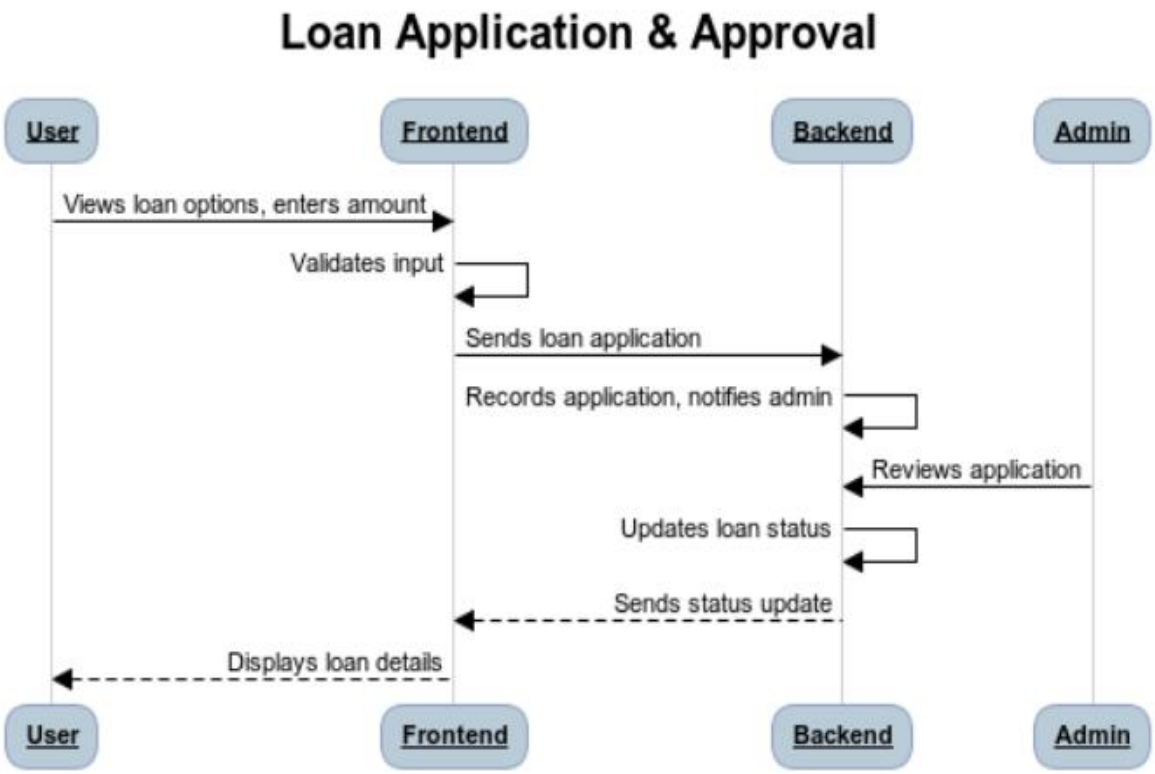
4. The backend sends the contribution data back to the frontend.



5. The frontend displays the contribution history to the user.

3.6.4 **Loan Application & Approval**

1. User logs in and navigates to the loan application section.
2. User views available loan options and terms (interest rates, loan terms).
3. User selects a loan option and enters the desired loan amount.
4. The frontend validates the input.
5. The frontend sends the loan application details to the backend.
6. The backend records the loan application with a "Pending" status.
7. The backend notifies the administrator(s) about the new loan application (e.g., email notification).
8. Administrator logs in and reviews the loan application.
9. Administrator approves or rejects the loan application.
10. The backend updates the loan application status in the database.
11. The backend notifies the user about the loan application status (e.g., email notification).
12. If approved, the loan details are displayed in the user's dashboard.

