

**BACHELOR’S DEGREE IN INFORMATION TECHNOLOGY**

------------------------------------------------------------------------------------

**FINAL YEAR PROJECT REPORT**

------------------------------------------------------------------------------------

**STUDENT NAME : KATONGO BUPE**

**STUDENT NUMBER : 2010100**

**SUPERVISOR : MR. C. SILWIZYA**

------------------------------------------------------------------------------------

**PROJECT TITLE:**

**DESIGNING AND DEVELOPING A CENTRALIZED LOAN APPLICATION MANAGEMENT SYSTEM FOR MICRO-FINANCE COMPANIES**

**Designing And Developing A Centralized Loan Application Management System For Micro-Finance Companies**

**By**

**Katongo Bupe | 2010100**

A Dissertation Submitted to Zambia University College of Technology in Partial fulfilment for the award of Bachelor of Information Technology

**ZAMBIA UNIVERSITY COLLEGE OF TECHNOLOGY**

**NDOLA**

**2023**

# NOTICE OF COPYRIGHT

I do hereby declare that the content of this document is my own and all other works by other people have been only referenced and that this work has not been previously presented to another University for the same purpose.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Katongo Bupe Date

(Candidate)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mr. Silwizya C Date

(Supervisor)

# DEDICATION

I dedicate this dissertation to my family who made it possible for me to be in college and supported me throughout. I also dedicate this document to the Zambia University College of Technology management for ensuring I had adequate access to learning resources and an excellent learning environment.



**Designing and Developing a Centralized Loan Application Management System for Micro-Finance Companies**

# CERTIFICATE OF APPROVAL

This dissertation by Katongo Bupe entitled, ‘**Designing And Developing A Centralized Loan Application Management System For Micro-Finance Companies**’ has been approved as a fulfilling requirement for the award of a Bachelor’s Degree in Information Technology.

**Examiners Name**

Jonathan Koloko (Mr) Date

........................................................ .........................................

Catherine Mwape (Ms) Date

........................................................ .........................................

Chairperson of the Board of Examiners Date

........................................................ .........................................

# ACKNOWLEDGEMENT

I would like to express my greatest appreciation to all the individuals who have helped and supported us throughout the project. I am thankful to my lectures for their ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project. I would also like to thank Mr. Silwizya who was always there to guide me and provide the necessary assistance.

I wish to thank my parents and guardians for their undivided financial support and interest who inspired me and encouraged me to go all the way, without whom I would not have been able to complete this project.

# Abstract

The concept of microfinance, involving the provision of small loans to underprivileged individuals and businesses, has emerged as a potent strategy for poverty alleviation[1]. This research endeavors to address the challenge of poverty reduction by establishing a system that offers secure and unsecured loans to individuals, thereby granting them a pathway to economic empowerment. Microfinances are particularly advantageous in developing countries such as Zambia, where microfinance institutions (MFIs) have become the primary funding source for microbusinesses, thus ensuring sustainable access to financial resources. However, a critical problem arises from the lack of a centralized platform for comparing interest rates and services across various MFIs, leading to inefficiencies in the loan application process.

The primary objective of this study is to develop a Centralized Loan Application Management System that harmonizes multiple microfinance institutions on a singular platform. Additionally, the project aims to create a Cross-Platform Centralized Loan Application Management System, automating the Loan Application Process. Moreover, the research endeavors to devise a Centralized Loan Application Management System that effectively mitigates the risk of loan defaults experienced by microfinance institutions.

This study tackles the question of how a Centralized Loan Application Management System can be created to accommodate multiple microfinance institutions on a single platform. It also delves into the development of a Cross-Platform Centralized System and explores strategies to reduce the occurrence of loan defaults within microfinance institutions. The proposed system will offer essential functionalities including login/registration, search, view, update, and application submission, presented through a user-friendly interface on both iOS and Android platforms. The application will be built using React Native and Node.js for the frontend and backend respectively, with Mongo DB serving as the database management system.

# Chapter One - Introduction

## 1.1 Introduction

The concept of "microfinance" refers to the lending of small sums of money to under-privileged business people and ordinary people, this study intends to develop a system for reducing poverty by giving the individuals access secured and unsecured loans. [2]

Microfinance has many advantages for developing countries like Zambia. This particular type of lending has been around for a while. In Africa and other developing nations, Microfinance institutions (MFIs) have become the primary source of funding for microbusinesses in many developing countries, providing sustainable access to financial resources.[2]

In the modern day and age, internet has become essential to everyone. This project focuses on the creation of a mobile application for an online centralized platform for microfinance Institutions.

## 1.1.1 Background of the Study

The concept of microfinance has gained prominence as a powerful approach for tackling poverty by providing small loans to disadvantaged individuals and businesses. This strategy aims to empower recipients economically and contribute to poverty alleviation. Microfinance institutions (MFIs) have become crucial in developing countries, including Zambia, where they serve as primary sources of funding for microbusinesses. However, the lack of a centralized platform for comparing services and interest rates across different MFIs has led to inefficiencies in the loan application process. These institutions play a pivotal role in ensuring consistent access to financial resources for microbusinesses, contributing to sustainable economic growth. Despite these benefits, challenges persist, such as the risk of loan defaults experienced by MFIs.

## 1.2Aim

This study aims at designing and developing an effective Centralized Loan Application Management System that will centralize various MFIs, alleviating the challenge of comparing interest rates, digitizing the Loan Application process and automating payment reminders and Credit Scoring to reduce the risk of loan defaults experienced by microfinance institutions. By achieving these objectives, the study seeks to contribute significantly to the enhancement of operational efficiency, user experience, and risk management within the microfinance landscape.

## 1.3 Problem Statement

The inflexibility of comparing interest rates and services between different microfinance Institutions becomes a challenge. The challenges of accessing microfinance in rural areas in Zambia[3]. The study found that one of the challenges is the inflexibility of comparing interest rates and services between different microfinance institutions. The study also found that borrowers in rural areas often have to travel long distances to access microfinance services. Additionally, Borrowers having access to the internet have to look up every microfinance company on the internet, this is time consuming and inefficient.

The use of physical loan application forms in microfinance in Zambia. The study found that physical loan application forms are still widely used in Zambia, despite the availability of digital alternatives[4]. The study also found that physical loan application forms can be time-consuming and inefficient, and can also lead to errors. A client/debtor has to submit hard copies of NRC and other required documents to a microfinance institution for a Loan application to be processed.

Handling loan defaults is a massive challenge most microfinance institutions face [5].This study examines how local microfinance institutions in Zambia handle loan defaults. The study found that local microfinance institutions use a variety of methods to manage loan defaults, including: Credit scoring, where borrowers are given a score based on their credit. This score is used to determine the borrower's risk of defaulting on a loan. Personal visits, Lender may visit borrowers to discuss their loan repayment status, and Phone calls, Lenders may call borrowers to remind them of their loan repayment due dates.

## 1.4 Research Objectives

1. To develop a centralized loan application management system that shall allow borrowers to compare interest rates between different microfinance institutions.
2. To implement a cross platform mobile app that shall Digitize the Loan Application Process.
3. To develop a system that shall automate payment reminders and Credit Scoring in order reduce the risks of loan defaults experienced by micro-finance institutions.

## 1.6 Research Questions

While the research was being undertaken, a mirror of research questions has been identified using the objectives specified in the proposal.

1. How can a centralized loan application management system mitigate inflexibility of comparing interest rates between different microfinance institutions?
2. How can a cross platform mobile app Digitize the Loan Application Process and transition microfinances from traditional ways?
3. How can automated payment reminder and Credit Scoring reduce the risks of loan defaults experienced by micro-finance institutions?

## 1.7 Scope and Limitation

The scope of this study encompasses the design, development, and implementation of a Centralized Loan Application Management System tailored for microfinance institutions (MFIs) in Zambia. It includes the creation of a digital platform facilitating borrowers to conveniently compare interest rates and services offered by different MFIs.

The study involves the development of a user-friendly cross-platform mobile application that streamlines and digitizes the loan application process, encompassing online forms and document uploads, thereby eliminating the need for physical paperwork.

Additionally, the scope covers the design and implementation of automated systems for sending payment reminders to borrowers and assessing credit scores, contributing to enhanced borrower accountability and risk assessment for MFIs.

The system will be equipped with components to login/register, search, view, update and apply on the application while providing a custom user-friendly interface on both platforms (iOS & Android). The android/IOS application will be developed using React Native and Node.js as back-end language and Mongo DB as the database management system.

## 1.8 Significance of the Study

The significance of this study can be understood in the context of the following key points:

1. Streamlining Interest Rate Comparison:

The inability to compare interest rates and services between different microfinance institutions has been identified as a significant challenge. This study's objective of developing a centralized loan application management system that facilitates easy comparison of interest rates will greatly benefit both borrowers and MFIs. By providing a unified platform for borrowers to assess interest rates, they can make more informed decisions about loan options, contributing to increased financial literacy and more competitive lending practices.

1. Digitization of Loan Application Process:

The reliance on physical loan application forms in microfinance processes, despite the availability of digital alternatives, poses inefficiencies and barriers. The implementation of a cross-platform mobile app to digitize the loan application process is a significant step toward reducing administrative burdens, eliminating errors associated with manual data entry, and expediting application processing. This digitization not only enhances the user experience for borrowers but also modernizes the operational landscape of MFIs, aligning them with contemporary technological trends.

1. Automation of Payment Reminders and Credit Scoring:

The challenge of managing loan defaults is a pressing concern for microfinance institutions. By developing a system that automates payment reminders and credit scoring. Automated payment reminders can improve borrower accountability, reducing instances of delinquency, while credit scoring mechanisms enhance the accuracy of risk assessment, enabling MFIs to make more informed lending decisions. This contributes to the financial sustainability of both borrowers and MFIs.

## 1.9 Conceptual Framework

The conceptual framework of this research project is built upon several key theoretical concepts. Firstly, microfinance is acknowledged as a powerful tool for poverty alleviation, involving the provision of small loans to underprivileged individuals and businesses to empower them economically. Secondly, centralization is a central theme, involving the development of a Centralized Loan Application Management System that harmonizes multiple microfinance institutions onto a single platform, streamlining and improving the loan application process.

Additionally, risk mitigation strategies are crucial, as the research aims to address loan defaults through effective risk management techniques that protect the interests of both lenders and borrowers. These interconnected concepts collectively form the foundation of the study's approach and methodology.

The visual representation below encapsulates the core architecture of our Centralized Loan Application Management System, highlighting the integration of three key components or dashboards. These components serve as pivotal interfaces, each catering to a distinct set of stakeholders within the microfinance ecosystem.