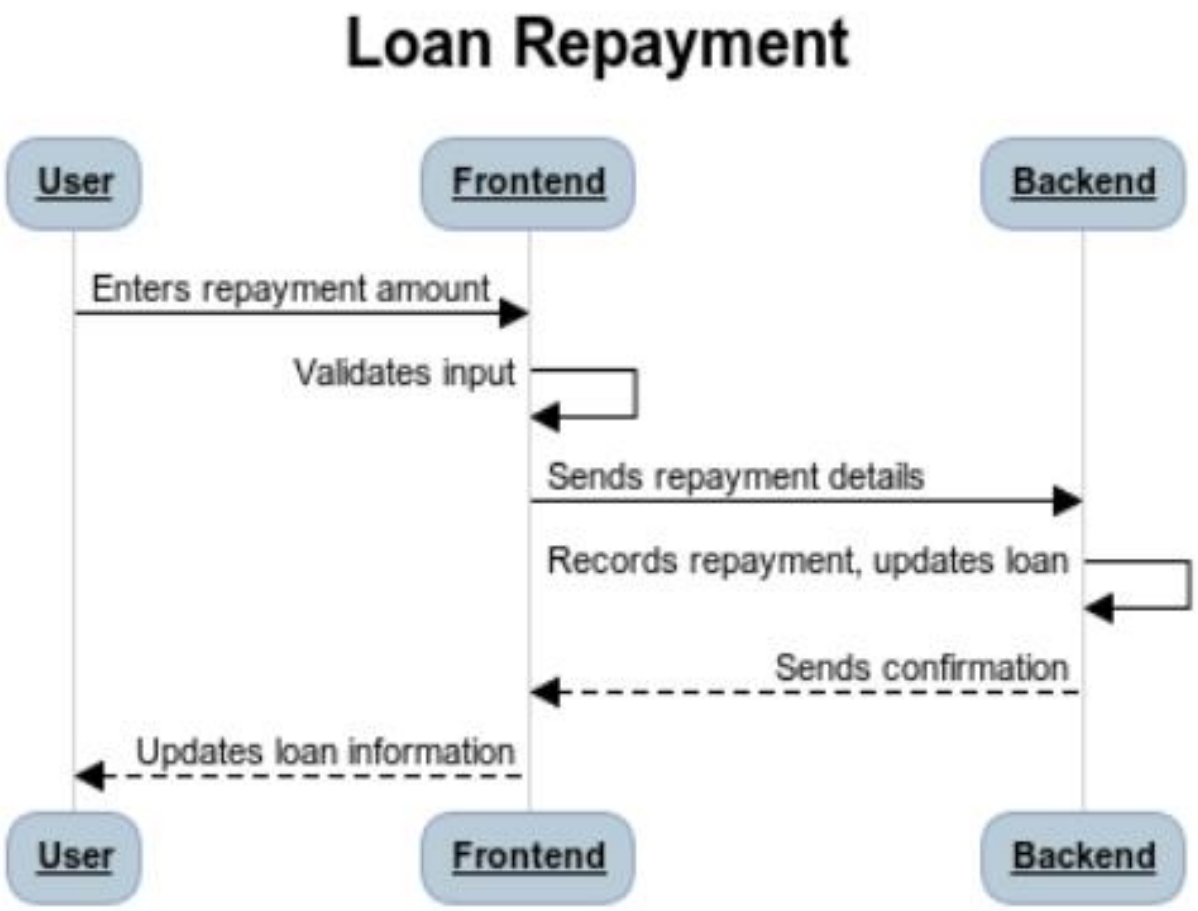


3.6.5 **Loan Repayment**

1. User logs in and navigates to the loan repayment section.
2. User views their loan details and repayment schedule.
3. User enters the repayment amount.
4. The frontend validates the input.
5. The frontend sends the repayment details to the backend.
6. The backend records the repayment in the database, updating the loan balance and repayment schedule.
7. The backend updates the loan status if the loan is fully repaid.

8. The backend sends a confirmation message to the frontend.

9. The frontend updates the user's loan information.



3.7 SECURITY CONSIDERATIONS

Security is a paramount concern for the Auchi Polytechnic Registry Welfare Scheme Web Application, given the sensitive nature of financial data. The following security measures will be implemented:

- **Password Hashing:** User passwords will never be stored in plain text. Instead, they will be hashed using bcrypt, a strong and widely-used password hashing algorithm. This ensures that even if the database is compromised, the actual passwords remain protected.
- **Input Validation:** All user input, both on the frontend and backend, will be rigorously validated to prevent common web vulnerabilities such as SQL injection and cross-site scripting (XSS) attacks. Input validation will check for data types, lengths, formats, and sanitize input to remove potentially malicious code.
- **Role-Based Access Control (RBAC):** Access to different parts of the application will be controlled based on user roles (member or administrator). Administrators will have elevated privileges to manage users, loans, and financial data, while members will only have access to their own information and allowed functionalities.
- **Secure API Endpoints:** All API endpoints will be protected using appropriate authentication and authorization mechanisms. JWT (JSON Web Tokens) may be used for secure session management. This ensures that only authenticated and authorized users can access the API.
- **Data Encryption (Consideration):** While not strictly required for all data, consider encrypting sensitive data at rest (in the database) and in transit (using HTTPS). This adds an extra layer of security.
- **Regular Security Audits:** Periodic security audits and penetration testing should be conducted to identify and address any potential vulnerabilities.

- **Protection Against Brute-Force Attacks:** Implement measures such as rate limiting and account lockout mechanisms to prevent brute-force password guessing attacks.
- **HTTPS:** All communication between the user's browser and the server will be encrypted using HTTPS to protect data in transit.

3.8 TECHNOLOGY STACK JUSTIFICATION

The chosen technology stack has been carefully selected to meet the requirements of the Auchi Polytechnic Registry Welfare Scheme Web Application:

- **Frontend:**
  - **HTML, CSS, JavaScript:** These are fundamental web technologies that provide the foundation for building interactive and responsive user interfaces. They are widely supported across all browsers and devices. A modern JavaScript framework (React, Vue, or Angular) will be evaluated during the implementation phase to enhance the development process and maintainability of the frontend code.
- **Backend:**
  - **Node.js & Express.js:** Node.js, a JavaScript runtime environment, offers high performance and scalability, making it ideal for building the backend API. Express.js, a popular Node.js web framework, simplifies the development of RESTful APIs and provides essential features such as routing, middleware, and request handling.

- **Database:**
    - **MongoDB:** MongoDB, a NoSQL database, offers flexibility in schema design, which is beneficial for handling the diverse data requirements of the application. Its scalability and performance make it suitable for storing and retrieving user data, financial records, and transaction logs.

- **Security:**
    - **bcrypt:** bcrypt is a widely recognized and secure password hashing algorithm that will be used to protect user passwords.
    - **JWT (Consideration):** JSON Web Tokens offer a robust and standardized way to manage user sessions and implement authentication and authorization.

3.9 CONCLUSION

This chapter has provided a detailed analysis and design for the Auchi Polytechnic Registry Welfare Scheme Web Application. It has outlined the system's requirements, architectural design, database structure, user interaction flows, and security considerations. This comprehensive design serves as a solid foundation for the implementation phase, which will be discussed in the next chapter. The chosen technology stack has been justified based on its suitability for meeting the project's requirements and ensuring a secure, scalable, and user-friendly system.

CHAPTER FOUR

IMPLEMENTATION

4.1 INTRODUCTION

The implementation phase of the Auchi Polytechnic Registry Welfare Scheme Web Application focuses on translating the conceptual system design into a functional system. This chapter provides an in-depth discussion of the tools and technologies used, the development of system modules and features, integration between the frontend and backend, database interactions, security measures, the testing process, and the challenges encountered during development. Each section will outline how the system was constructed step-by-step, ensuring that the goals and objectives of the project were met.

4.2 DEVELOPMENT TOOLS AND TECHNOLOGIES USED

To successfully implement the Auchi Polytechnic Registry Welfare Scheme Web Application, a range of modern and reliable development tools and technologies were employed. These tools not only facilitated the development process but also ensured that the system is scalable, secure, and user-friendly.

Frontend Technologies

The frontend of the application, responsible for the user interface (UI), was built using core web technologies, along with frameworks and libraries to enhance the development process and ensure the system is responsive and engaging for users.

- **HTML, CSS, and JavaScript**: These are the foundational languages for building the frontend. HTML provided the structure for the web pages, while CSS was responsible for styling, including layout and color schemes. JavaScript added dynamic behavior to the web pages, such as form validation, user interactions, and real-time updates without needing to reload the page.
- **Bootstrap**: Bootstrap, a front-end framework, was chosen for its ability to create responsive designs quickly. It ensures that the application provides an optimal viewing experience on devices of various sizes, including mobile phones, tablets, and desktops. Bootstrap's prebuilt components, such as navigation bars, modals, and forms, saved development time and provided a consistent and modern UI.
- **React.js (Optional)**: Although the initial implementation did not include React.js, it was evaluated for future enhancements. React's component-based architecture and virtual DOM allow for efficient rendering and state management, which would be useful if the application needed to scale in complexity over time.

Backend Technologies

The backend of the system was implemented to manage the server-side logic, API endpoints, and database interactions. Given its lightweight nature and ease of use with JavaScript, Node.js was the ideal choice for this application.

- **Node.js**: Node.js was selected as the runtime environment for the backend because of its event-driven, non-blocking I/O model. This allows the server to handle multiple requests concurrently without performance degradation, making it a highly scalable solution for applications with a high volume of users.