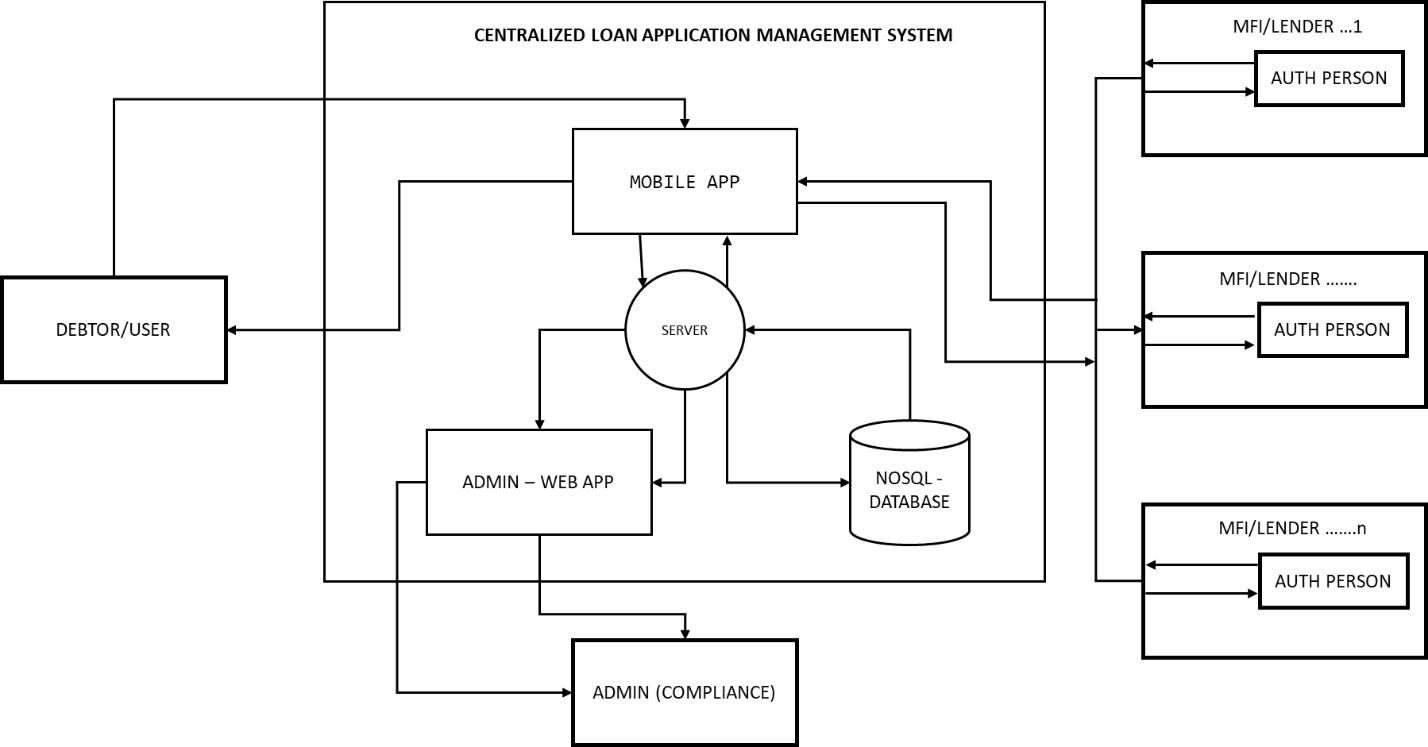


Figure 1: Conceptual Framework

1. User/Debtor Dashboard: The first component of our system is designed with the user or debtor in mind. This user-friendly dashboard provides a secure and intuitive space for individuals seeking financial support. Users can conveniently register, log in, explore loan options, submit applications, and track their loan progress. This component focuses on enhancing the borrower's experience and facilitating their journey towards financial empowerment.
2. Lender/MFI (Microfinance Institution) Dashboard: The second dashboard is dedicated to microfinance institutions (MFIs), the entities responsible for lending and managing financial resources. This component empowers lenders with tools for evaluating loan applications, processing requests, and managing borrower data efficiently. It streamlines MFI operations, ensuring that the lending process is secure, streamlined, and transparent.
3. Admin Dashboard for MFI Eligibility Approval: The third component of our system serves a critical role in maintaining the integrity and eligibility standards of MFIs. The Admin dashboard is tailored for administrators responsible for approving MFI eligibility. This component allows administrators to review andt validate the qualifications of microfinance institutions, ensuring that they meet the necessary criteria to participate in the lending ecosystem. This crucial step guarantees that only credible MFIs are included in the centralized platform.

This diagram symbolizes the core structure of our Centralized Loan Application Management System, depicting how these three components work together harmoniously to create a robust, efficient, and secure environment for all stakeholders. It showcases our commitment to enhancing the microfinance landscape by fostering financial inclusivity, streamlining lending processes, and maintaining the highest standards of eligibility for participating MFIs.

# Chapter Two - Literature Review

## 2.1 Introduction

The literature review presented in this chapter serves as a comprehensive exploration of key aspects related to the centralization of microfinance services, the digitization of loan application processes, and the use of automated credit scoring and payment reminders. The review was conducted in three distinct phases, each aligning with the specific objectives of this research project.

Phase 1 delves into the centralization of microfinance services, a strategy that has gained prominence as an effective means to enhance the accessibility and availability of financial services for those in need. One notable facet of this phase is the utilization of software systems for comparing loan rates across multiple microfinance institutions. This approach has recently drawn significant interest, offering the potential to transform the microfinance landscape by facilitating informed decision-making and empowering underserved individuals.

Phase 2 of our literature review focuses on the burgeoning field of technology-driven digitization of the loan application process in microfinance. Within this context, this study reviewed five articles that provide a comprehensive overview of the current state of the art in digitized loan applications. These articles offer valuable insights into both the benefits and challenges associated with the integration of technology in this crucial aspect of microfinance.

Finally, Phase 3 explores the adoption of automated credit scoring and payment reminders in microfinance, two technological innovations that are reshaping the industry. These advancements aim to enhance the accuracy of assessing borrower creditworthiness and reduce loan defaults, ultimately contributing to the financial sustainability of microfinance institutions and facilitating borrowers' loan repayment obligations.

## 2.2 Review of the Related Works:

The Literature was reviewed in three (3) phases corresponding to the objectives of this project.

1. The centralization of microfinance services has gained prominence as an effective strategy to address the challenges of accessibility and availability faced by the economically disadvantaged population. The utilization of software systems for comparing loan rates across multiple microfinance institutions has garnered significant attention in recent years due to its potential to transform the microfinance landscape. [6]

The imperative need for centralization in microfinance to enhance the accessibility and availability of financial services for impoverished individuals. By centralizing micro financial services under a unified umbrella, this approach has the potential to expedite poverty reduction efforts. [7] The paper underscores the importance of addressing the missing link between lenders and borrowers, particularly within the Indian context, as a pivotal step toward achieving the goals of poverty reduction through microfinance. Centralization emerges as a key strategy in this endeavor, offering the promise of greater efficiency, accessibility, and ultimately, improved livelihoods for the economically disadvantaged.

The paper underscores the crucial role played by centralization in bridging this gap between lenders and borrowers, emphasizing that it is a prerequisite for reducing poverty effectively Centralization not only facilitates the streamlining of microfinance operations but also promotes synergies among various stakeholders, making it easier for borrowers to access credit services.[8]

The impact of high microcredit interest rates on impoverished individuals. While microfinance institutions argue that such rates are necessary for financial sustainability and service expansion, the paper prompts a critical examination of whether the poor are being exploited in the process. [9] This discussion is essential for ensuring that the microfinance sector effectively fulfills its mission of poverty alleviation and empowerment without inadvertently creating new financial vulnerabilities for the very populations it aims to assist. The paper's findings and insights contribute to the ongoing discourse surrounding microcredit interest rates and their implications for the well-being of the economically disadvantaged.

The paper's theoretical framework centers on the evolving landscape of microfinance institutions (MFIs) in developing and transitional economies. Over the past two decades, MFIs have shifted their focus toward achieving financial sustainability by charging interest rates that are sufficiently high to cover their operational costs. This strategic shift is underpinned by the belief that financial sustainability is essential for the long-term viability and expansion of microcredit services.

The paper critically examines the implications of this approach, particularly in terms of its impact on impoverished borrowers. It considers whether these high interest rates inadvertently transform microfinance providers into "new moneylenders" and explores the potential exploitation of low-income clients who may face significant financial burdens due to the cost of credit.

Lastly, The importance of understanding and addressing the variations in microcredit interest rates on a global scale. While small loan sizes are a commonly cited reason for high microcredit interest rates, the paper goes further to investigate the multifaceted factors contributing to these disparities. The research not only raises questions about the impact of interest rate differentials on borrower welfare but also examines the role of competition and public policy in shaping the microcredit landscape. [10]

The research paper explores several fundamental questions that shed light on the variations in microcredit interest rates:

* 1. Comparative Borrower Welfare:

The paper investigates how borrowers fare in both low-interest and high-interest microcredit environments. It seeks to understand the impact of interest rate disparities on the well-being of microcredit clients.

* 1. Competition and Efficiency:

The authors examine the relationship between competition among microfinance institutions (MFIs) and their operational efficiency. They assess whether increased competition necessarily results in lower microcredit interest rates.

* 1. Public Policy and Regulatory Environment:

The paper also delves into the role of public policy in influencing the domestic microfinance sector. It explores the characteristics of an "appropriate" regulatory environment that can foster responsible microcredit practices and ensure the interests of borrowers are protected.

1. The use of technology to digitize the loan application process in microfinance is a rapidly growing field. The five articles reviewed here provide an overview of the current state of the art, as well as insights into the potential benefits and challenges of this approach.

The research paper by [4] underscores the need for a nuanced approach to digitizing microfinance. While digitization offers significant benefits, it also poses challenges, particularly in preserving the essential "human face" of microfinance institutions. The paper highlights the importance of maintaining personal relationships and trust between loan officers and clients, even as digital technologies are integrated into microfinance operations.

The research paper draws on questionnaires and semi-structured interviews with managers and loan officers from four microfinance institutions in Zambia to shed light on the complexities of digitizing the lending process. Several key findings and discussions emerge:

The Dilemma of Digitization: The paper addresses the dilemma of digitization in microfinance, particularly in mature markets. It raises concerns about the potential loss of the personal, trust-based relationships between loan officers and clients if digitization completely replaces human interaction.[11]

Balancing Act: The study advocates for a blended approach that combines digital technologies with the flexibility and personal touch of loan officers. It argues that such an approach can help microfinance institutions maintain their competitive edge while enhancing the quality of soft information crucial for financial inclusion.

Soft Information and Trust: The paper emphasizes the importance of soft information in microfinance, which often relies on personal knowledge and relationships to assess borrowers' creditworthiness. It suggests that a complete shift to digital processes may compromise the collection and utilization of this critical information.

The research argues that a balanced approach, leveraging the strengths of both digital technologies and the human touch, is essential for microfinance institutions to remain competitive and continue fostering financial inclusion, particularly in less mature markets. This perspective contributes to the ongoing discourse on the evolving landscape of microfinance and the delicate balance between innovation and tradition in the pursuit of greater financial access for underserved populations.

In another research paper [12] sheds light on the technological challenges and opportunities within the digital lending industry. It underscores the importance of automation, technology maturity, and a seamless user experience. The proposed system architecture aligns with local regulations and incorporates key national entities, demonstrating the need for a tailored approach in this dynamic industry.

The paper operates within the theoretical framework of digital lending, emphasizing the need for advanced technology and automation in the industry. It recognizes the significance of predictive machine learning algorithms in credit scoring models and the importance of integrating various technological components to create a seamless user experience.

The research paper discusses several key findings and issues in the context of digital lending:

Automation and Technology Maturity: The paper acknowledges the role of automation in accelerating processing speed within the digital lending sector. However, it highlights that certain aspects of digital lending, such as background checks on customers, still rely on third-party applications, indicating the need for technological maturity in these areas.

Loan Origination System: The paper underscores the importance of the loan origination system as the core component of the digital lending ecosystem. It discusses the need for continuous improvement and enhancement of this system to meet the evolving technological landscape.

Seamless User Experience: The research paper emphasizes the adoption of high-end mobile app technology to provide a seamless user experience for customers. Creating comfort and ease of use on digital lending platforms is crucial for customer satisfaction and engagement.

Planning and Roadmap: To keep up with the rapid pace of technological adoption in the digital lending industry, the paper suggests the importance of planning and creating a roadmap for technology stack development. This roadmap should align with local regulations and incorporate national entities, such as the national single ID server and local credit bureau.

This research contributes to the understanding of the digital lending landscape in Indonesia and the necessity of technological advancements to meet the evolving needs of customers and regulatory requirements. It emphasizes the importance of a well-planned and adaptable technology roadmap to ensure the sustained growth and success of digital lending businesses in the Indonesian market.[13]

1. The use of automated credit scoring and payment reminders is a growing trend in microfinance. These technologies can help microfinance institutions (MFIs) to reduce loan defaults by more accurately assessing the creditworthiness of borrowers and by reminding borrowers of their loan repayment obligations.

The research paper [14]underscores the transformative impact of AI/ML technologies on the credit process of financial institutions. By enhancing the accuracy of loan default prediction and credit scoring, these technologies contribute to risk reduction and informed lending decisions.

The paper introduces the Seven Seas loan prediction solution, emphasizing its potential benefits for financial institutions. It highlights the broad market scope for such a solution, not only in India but also globally, signaling the significant opportunities for financial institutions to embrace disruptive technology in their transformation initiatives.

The research paper addresses several key findings and issues related to loan prediction and financial technology:

Profitability and Risk Reduction: The paper acknowledges the pivotal role of credit business in a financial institution's profitability and discusses how AI/ML technologies are instrumental in significantly reducing the risk associated with loan defaults.

Data Science Techniques: It explores various data science techniques, such as logistic regression, SVM, neural networks, and random forests, and their contributions to enhancing the accuracy of loan default prediction. These techniques enable financial institutions to make more informed decisions regarding loan disbursements.

Credit Scoring Model: The paper outlines the development of an alternative credit scoring model using machine learning. This model aids in predicting creditworthiness, allowing financial institutions to set the terms of loan disbursements effectively.

Loan Origination Process: The research paper provides insights into the loan origination process, shedding light on the steps involved in loan application and evaluation, with a focus on the application of machine learning techniques.

This research contributes to the understanding of the role of AI/ML in financial technology and the potential for advanced data science techniques to optimize credit processes and reduce the risk of loan defaults in the financial sector.

Microloans have become a crucial financial tool for individuals in fragile economies, offering opportunities for economic growth and empowerment. However, the timely repayment of microloans remains a significant challenge, impacting both lenders and borrowers.

The effectiveness of personalized text message reminders in improving loan repayment rates for microloans in the Philippines. It highlights the potential and limitations of communication technology in addressing payment delays and defaults and underscores the importance of personal connections between borrowers and financial institution staff in overcoming market failures. [15]

The research article [15] sheds light on the impact of personalized text message reminders on microloan repayment rates in the Philippines. It underscores the effectiveness of including the account officer's name in these messages, highlighting the role of personalization in borrower behavior.

The research article presents several key findings and discussions related to personalized text message reminders and microloan repayment:

Personalized Messages: The study demonstrates that personalized text message reminders were effective in improving loan repayment rates, but only when they included the account officer's name. This personal touch played a crucial role in encouraging borrowers to fulfill their repayment obligations.

Account Officer Relationship: The effectiveness of personalized messages was observed primarily for clients who had previously interacted with the account officer servicing their loans. This finding underscores the importance of the existing personal connection between borrowers and bank employees in enhancing loan repayment behavior.

The study contributes to the understanding of how communication technology can be harnessed to address payment delays and defaults in microfinance. It emphasizes the potential of personal connections between borrowers and financial institution staff in overcoming market failures and improving the efficiency of microloan programs. This research serves as a valuable reference for microfinance practitioners and policymakers seeking to enhance the repayment behavior of microloan borrowers through technology and personalization.

The prediction of loan defaults is of paramount importance for financial institutions and banks, as a significant portion of their revenue is reliant on the interest and EMIs generated from loan repayments. Many loans issued by financial institutions carry high interest rates due to the absence of collateral and the uncertainty associated with borrowers. Hence, the development of a model capable of predicting loan defaults holds immense value for financial institutions and banks. Such a model assesses customer data based on specific parameters and provides an accurate prediction, aiding financial institutions in making informed decisions about approving or rejecting loan applications.

"Swindle," a predictive model that utilizes the CatBoost algorithm for loan default prediction, incorporates a document verification module using Tesseract and Camelot, and integrates a personalized loan module to mitigate risks associated with loan issuance. [16]

The paper operates within the theoretical framework of predictive modeling, machine learning, and financial risk assessment. It acknowledges the pivotal role of predictive models in evaluating the creditworthiness of borrowers and reducing the risk of loan defaults. The utilization of machine learning algorithms, such as CatBoost, underscores the importance of advanced analytics in enhancing loan default prediction.

The research paper presents several key findings and discussions related to Swindle and its capabilities:

Loan Default Prediction: Swindle employs the CatBoost algorithm to predict loan defaults accurately. This machine learning technique evaluates various customer parameters and historical data to provide a reliable prediction of whether a borrower is likely to default on their loan.

Document Verification Module: The paper introduces a document verification module using Tesseract and Camelot. This module ensures the authenticity and accuracy of documents submitted by borrowers, further reducing the risk of fraudulent loan applications.

Personalized Loan Module: Swindle incorporates a personalized loan module that tailors loan terms and conditions to individual borrowers. This approach enhances the institution's ability to cater to the specific needs and risk profiles of borrowers.

Risk Mitigation: The primary objective of Swindle is to mitigate the risk associated with loan issuance by providing financial institutions with a robust predictive model. By accurately assessing loan applicants and their likelihood of default, financial institutions can make informed lending decisions.

The research paper by S. Barua, D. Gavandi, P. Sangle, L. Shinde, and J. Ramteke (2021) introduces Swindle, a comprehensive model for predicting the probability of loan defaults. Utilizing the CatBoost algorithm, a document verification module, and a personalized loan module, Swindle addresses the critical challenges faced by financial institutions and banks in assessing borrower creditworthiness and managing risk.

The paper underscores the significance of advanced analytics and machine learning in enhancing loan default prediction, document verification, and personalized lending. Swindle represents a valuable tool for financial institutions seeking to make more informed decisions regarding loan approvals while reducing the risk of defaulters and unauthorized borrowers. This research contributes to the ongoing discourse on the application of technology and data analytics in the financial sector to enhance risk management and improve lending practices.

Credit scoring models play a crucial role in the assessment of borrowers' creditworthiness, enabling financial institutions to make informed lending decisions. [14]

The development of a credit scoring model specifically tailored for commercial loans, aiming to enhance the accuracy and consistency of credit assessments in the financial industry. [17]

The paper operates within the theoretical framework of credit risk assessment and credit scoring models. It recognizes the importance of quantifying the creditworthiness of commercial loan applicants and the need for systematic and data-driven methods to evaluate borrowers' financial health and repayment capacity.

The research paper [18] offer valuable insights into the development of a credit scoring model tailored for commercial loans. Such models have played a significant role in the financial industry by improving the efficiency of credit assessments and helping financial institutions manage credit risk effectively. This paper contributes to the ongoing evolution of credit scoring practices and their impact on lending decisions in the commercial sector.

## 2.3 Summary

The summary section provides a concise recap of the key findings and insights from the literature review. It may also reiterate the significance of the research topic and how the proposed system will contribute to the field. This section prepares the reader for the subsequent chapters of the research, where the focus shifts from the literature review to the actual research methodology, data collection, analysis, and discussion of findings.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Title | Author(s) and Year | Work Done | Gap Identified | Proposed Solution/Approach |
| Centralization of Microfinance | Sonia Singh, Hiranmay Saha (2014) | Discusses the growth and transformation of microfinance organizations and the missing link between lenders and borrowers in the Indian context. | Lack of a centralized platform for efficient interest rate comparisons across microfinance institutions. | Develop a Centralized Loan Application Management System to harmonize multiple microfinance institutions on one platform, enabling borrowers to compare interest rates. |
| Digitalizing microfinance in Europe | Justyna Pytkowska, Piotr Korynski (2017) | Examines the digitization of microfinance and its potential impact on financial inclusion and the "human face" of microfinance institutions. | The lack of traditional human touch in microfinance due to digitization. | Implement a digitized Loan application system for a blended approach, combining digital technologies with human interaction to enhance financial inclusion while retaining the human touch. |
| Loan Prediction Software for Financial Institutions | N. Darapaneni, A. Kumar, A. Dixet, M. Suriyanarayanan, S. Srivastava, A. R. Paduri (2022) | Presents the Seven Seas model as a solution for predicting loan defaults, aiding financial institutions in making informed lending decisions. | The need for accurate loan default prediction to aid financial institutions. | Development of a credit scoring Algorithm that focuses on first hand data presented from the credit client, this means a smaller dataset will mitigate expensive calculations to predict loan defaults. |
| A Personal Touch in Text Messaging | Dean Karlan, Melanie Morten, Jonathan Zinman (2020) | Investigates the impact of personalized text message reminders on microloan repayment in the Philippines. | Lack of personalized text message system for reminders for loan repayment. | Implement personalized in text messaging system for improved microloan repayment, enhancing borrower engagement and improving repayment rates. |

Table 1: Summary of Literature

## 2.4 Proposed System

In this part of the chapter, the researcher transitions from the review of existing literature to discussing their own proposed system or research. Here, the focus is on presenting the research question, objectives, and hypotheses. The researcher outlines their proposed approach, methods, and the expected contributions of the study. It's the point in the literature review where the research begins to carve its unique path and explain how it will build upon or address the gaps identified in the existing literature.

## 2.5 System Review

In the pursuit of addressing the key objectives of this project, it is crucial to examine existing systems and articles related to the central theme of microfinance institutions (MFIs) and their role in providing financial services to underserved populations. The landscape of such systems globally reflects varying degrees of technological advancement, particularly through mobile applications and web-based platforms. Through our research, this study has identified several noteworthy systems and institutions, each with its unique strengths and limitations.