- **Express.js**: Express.js, a web application framework built on top of Node.js, simplified the routing and middleware handling process. Express.js allowed for the easy creation of RESTful API endpoints, which were crucial for handling client requests related to user registration, login, loan management, and contribution tracking.

Database

The choice of database plays a significant role in the performance and scalability of an application. MongoDB, a NoSQL database, was selected due to its flexible schema and ability to handle large amounts of data efficiently.

- **MongoDB**: MongoDB was chosen as the database solution because it stores data in a flexible, JSON-like format, which is perfect for an application with dynamic data, such as user profiles, contributions, and loan records. The database is designed to scale horizontally, allowing for easy expansion as the number of users and records grows over time.
- **Mongoose**: Mongoose, an Object Data Modeling (ODM) library for MongoDB, was used to define the schema for each collection (e.g., users, contributions, loans). Mongoose facilitates schema-based data modeling, validation, and querying, making it easier to interact with the database.

Security Technologies

The security of the application was a top priority during implementation. Several libraries and techniques were integrated into the backend to ensure that user data, especially sensitive information like passwords, was adequately protected.

- **bcrypt**: bcrypt was used to hash passwords before they are stored in the database. This ensured that even if the database were compromised, the stored passwords could not be easily retrieved.
- **JWT (JSON Web Tokens)**: JWT was used for managing user authentication and authorization. After a successful login, a JWT token is issued, which the user must include in the header of subsequent requests. This token ensures that only authorized users can access protected resources.
- **Helmet.js**: To enhance the security of HTTP headers, Helmet.js was used to set various HTTP headers, such as content security policies, to protect the application from common web vulnerabilities, including cross-site scripting (XSS) and clickjacking attacks.

4.3 SYSTEM MODULES AND FEATURES

The web application is divided into multiple modules, each designed to handle a specific functionality. These modules are designed to meet the requirements of both the administrative staff and users (members of the welfare scheme). Below is a detailed discussion of the modules and their features.

User Management Module

The User Management module is responsible for managing user accounts, including registration, login, and profile management. The module ensures that users can securely access their accounts and make contributions to the welfare scheme.

- **Registration**: Users can register by providing basic personal details such as name, email, and password. The system validates these details and hashes the password using bcrypt before saving it to the database.
- **Login**: After registration, users can log in using their email and password. The system compares the provided password with the hashed version stored in the database. If they match, a JWT token is generated and sent to the user, allowing them to access protected resources.
- **Profile Management**: Users can update their profiles, including changing their password, contact details, and other personal information.

Contribution Management Module

This module allows users to make contributions to the welfare scheme and view their contribution history. It ensures that each contribution is logged accurately and associated with the correct user.

- **Contribution Recording**: When a user makes a contribution, the system records the amount, date, and type of contribution (monthly, special). This information is stored in the MongoDB database and associated with the user's account.
- **Contribution History**: Users can view their past contributions, including the total amount contributed, the date of each contribution, and the contribution type. This history is accessible from the user dashboard.

Loan Management Module

The Loan Management module allows users to apply for loans, track their loan applications, and view repayment schedules. Administrators can approve or reject loan applications based on predefined criteria.

- **Loan Application**: Users can apply for loans by filling out an application form that includes details such as the loan amount, term, and purpose. The application is then sent to the administrator for approval.
- **Loan Approval and Tracking**: Administrators review loan applications and either approve or reject them. Upon approval, the user is notified, and the loan details are stored in the database, including the repayment schedule and interest rate.
- **Repayment Tracking**: The system allows users to track their loan repayment schedules. Users can make payments, which are recorded and updated in the system. The remaining balance is automatically adjusted as payments are made.

Administrative Panel

The administrative panel is designed for the system administrators to manage users, view reports, and handle system maintenance tasks.

- **User Management**: Administrators can manage user accounts by viewing a list of all users, deactivating inactive accounts, and updating user details.
- **Loan Management**: Administrators can approve or reject loan applications, view loan repayment statuses, and track the financial health of the scheme.
- **Reports and Analytics**: The system provides real-time reports on user activity, loan status, and contributions, allowing administrators to make data-driven decisions.

4.4 BACKEND IMPLEMENTATION

The backend of the application was developed using Node.js and Express.js. The server-side logic was responsible for handling HTTP requests, managing user authentication, interacting with the database, and processing business logic.

API Endpoints

The backend was built around RESTful principles, with each resource (e.g., users, loans, contributions) having its corresponding API endpoint. These endpoints allowed the frontend to interact with the backend and perform CRUD (Create, Read, Update, Delete) operations on the data.

For example, the user registration endpoint:

js

CopyEdit

```
app.post('/api/register', async (req, res) => {
   const { name, email, password } = req.body;
   const hashedPassword = await bcrypt.hash(password, 10);
   const newUser = new User({ name, email, password: hashedPassword });
   await newUser.save();
   res.json({ message: 'User registered successfully' });
});
```

Session Management

JWT was used for session management. Upon successful login, a JWT token was generated and sent to the user. This token is stored in the client's browser and included in subsequent API requests to authenticate and authorize the user.

js

CopyEdit

```
const token = jwt.sign({ userId: user._id }, 'secret_key');
res.json({ token });
```

Error Handling and Middleware

Error handling middleware was implemented to capture errors that may arise during API requests. This middleware catches errors, logs them, and sends appropriate responses to the client.

js

CopyEdit

```
app.use((err, req, res, next) => {
  console.error(err);
  res.status(500).send('Something went wrong!');
});
```

4.5 FRONTEND IMPLEMENTATION

The frontend was designed to be user-friendly and responsive. It involved the development of various pages, including registration, login, user dashboard, loan application, and contribution tracking.

Responsive Design

Bootstrap's grid system was used to create a flexible and responsive layout. The frontend adjusts its layout depending on the device's screen size, ensuring that the application is accessible on mobile phones, tablets, and desktops.

User Interaction

JavaScript was used to handle user interactions, such as form submissions, data validation, and asynchronous updates. For instance, when a user submits a loan application, the form sends data to the backend via an AJAX request, and the user is notified of the application's success or failure without refreshing the page.

js

CopyEdit

```
$('#loan-form').submit(function(event) {
    event.preventDefault();
    $.ajax({
        url: '/api/loan/apply',
        method: 'POST',
        data: $('#loan-form').serialize(),
        success: function(response) {
```

```
            alert('Loan application submitted successfully!');
        }
    });
});
```

4.6 DATABASE INTEGRATION

MongoDB was chosen as the database for the system due to its scalability and flexibility. The system used Mongoose to define schemas for each data type (users, contributions, loans).

Data Models

Each collection in MongoDB was defined with a Mongoose schema, specifying the fields and their data types. For instance, the User schema contains fields like name, email, and password, while the Loan schema includes fields such as amount, interestRate, and repaymentSchedule.

js

CopyEdit

```
const userSchema = new mongoose.Schema({
    name: { type: String, required: true },
    email: { type: String, required: true, unique: true },
    password: { type: String, required: true }
});
```

Database Queries

The backend interacts with the database using Mongoose methods. For example, when retrieving a user's contribution history, the following query is used:

js

CopyEdit

```
User.findById(userId).populate('contributions').exec((err, user) => {
  if (err) {
    return res.status(500).send('Database query failed');
  }
  res.json(user.contributions);
});
```

4.7 SECURITY MEASURES IMPLEMENTED

Security was a major focus during the implementation. Several best practices were followed to safeguard user data and prevent unauthorized access.

- **Password Hashing (bcrypt with Salt):** Password hashing is a fundamental security practice to protect user credentials. Instead of storing passwords in plain text, which would be disastrous if the database were compromised, we use a one-way cryptographic function to transform the password into an irreversible hash.For this application, we employed bcrypt, a widely recognized and robust password hashing algorithm. bcrypt is specifically designed to be resistant to brute-force attacks, even with the use of powerful GPUs or specialized cracking hardware. Its strength lies in its adaptive nature and the

incorporation of a "salt." A crucial aspect of our bcrypt implementation is the use of a "salt." A salt is a randomly generated string that is unique to each password. It's concatenated with the password *before* the hashing process. This ensures that even if two users have the same password, their hashes will be different due to the unique salt. Salting also makes it much harder for attackers to use pre-computed "rainbow tables" to crack hashes.

- JWT (JSON Web Tokens) were used for authorization. After a user successfully logged in, the server generated a JWT containing the user's ID and other relevant information. This token was then sent to the client and stored in local storage or a secure cookie. For every subsequent request to a protected resource, the client included the JWT in the Authorization header. The server then verified the JWT's signature and payload to ensure its authenticity and authorize the user's access. A secret key, stored securely on the server, was used to sign the JWTs. A strategy was implemented for token expiration and renewal to enhance security. This included setting an appropriate expiration time for the JWTs and providing a mechanism for users to refresh their tokens before they expire.