ZAMCASH is a prominent microfinance institution operating in Zambia, offering unsecured loans to individuals through both their website and Android application. While it plays a vital role in providing financial services to the underserved, there are notable limitations in its current setup. ZAMCASH does not support multiple microfinance institutions on its platform, lacks automated payment reminders for clients/debtors, and offers limited cross-platform compatibility. The absence of these features poses challenges in terms of borrower flexibility and efficient loan management.[19]

Another notable MFI in Zambia, IZWE LOANS, faces similar limitations as ZAMCASH. It does not provide support for multiple microfinance institutions, lacks automated payment reminders, and does not offer cross-platform compatibility. These limitations may impact borrower experiences and the institution's competitiveness in the microfinance market.[20]

LUPIYA, another microfinance institution operating in Zambia, distinguishes itself by offering automated payment reminders to its clients. This feature contributes to improved borrower experiences and higher loan repayment rates. LUPIYA also provides users with the flexibility to choose custom interest rates and conducts eligibility checks for clients/debtors seeking loans. However, it is worth noting that LUPIYA is not cross-platform; it primarily operates through its website.

FINCA: Operating in various countries, including Zambia, FINCA is another microfinance institution that faces similar limitations as the aforementioned institutions. It does not support multiple microfinance institutions on its platform, lacks automated payment reminders, and offers limited cross-platform compatibility. Addressing these limitations could significantly enhance its service offerings. FINCA provides secured loans, conducts eligibility checks in person with clients/debtors, and offers diverse payment methods, including hard cash, VISA, debit cards, MTN, and Airtel Money. Notably, FINCA's loan application process still relies on traditional methods involving pen-and-paper forms, and they manually remind clients/debtors to repay their loans.[21]

In summary, while these microfinance institutions in Zambia play crucial roles in expanding financial inclusion, there are noteworthy gaps and limitations within their current systems. These gaps include the absence of support for multiple microfinance institutions on a single platform, the lack of automated payment reminders, and limited cross-platform compatibility. This project aims to address these gaps by developing a comprehensive Centralized Loan Application Management System that harmonizes multiple MFIs on one platform, digitizes the loan application and approval processes, and automates payment reminders and credit scoring. This holistic approach promises to enhance borrower experiences, streamline loan management, and reduce loan default risks, contributing significantly to the microfinance sector in Zambia and beyond.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EXISTING SYSTEMS** | **DIGITAL LOAN APPLICATION PROCESS** | **AUTOMATED PAYMENT REMINDERS** | **CROSS PLATFORM** | **SUPPORTS MULTIPLE MICROFINANCE INSTITUIONS** |
| ZAM-CASH |  | X | X | X |
| IZWE  LOANS |  | X | X | X |
| LUPIYA |  |  | X | X |
| FINCA |  | X | X | X |
| PROPOSED SYSTEM |  |  |  |  |

Table 2: System Review

# Chapter Three - Research Methodology

## 3.1 Introduction

In this chapter, this study delves into the methodology that underpins the software development process. It serves as a roadmap to guide the entire software development lifecycle as it forms the basis for how requirements are gathered, analyzed, and validated.

This chapter also presents the methodology that was used in the case study for this project. This includes the research design, the population and sampling techniques, the methods of data collection and how the data were analysed in order to answer the research questions of the study.

## 3.1.1 Research Design

This research study employed a mixed-methods approach, combining both quantitative and qualitative research methods to provide a comprehensive understanding of microfinance in Zambia.

### 3.1.1.1 Quantitative Research:

The study adopted a descriptive research design with a quantitative approach. To collect quantitative data, online questionnaires were distributed to a diverse sample of microfinance clients. These questionnaires were structured to gather numerical and categorical data on various aspects of microfinance, such as loan applications, interest rates, credit scoring, and loan defaults.

**Data Analysis:** For the quantitative component, the collected questionnaire data underwent statistical analysis using responses on google forms. This included computing descriptive statistics, such as means and percentages, and performing cross-tabulations to explore relationships between variables. The quantitative data analysis aimed to provide an objective overview of key aspects of microfinance.

For the qualitative component, responses to open-ended questions were transcribed and subjected to thematic analysis. Thematic analysis involved identifying recurring themes and patterns in the qualitative data. This allowed for an in-depth exploration of clients' experiences and attitudes related to microfinance services.

By combining quantitative and qualitative research methods, the study aimed to offer a well-rounded perspective on microfinance in Zambia, covering both numerical insights and rich, context-specific qualitative data. This approach enabled a more comprehensive examination of the microfinance landscape in the region, highlighting both the quantitative findings and the nuanced perspectives of microfinance clients.

## 3.2 Software Development Methodology

This section provides an overview of the selected software development methodology that will guide the planning, execution, and monitoring of the project. The choice of a software development methodology is a pivotal decision as it shapes the project's approach, pace, and responsiveness to evolving requirements.

Rapid Application Development (RAD) Methodology

The chosen software development methodology for this project is Rapid Application Development (RAD). RAD is characterized by its commitment to expeditious software delivery and adaptability to evolving project requirements.

Rapid Application Development (RAD) is a software development methodology that prioritizes speed and agility in delivering software solutions. RAD focuses on reducing development time and accelerating the delivery of functional software by emphasizing prototyping, iterative development, and close collaboration with customers.[22]

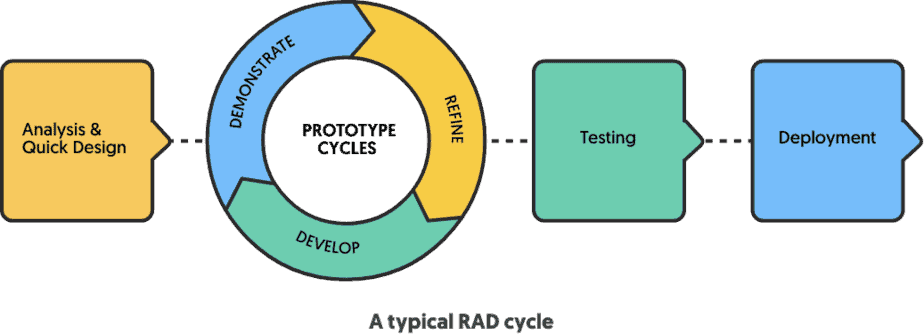


Figure 2: RAD, Software development Methodology

RAD encourages the development of working prototypes of the software. These prototypes allow stakeholders to visualize and interact with the system early in the development process. Feedback gathered from these prototypes facilitates continuous refinement and alignment with user expectations.[22]

RAD adopts an iterative approach to software development, breaking the project into smaller, manageable segments. Each iteration results in a functional part of the software, and these parts are progressively integrated into the final product. This iterative process fosters flexibility, as it allows for incremental updates and adjustments based on evolving requirements.[23]

RAD emphasizes ongoing collaboration with customers and stakeholders. By involving them throughout the development process, RAD ensures that the software aligns closely with user needs and expectations. Frequent interactions and feedback loops enable swift adjustments and enhancements.

### 3.2.1 Functional Requirements

Functional requirements are explicit statements that describe what a system or software application must do, focusing on user needs, and they need to be specific, verifiable, and prioritized. These requirements outline the system's expected behavior and serve as a foundation for testing and development, typically described through use cases and scenarios. [24] The following are the functional requirement for the Centralized Loan Application Management System for Microfinance Institutions.

1. User Authentication and Authorization:

* Users should be able to register and log in to the system.
* Different user roles (e.g., admin, applicant) should have varying levels of access and permissions.

1. Loan Application Management:

* Applicants should be able to submit loan applications through the system.
* Loan officers should be able to review, approve, or reject loan applications.
* The system should support different types of loan products (e.g., collateral loans, Unsecured loans).

1. Data Management:

* The system should securely store and manage applicant and member data.

1. Loan Processing:

* The system should facilitate the processing of loan requests, including verifying applicant eligibility and creditworthiness.
* Loan officers should be able to calculate loan amounts, interest rates, and repayment terms based on predefined criteria.

1. Document Upload and Management:

* Applicants should be able to upload necessary documents for loan processing.
* Loan officers should have access to uploaded documents for review.

1. Communication and Notifications:

* The system should send notifications to applicants regarding the status of their loan application.

### 3.2.2 Non-Functional Requirements

Non-functional requirements, often referred to as quality attributes or system characteristics, describe the qualities or attributes that a system or software application should possess rather than specific functions it should perform. They address aspects like performance, security, scalability, reliability, usability, and maintainability. [25]

1. Security:
2. The system should ensure secure data transmission and storage, including encryption of sensitive information.
3. User access should be protected through robust authentication mechanisms.
4. Usability and User Experience:

* The system should have an intuitive user interface to enhance user experience.
* It should be accessible from both Android and iOS devices.

1. Performance and Scalability:

* The system should be able to handle a high volume of concurrent users and loan applications.
* Response times should be reasonable even during peak usage.

1. Reliability and Availability:

* The system should have a high level of uptime and minimal downtime for maintenance.
* Backup and recovery mechanisms should be in place to prevent data loss.

1. Scalability:

* The system should be designed to accommodate future growth and additional features.

## 3.3 Software Design

In this section, this study delves into the intricacies of the software design process, which constitutes a crucial phase in the development of the Centralized Loan Application Management System. Software design involves the detailed planning and specification of individual software components, encompassing the definition of functionality, interfaces, and data structures. [26]

This phase is pivotal in translating the architectural blueprint into a functional and efficient software solution. The software design process for the Centralized Loan Application Management System is supported by a range of tools and techniques aimed at visualizing and documenting software components, ensuring efficient development and maintenance.

In the development of the Centralized Loan Application Management System, a carefully selected set of tools and practices plays a pivotal role in streamlining the software design and development process. These tools and techniques ensure that the system is not only technically robust but also user-friendly and adaptable to the unique needs of multiple microfinance institutions.

## 3.3.1 UML DIAGRAMS

Unified Modelling Language Diagrams are a graphical notation used to construct and visualize object-oriented systems.[27]

### 3.3.1.1 Contextual Model

A UML context diagram is a high-level visual representation used in system design and analysis to show a system's boundaries, its interactions with external entities, and the flow of information between them. It consists of a system boundary, external entities, arrows representing data or control flow, and labels for interaction descriptions. This diagram simplifies complex systems by focusing on critical elements and their relationships, serving as a starting point for more detailed modeling in the UML framework.[28]

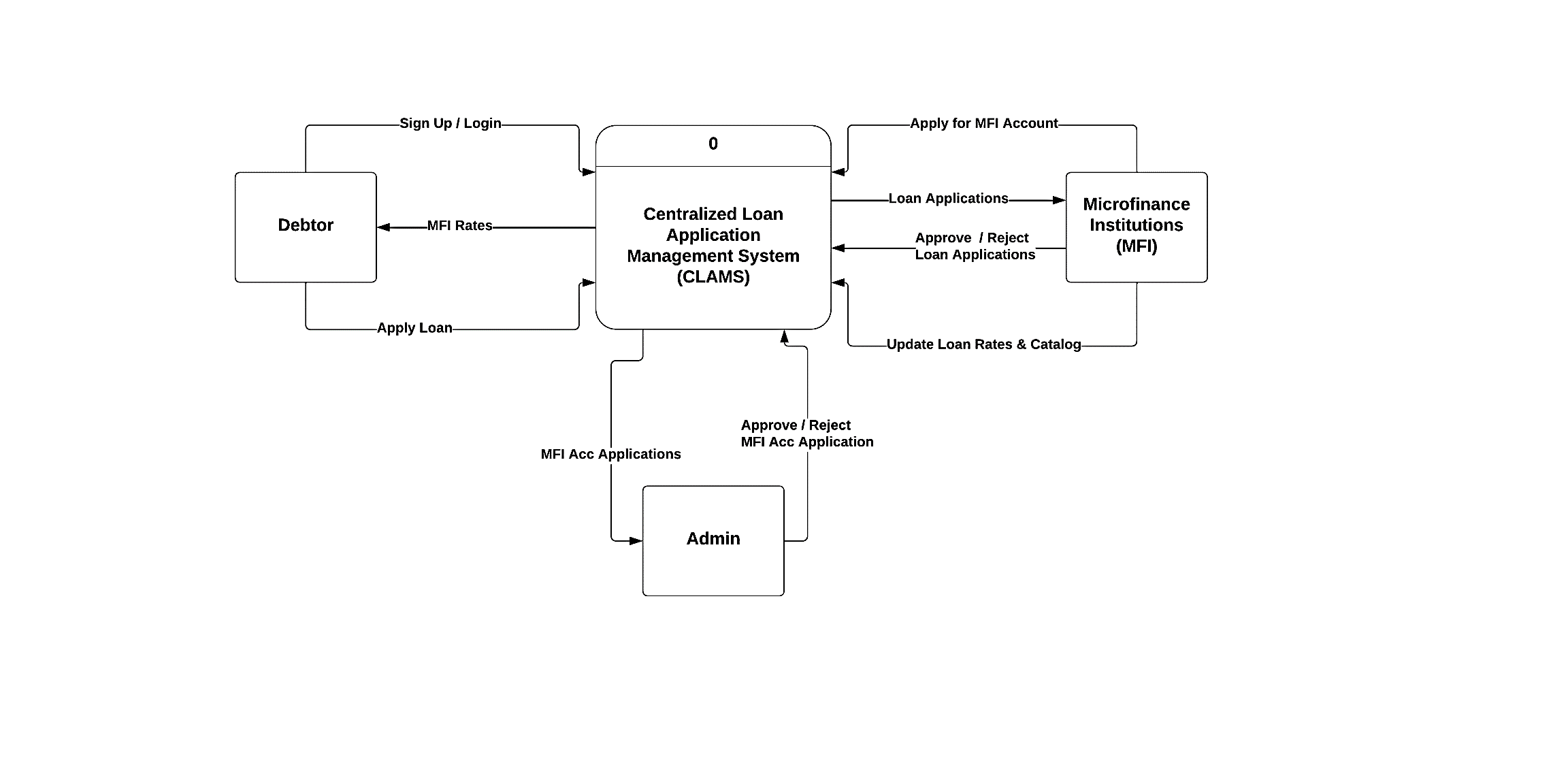


Figure 3: Contextual Diagram

### 3.3.1.2 Use Case Model

Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact. In this project UML diagrams such as Use case diagrams, Class diagram and flowcharts were used. The use case diagram demonstrates how events are triggered into the system as the actors (users) interact with the website.[29] The system is comprised of three users namely; Administrator, Debt and MFI.

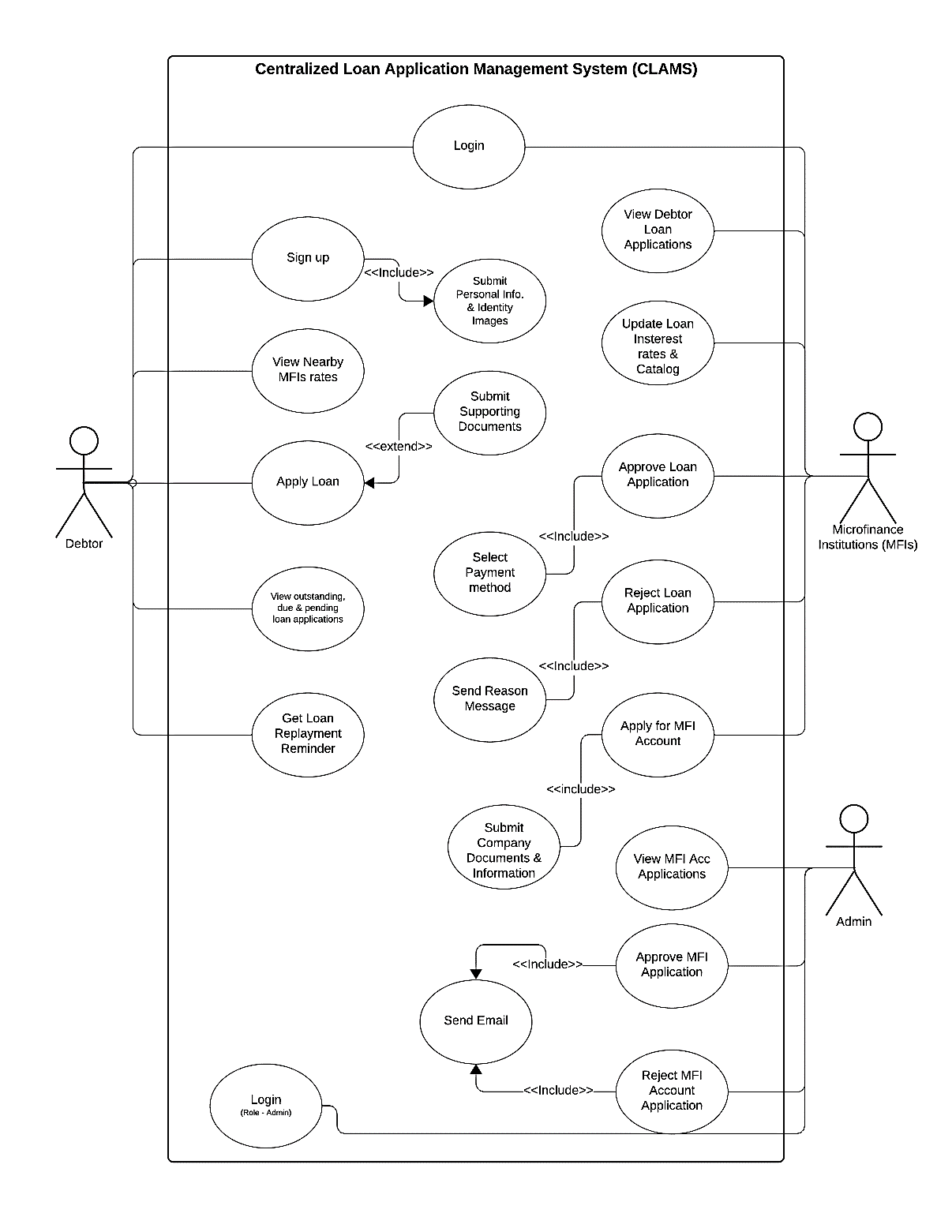


Figure 4: Use-case Diagram

### 3.3.4 Flow Chart

A flowchart is a picture of the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes, such as a manufacturing process, an administrative or service process, or a project plan. It's a common process analysis tool and one of the seven basic quality tools.[31]

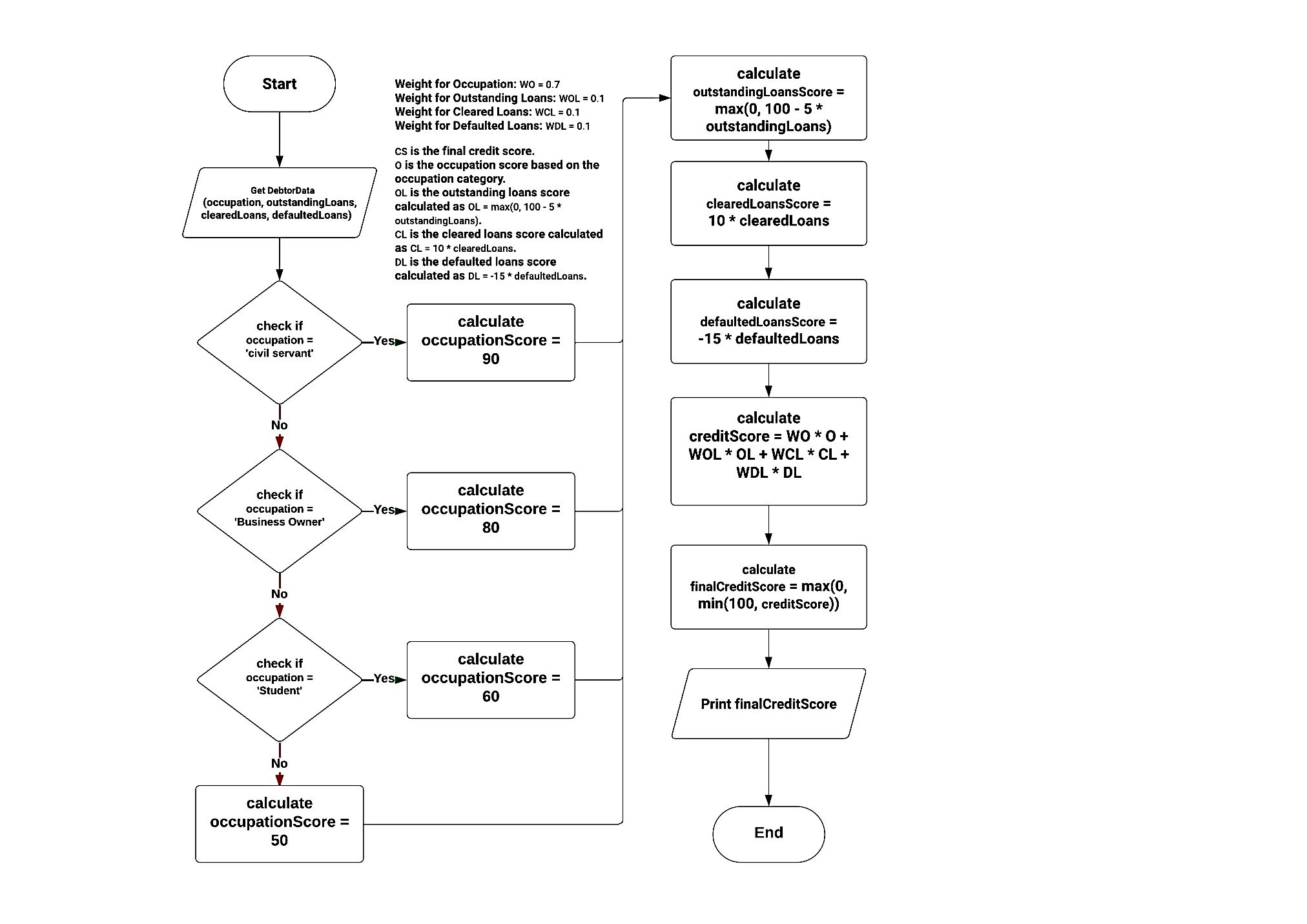
Elements that may be included in a flowchart are a sequence of actions, materials or services entering or leaving the process (inputs and outputs), decisions that must be made, people who become involved, time involved at each step, and/or process measurements. 