Secure Headers

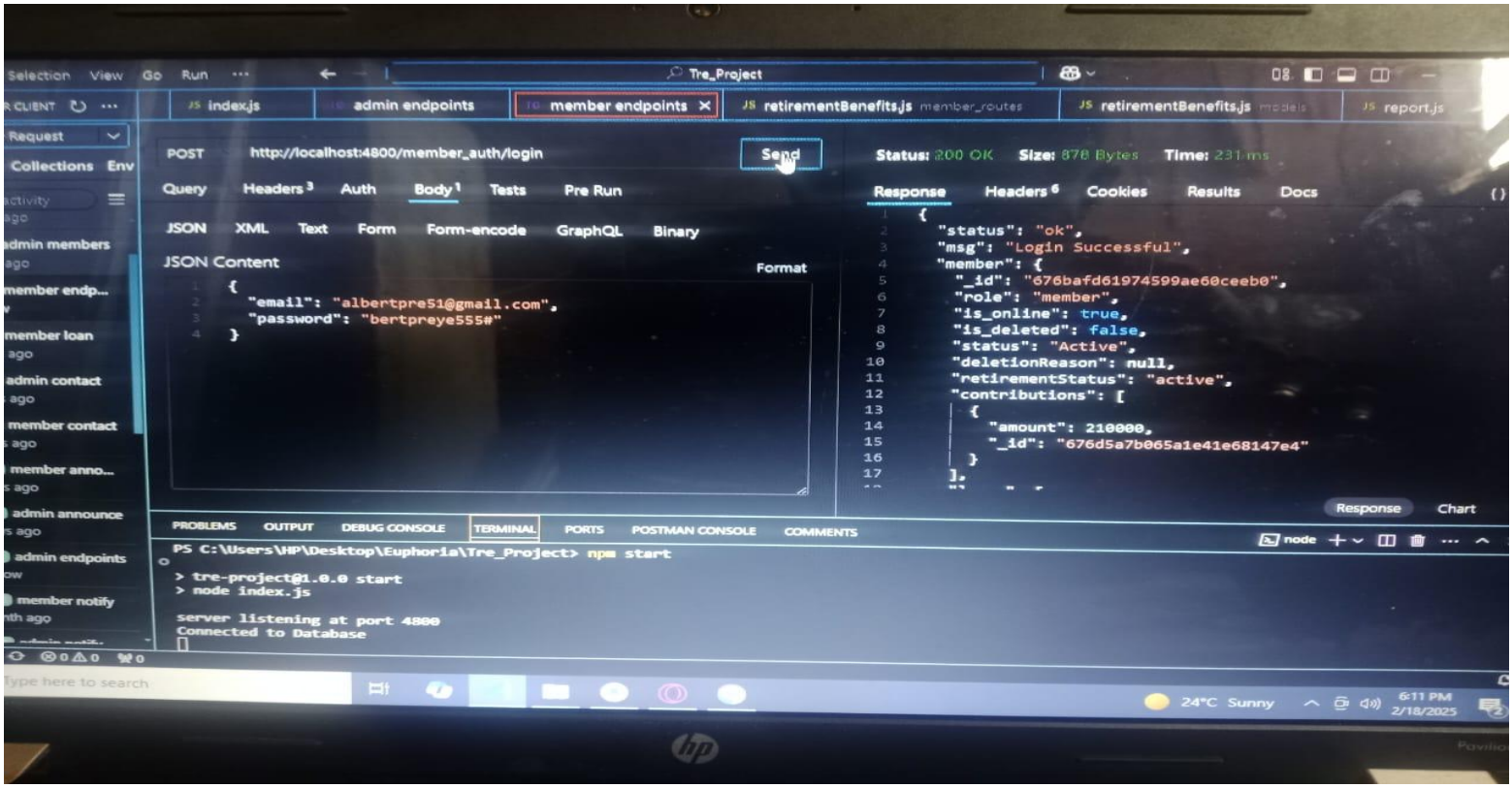
Helmet.js was used to secure HTTP headers and prevent common attacks such as cross-site scripting (XSS) and cross-site request forgery (CSRF).

4.8 TESTING AND DEBUGGING PROCESS

The development process was accompanied by extensive testing to ensure the system was working as expected.

Unit Testing

Jest was used to write unit tests for backend functionality, ensuring that each function performed



its intended task.