Figure 5: Flow Chart

.

## 3.4 Software Development Tools

The development stack selected for the project comprises a combination of technologies and tools to efficiently deliver the desired software solution:

Frontend: The user interface of the application was crafted using React Native. This choice provides several advantages, foremost among them being cross-platform compatibility for both iOS and Android. React Native enables developers to create a single codebase that can run on multiple platforms, reducing development effort and ensuring a consistent user experience across devices. Its component-based architecture and extensive library of pre-built UI components make it a popular choice for mobile app development.[32]

Backend: Node.js was chosen as the backend technology for this project. Node.js is known for its speed and efficiency, which are particularly valuable for real-time applications. It's responsible for handling the application's logic, serving as the intermediary between the frontend and the database. Node.js's non-blocking, event-driven architecture is well-suited for building responsive and scalable server applications, making it a solid choice for supporting the application's core functionality.[33]

Database Management: The project's database needs are managed using MongoDB. MongoDB is a NoSQL database management system that is known for its flexibility and scalability. Its lightweight nature is particularly well-suited for mobile applications, as it can efficiently handle the storage and retrieval of data while adapting to the evolving needs of the project. Its document-based, schema-less structure allows for easy data modeling and modification, making it a practical choice for this project's dynamic data requirements.[34]

This development stack was carefully selected to create a robust and efficient application that can run on both major mobile platforms, offer a responsive and real-time user experience, and handle the project's data storage and retrieval needs in an adaptable and efficient manner. By combining React Native, Node.js, and MongoDB, the development team has leveraged modern technologies to create a versatile and performant solution.

Effective software development relies on a set of essential tools that streamline the process of writing, compiling, and maintaining code. These tools empower developers to work efficiently, collaborate seamlessly, and maintain code quality.

In the context of the Centralized Loan Application Management System project, the following software development tools have been employed:

1. Integrated Development Environments (IDEs):

Integrated Development Environments play a fundamental role in modern software development. For this project, Visual Studio Code (VS Code) is the IDE of choice. VS Code offers a wide range of features that enhance productivity, including code completion, debugging capabilities, a vibrant extension ecosystem, and integrated version control. It enables developers to write, test, and debug code efficiently while fostering a collaborative and consistent development environment.[35]

1. Code Editors:

In addition to the primary IDE, developers also leverage code editors tailored to the specific needs of various programming languages and frameworks. VS Code's extensibility allows for the integration of language-specific extensions, enabling syntax highlighting, code formatting, and context-aware autocompletion. This flexibility ensures that code is written in adherence to best practices and coding standards.[35]

1. Version Control Systems (VCS):

Git, a distributed version control system, combined with GitHub, a web-based platform for hosting and collaborating on Git repositories, plays a pivotal role in the project's development workflow. Version control systems like Git offer the ability to track changes in the source code, manage branching and merging, and facilitate collaborative development.[36] GitHub provides a centralized platform for hosting the project's source code, tracking issues, and managing contributions from team members and external collaborators. This combination ensures code integrity, version management, and a seamless collaborative environment.[37]

1. Build Tools:

Node.js-based build tools, such as npm (Node Package Manager) and yarn, are instrumental in managing project dependencies, automating build processes, and facilitating the deployment of the Centralized Loan Application Management System. [38]These tools simplify the installation of necessary packages, enable script execution for tasks like compiling code or bundling assets, and ensure a smooth development and deployment pipeline.

1. Prototyping and Wireframing

Wireframing tools are instrumental in the initial stages of designing the system's user interface. They allow for the creation of low-fidelity sketches, facilitating a swift and iterative approach to frontend design. In parallel, the use of prototyping tools, such as Adobe XD and Microsoft Visio, takes the design process a step further. [39] These tools empower the development team to craft interactive, high-fidelity mock-ups, providing a realistic and tangible preview of the system's user interface[40]. This visual clarity helps stakeholders to better understand the end product and guides the development process towards a user-centric design. [41]

These software development tools collectively empower the development team to create a robust, maintainable, and collaborative codebase for the Centralized Loan Application Management System. They enhance productivity, code quality, and the ability to adapt to evolving project requirements and industry best practices.

# Chapter Four - Research Results and Analysis & Implementation

## 4.1 Introduction

This chapter of the report outlines the results from the research that was conducted and highlights the methodology used in the actual implementation of the project.

## 4.2 Data Analysis

Statistically the research uncovered that 98.8% percent of the sampled population would value a mobile application that allows the users to find multiple Microfinance institutions on a single platform for the purpose of efficient loan interest comparison.

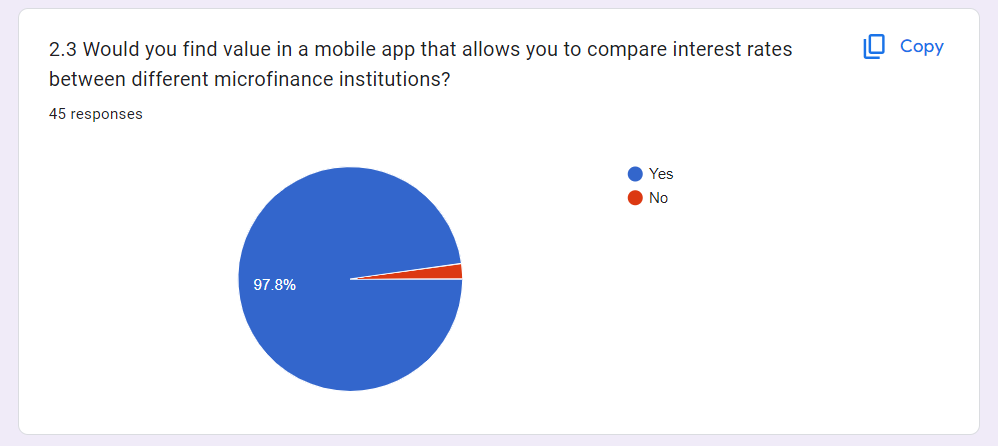


Figure 6: Question 2.3 & Response to questionnaire

Out of 48 responses from the sampled population, 51.1% visit local offices of microfinance institutions physically to get access to their services. This proves that half of the sampled population can’t access microfinance services from the comfort of their homes.

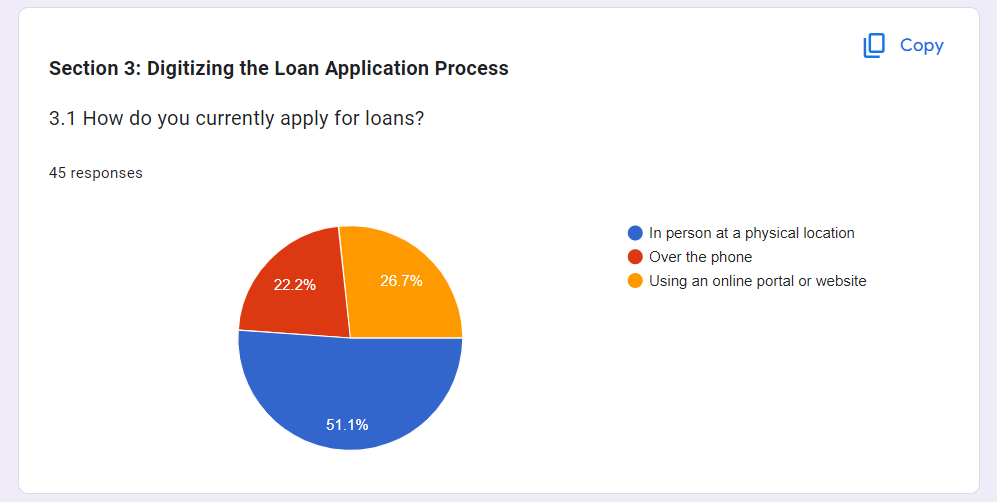


Figure 7: Question 3.1 & Response to questionnaire

Overall, 48 responses were received, statistically this research uncovered that 71.1% of the samples population actually prefer received automated payment reminders via text messages compared to phone calls or emails.

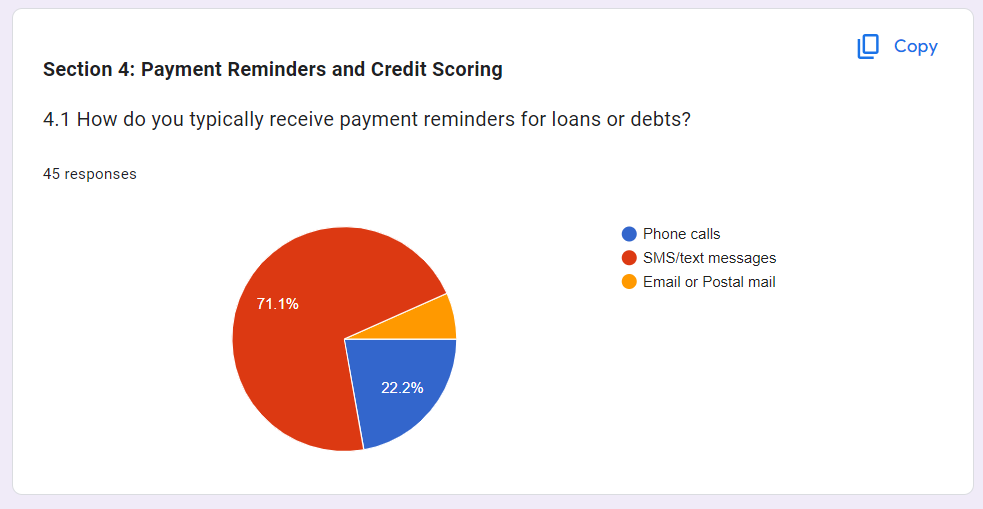


Figure 8: Question 4.1 & Response to questionnaire

## 4.3 Results of the Study

Overall, 48 responses were received, out of which 26, representing 55.6 have actually interacted with microfinance services and the rest witnessed other people interact with microfinance institutions. With regard to the need for a mobile app that digitalize the loan application process, 97.8% of the sampled population would actually be interested in a mobile application that digitalize the loan application process.

This research also uncovered that; automated payment reminders would be helpful in ensuring timely payment of loans hence mitigating the risks of loan defaults.

## Experiment Results and Analysis

The study aimed at gauging respondent’s loan application practices and therefore asked a series of questions related to the types of microfinance institutions given to, frequency and types of loans they apply and how they apply and interact with microfinance institutions and the means through which these interactions were made.