Thunder client was used to test the different API end points

Manual Testing

Manual testing was performed on the frontend to validate the user interface, form submissions, and responsiveness.Manual testing of the frontend was a crucial part of the quality assurance process. It focused on validating the user interface (UI), ensuring correct form submissions, and verifying the responsiveness of the application across different devices and browsers. This involved a systematic approach to testing various aspects of the frontend, simulating real-world user interactions, and documenting any issues or bugs.

4.9 CHALLENGES FACED AND SOLUTIONS

During the implementation phase, several challenges were encountered, including issues with database integration, security, and frontend responsiveness.

Database Integration

There were challenges in linking user contributions to the correct user records. This was solved by ensuring proper referencing of user IDs in the MongoDB collections.

Security Issues

There were concerns regarding the security of sensitive data. These were addressed by using bcrypt for password hashing, JWT for session management, and Helmet.js for securing HTTP headers.

4.10 CONCLUSION

The implementation of the Auchi Polytechnic Registry Welfare Scheme Web Application was completed using a modern technology stack, including Node.js, Express.js, MongoDB, html,css and javascript . The system modules were successfully implemented, providing functionality such as user management, contribution tracking, loan management, and administrative controls. The application is secure, scalable, and user-friendly, and is now ready for deployment and testing in a live environment. The next chapter will explore the testing and deployment of the system.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

The development of the Financial Management System (FMS) for the Auchi Polytechnic Registry Welfare Scheme was aimed at overcoming inefficiencies and inconveniences caused by manual financial processes. The system has been designed to improve operations, enhance transparency, and improve levels of user satisfaction among administrators and members.

A structured approach was adopted, beginning with a thorough analysis of the existing system, followed by the design and implementation of a new web-based system. The system was built on Node.js and Express.js for the backend to ensure scalability and robust performance. MongoDB was employed as the database because of its flexibility and ability to support various data requirements, such as member contributions, loans, and retirement benefits. Security features like bcrypt for password hashing and JWT for authentication were implemented to protect valuable financial data.

The system was rigorously tested to satisfy the functionality, usability, and security requirements. The core functionalities such as contribution tracking, loan management, and retirement benefit calculation were included, and members were able to view a clear and user-friendly platform for managing their money.

5.2 LIMITATIONS AND CHALLENGES

While the system fulfills its primary objectives, some limitations and challenges were experienced during development and testing:

1. **Technology Infrastructure:**

- It depends on the existing technology infrastructure in Auchi Polytechnic. Poor internet and outdated hardware may hinder its effectiveness.

2. **Budget Constraints:**

- Lack of adequate funding for software licenses, hardware upgrades, and technical support can restrict the scalability and long-term maintenance of the system.

3. **User Adoption:**

- Some members and administrators will be resistant to giving up manual procedures in favor of a computer system unless they are computer literate. Comprehensive training schemes will be needed for a smooth implementation.

4. **Data Security Concerns:**

- Since the system has high-security measures, conformity with data protection legislation and shielding from cyber-attacks will still be an issue, particularly under limited resources.

5. **Scalability:**

- The system is optimized to suit the current base of membership. But since the welfare plan expands, there could be the need for database tuning and balancing load to manage higher traffic and transactions.

5.3 **RECOMMENDATIONS**

To enhance the usability, security, and scalability of the financial management system, the following is recommended:

1. **Enhance Security Measures:**

- Implement multi-factor authentication (MFA) to provide an additional layer of security for accounts.
- Conduct security audits and penetration testing on a periodic basis to find and repair vulnerabilities.

2. **Improve Scalability**

- Enhance database queries and implement caching systems to handle more transactions in a more efficient way.
- Consider cloud hosting solutions to ensure they are scalable and stable when the membership pool grows.

3. **User Training and Support:**

- Create extensive training modules for members and administrators to ensure easy implementation of the system.
- Offer regular technical support to solve user issues and support ongoing development.

4. **Advanced Features:**

- Implement real-time notifications through SMS or mobile devices to inform members about their contributions, loan status, and benefits.
- Add a file upload feature for users to upload supporting documents during loan applications.

5. **Monitoring and Maintenance:**

- Employ error logging and monitoring tools (e.g., Sentry, LogRocket) to detect and repair system issues on time.
- Collect user feedback on a regular basis to identify areas for improvement and have the system adjust to user needs.

5.4 CONCLUSION

The effective design and implementation of the Financial Management System for the Auchi Polytechnic Registry Welfare Scheme demonstrate how emerging web technologies can transform financial operations. Utilizing **Node.js**, **Express.js**, and **MongoDB**, the system provides a secure, scalable, and user-friendly platform for handling contributions, loans, and retirement benefits.