## 4.4 System Implementation

**Introduction**

In order to make this project a success, the ‘Rapid Application Development’ method was chosen as the preferred software development model. Rapid Application development is suited for situations in which there is a limited time to accomplish objectives of a software project and hence the normal waterfall method of development can not be used. It with this reason that the developers of this system opted to use RAD as the preferred method of software development. This section provides a deep insight of how exactly this system was implemented. Included in this section are screenshots of the various pages on the platform.

**Microfinance Institution Verification**

The developers included a Microfinance Institution (MFI) user account verification functionality into this system. The reason for this addition was due to the fact that any fraudster can pose as an MFI and create an account with Amastata.

Therefore, in order to fully ensure that all MFI accounts created are legitimate organisations, verification of PACRA[42] registration details by the MFI must be submitted upon registration. Additionally, the MFI representative must submit a copy of their NRC in order for their account to be verified. When the account request has been approved by Amastata administrators only then can the account be published on the Mobile Application as a legitimate MFI.

**User Interface Elements**

The user interface of this system was built using React Native. Due to the limited time that was assigned to the development of this project, the developers so the need to incorporate user interface frameworks in order to speed up the development of this system. Expo was used in the development of this project.

**System Backend Elements**

The backend of the system was built using Node JS and MongoDB. Due to the limited time that was assigned to the development of this project, the developers incorporated the use of the lightweight JavaScript Library, Express, to speed up development time and increase the security of the mobile application to be produced.

**Technical Details**

Since the Centralized Loan Application mobile application and eeb application, various web-based technologies were used in the development of this system. Technologies used in the development of this system are:

* 1. React Native is an open-source framework for building mobile applications using JavaScript and React. It allows developers to create mobile apps for multiple platforms, such as iOS and Android, using a single codebase. React Native offers a component-based architecture and a rich ecosystem of libraries and tools for building user interfaces.[32]
  2. Expo is a set of tools, libraries, and services that simplifies and accelerates the development of React Native applications. It provides a development environment with features like a live preview, easy project setup, and access to native device features without the need for complex native code development.[43]
  3. Node.js is a runtime environment that allows developers to run JavaScript on the server-side. It's known for its non-blocking, event-driven architecture, making it well-suited for building scalable and real-time applications. Node.js is commonly used for server-side web development, enabling developers to build efficient and performant applications.[33]
  4. npm is the default package manager for Node.js. It allows developers to easily manage and share packages (libraries and modules) of JavaScript code. Developers can use npm to install, update, and publish packages, simplifying the process of integrating third-party code into their projects.[38]
  5. Express is a web application framework for Node.js. It simplifies the process of building web applications and APIs by providing a set of robust features, such as routing, middleware, and templating. Express is often used in conjunction with Node.js to create efficient and modular server-side applications. [44]

**Database System** - A database system is a collection of application programs that interact with the database along with the database management system and the database itself, a Database Management System is a software that manages and controls access to the database. This project makes use of the popular MongoDB.

1. MongoDB is a NoSQL database management system that is known for its flexibility and scalability. Its lightweight nature is particularly well-suited for mobile applications, as it can efficiently handle the storage and retrieval of data while adapting to the evolving needs of the project. Its document-based, schema-less structure allows for easy data modeling and modification, making it a practical choice for this project's dynamic data requirements.[34]
2. Mongoose is an Object Data Modeling (ODM) library for MongoDB and Node.js. It provides a structured way to interact with MongoDB by defining schemas and models for data. Mongoose simplifies data validation, querying, and interactions with MongoDB, making it easier for developers to work with the database.[45]

## 4.5 Evaluation & Testing

The success of the Centralized Loan Application Management System hinges on the rigorous evaluation and testing of its software components. To ensure that the system is functional, secure, and compliant with the specified requirements, a range of evaluation and testing techniques are employed throughout the software development life cycle. These techniques include:

1. Unit Testing:

Unit testing is an indispensable part of the development process. It focuses on evaluating the individual components or units of code, such as functions and methods, in isolation. Test cases are designed to verify that each unit functions correctly and produces the expected outcomes.[46] For this project, tools like Jest are used to automate unit testing, helping to identify and rectify bugs and issues at an early stage of development.

2. Integration Testing:

Integration testing evaluates the interactions between various units or modules of the software. It ensures that these units work together cohesively and that data flows correctly between them. [47] The project employs integration testing to validate the interactions between client-side and server-side components, guaranteeing seamless communication and data exchange.

3. System Testing:

System testing assesses the system as a whole, verifying that it fulfills the specified requirements and functions as intended. It includes end-to-end testing of the Centralized Loan Application Management System, encompassing the entire software stack, from the user interface to the database. Test scenarios are designed to replicate real-world user interactions, ensuring that the system performs reliably and meets user expectations.[48]