Despite the issues encountered, the project was a success in its overall objective of improving transparency, efficiency, and user satisfaction. There are certain limitations, however, which indicate where future development can take place. Enhancing security, scalability, and user take-up will be critical to ensuring the system's long-term success.

In the coming times, incorporating newer functionalities such as real-time communication, AI-based automation, and mobile app integration will further enhance the system to be more effective. Updates and refinements on a regular basis will be needed to maximize performance and security as the welfare scheme expands.

In conclusion, the project provides a sound foundation for a modern financial management system tailored to the needs of welfare schemes. By addressing the weaknesses recognized and implementing the recommended enhancements, the system can evolve into a comprehensive and highly efficient solution for financial management in such organizations.

REFERENCES

Amare, T. Alemu. (2024). Mobile Money Systems in Sub-Saharan Africa: Enhancing Accessibility and Transparency.

Anjan, A. Kaikini. (2020). Digital Financial Platforms in Cooperative Societies: A Case Study from India.

Brown, E. (2019). Web Development with Node and Express. O'Reilly Media.

Bushra, A. (2024). The Role of Digital Platforms in Enhancing Financial Literacy and Wellbeing. Journal of Financial Literacy and Wellbeing.

Dauda, Rilwan. (2021). Web-Based Management Systems for Welfare Schemes: A Nigerian Case Study.

Flanagan, D. (2020). JavaScript: The Definitive Guide. O'Reilly Media.

International Labour Organization (ILO). (2023).

Digital Solutions for Financial Transparency in Resource-Constrained Environments.

Sutton, I. (2017). Security Risk Assessment and Management: Protecting Business Information Systems.

W3C. (2023). Web Accessibility Guidelines (WCAG).

Zendesk. (2023). What Is a Ticketing System? How It Helps in Customer Support.

APPENDIX

Server Configuration and Routing Code:
This section contains the server configuration and routing code for the application. The code sets up an Express server, connects to a MongoDB database, and defines routes for both member and admin functionalities.

```javascript
const express = require('express');
const app = express();

const dotenv = require('dotenv');
dotenv.config();

const mongoose = require('mongoose');

// Connect to the database
mongoose.connect(process.env.MONGO_URI);
const con = mongoose.connection;
con.on('open', error => {
    if (error) {
        console.log(`Error connecting to database: ${error}`);
    } else {
        console.log("Connected to Database");
    }
});

// Middleware for parsing JSON and URL-encoded data
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

// Member routes
app.use('/member_auth', require('./member_routes/auth'));
app.use('/member_profile', require('./member_routes/profile'));
app.use('/member_loan', require('./member_routes/loan'));
app.use('/member_contributions', require('./member_routes/contributions'));
app.use('/member_contact', require('./member_routes/contact'));
app.use('/member_announcements', require('./member_routes/announcements'));
app.use('/member_notifications', require('./member_routes/notifications'));
```

```
app.use('/member_retirementBenefits', require('./member_routes/retirementBenefits'));

// Admin routes
app.use('/admin_auth', require('./admin_routes/auth'));
app.use('/admin_profile', require('./admin_routes/profile'));
app.use('/admin_announcements', require('./admin_routes/announcements'));
app.use('/admin_contributions', require('./admin_routes/contributions'));
app.use('/admin_expenses', require('./admin_routes/expenses'));
app.use('/admin_loan', require('./admin_routes/loan'));
app.use('/admin_retirementBenefits', require('./admin_routes/retirementBenefits'));
app.use('/admin_contact', require('./admin_routes/contact'));
app.use('/admin_members', require('./admin_routes/members'));
app.use('/admin_notifications', require('./admin_routes/notifications'));
app.use('/admin_report', require('./admin_routes/report'));

// Start the server
const port = process.env.PORT;
app.listen(port, () => {
   console.log(`Server listening at port ${port}`);
});

module.exports = app;
```

Key Components

Database Connection: The application establishes a connection to a MongoDB database via the connection string in the environment variable `MONGO_URI`.

Middleware: The server uses `express.json()` and `express.urlencoded()` middleware to parse the incoming requests with JSON and URL-encoded bodies.

Routing: The application defines routes for members as well as admins with separate route handlers for each functionality such as authentication, handling profiles, loans, contributions, etc.

Server Initialization: The server is listening on the port from the environment variable `PORT`.