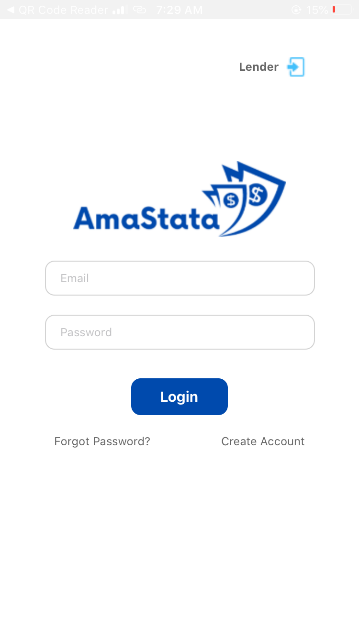
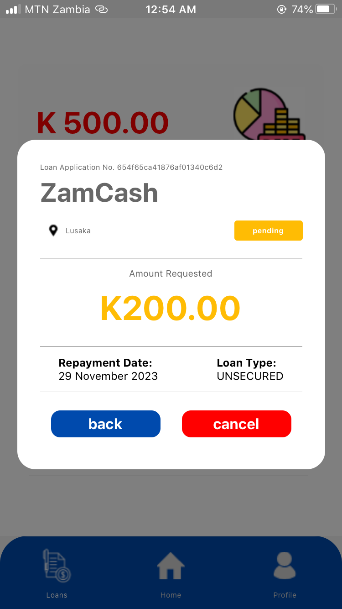
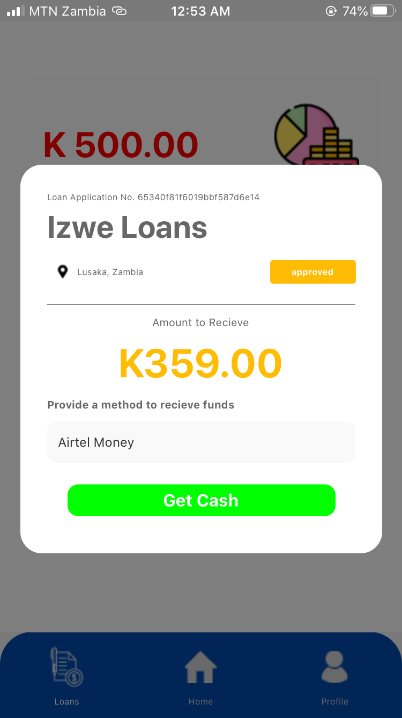
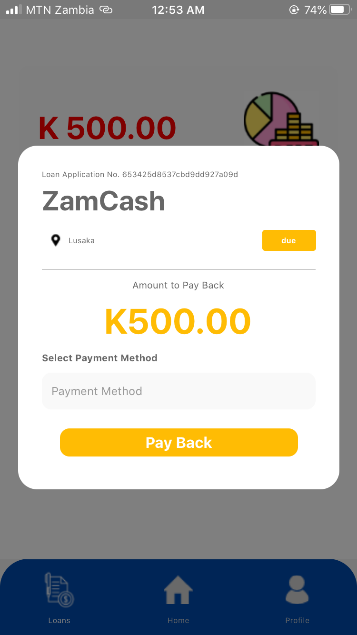
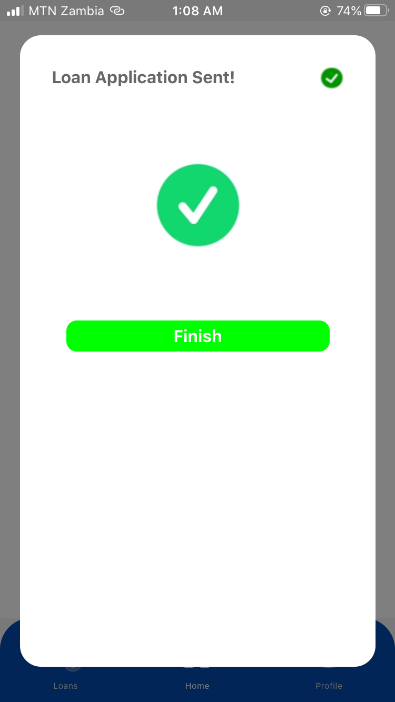
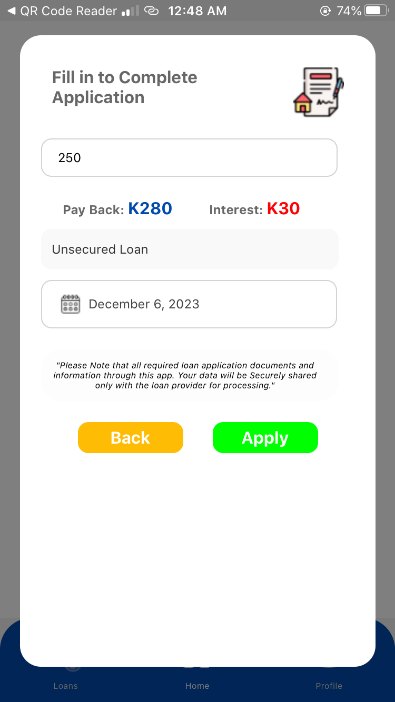
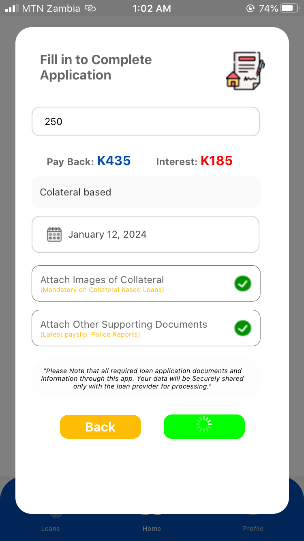
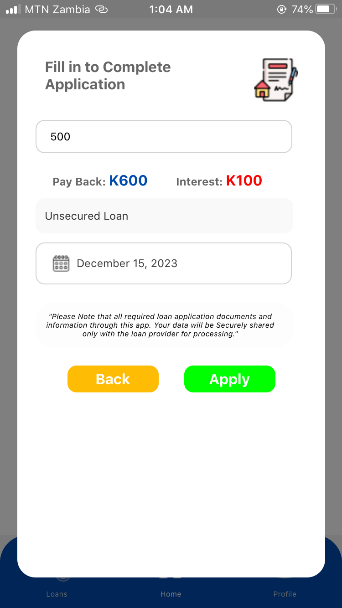
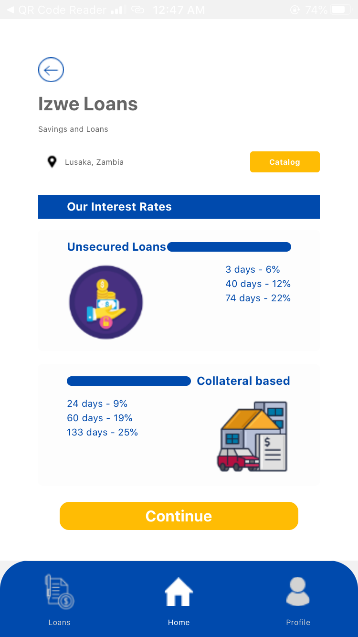
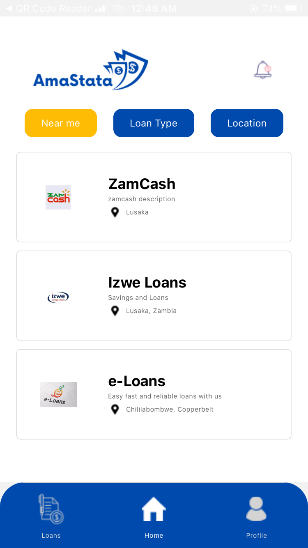
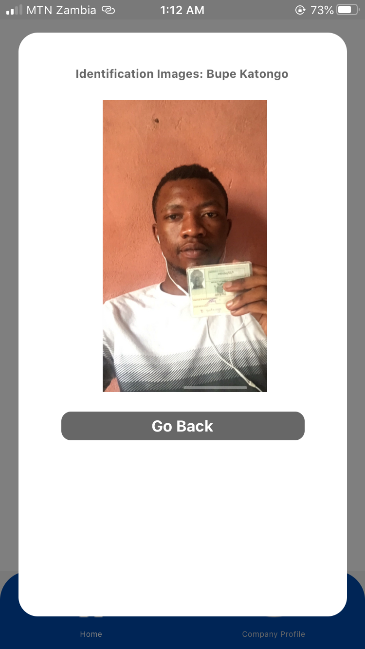
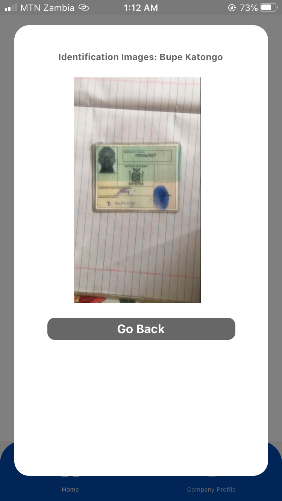
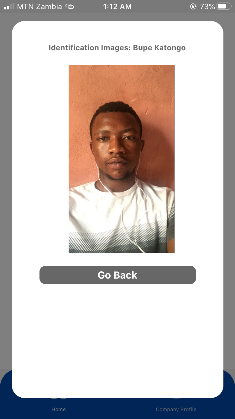
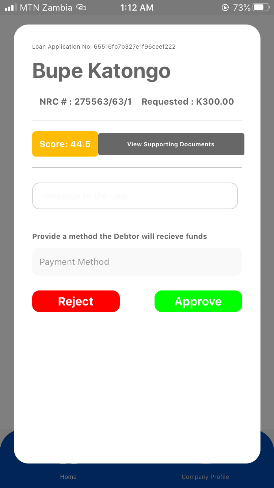
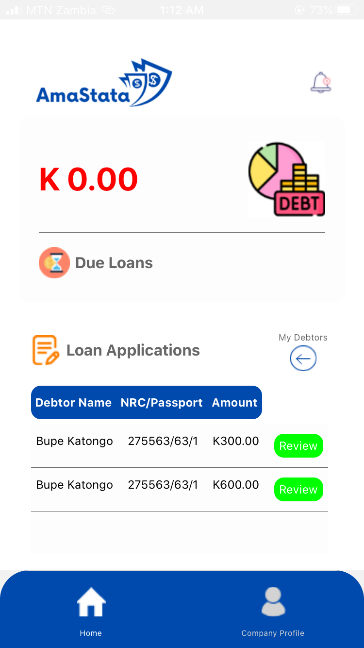
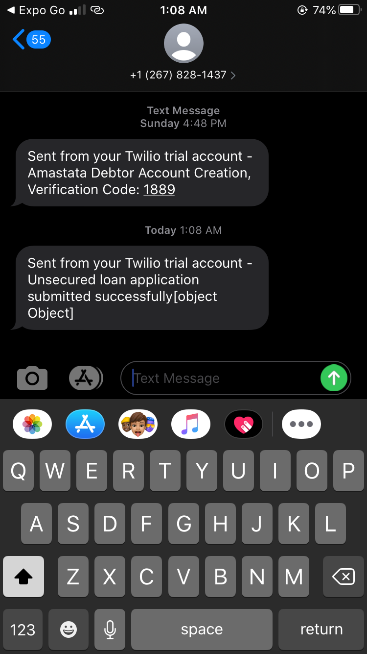
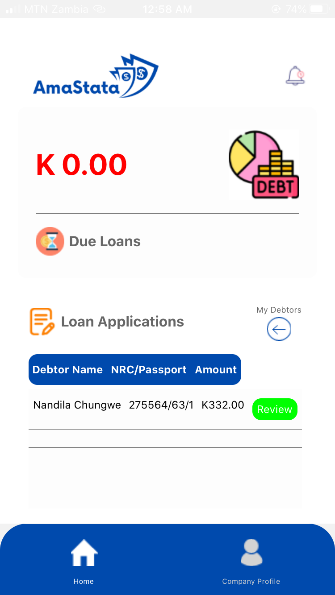
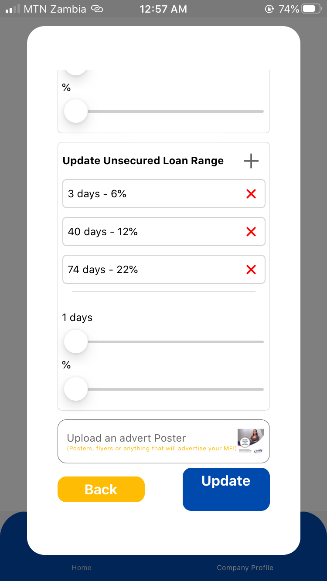
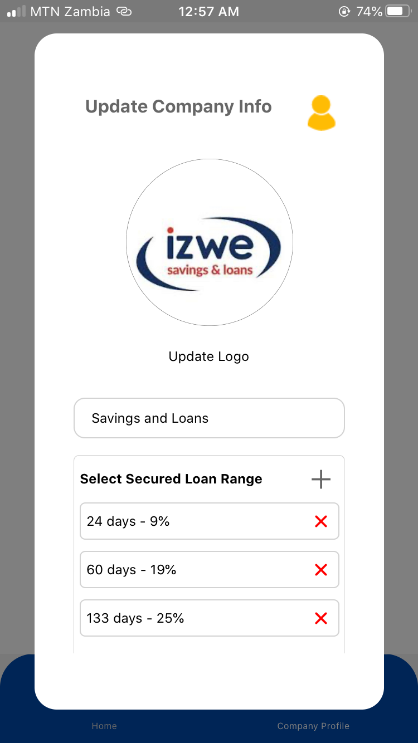
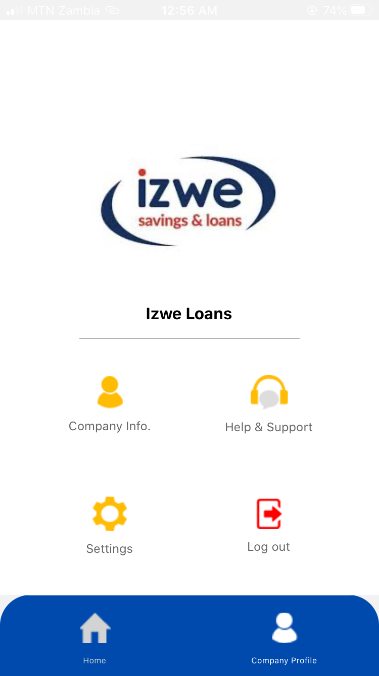
4. User Acceptance Testing (UAT):

User Acceptance Testing is a critical phase in software evaluation, involving end-users to assess the system's suitability for its intended purpose. Users, including both administrators and clients of the microfinance institutions, actively participate in testing to validate that the system aligns with their operational needs and is user-friendly.[49]

These evaluation and testing techniques are integral to the development process, serving to identify and address defects, validate functionality, and ensure the system's reliability and security. By following a comprehensive testing strategy, the Centralized Loan Application Management System aims to deliver a high-quality and dependable solution for microfinance institutions in Ndola, Zambia.

**SYSTEM SCREENSHOTS HERE…**

Centralizing Microfinance Institution

USER SIGN UP LENDER LOGIN & SIGN UP

Digitizing the loan application Process

Automated Payment Reminder and Credit Scores

# Chapter Five - Conclusion and Recommendation

## 5.1 Introduction

In conclusion, this dissertation has delved into the multifaceted realm of microfinance, focusing on the development of a Centralized Loan Application Management System (CLAMS) to address the challenges faced by microfinance institutions (MFIs) in Zambia. The significance of microfinance in poverty alleviation cannot be overstated, and the proposed system aims to enhance the efficiency and accessibility of financial services to underprivileged individuals and businesses.

The study's primary objective was to create a harmonized platform that aggregates multiple MFIs, offering a centralized space for loan comparison and application. The proposed CLAMS not only streamlines the application process but also contributes to mitigating the risk of loan defaults, a pressing issue for MFIs. The research has provided insights into how such a system can be tailored to meet the needs of clients and debtors in a localized context.

## 5.2 System Accomplishments

The systems based success on whether or not the system solved the problem it was built for, the delivery time of the system, cost against budget, and if it does what it was intended to do.

The developed system is successfully developed a cross platform mobile application and web application that facilitates microfinance institution on a single platform for the purpose of efficient and effective loan interest comparisons among different microfinance institutions.

The system is able to detect near loan risk defaults and inform both the debtor and the lender in order for them to make timely decisions. The system also mitigates microfinance fraud by ensuring that only registered institutions are able to utilise the services of the platform by means of the verification process.

The system also facilitates local microfinance institutions to view the total amount in debt, outstanding loans and debtor information as well and pending loan applications.

## 5.3 System Limitations

Despite possessing the core features it was designed to have, the system has quite a number of limitations. The system’s limitations are listed below:

1. Users are unable to view a MFI’s exact office location on a map.
2. The system can only be accessed through mobile applications and cannot be used as a progressive web application.
3. The system does not have a provision for live chat with the system administrators. Communication can only be done via email.

## 5.4 Future Enhancements

The following features listed below will be added to the application as enhancements in the future:

* 1. The use of google maps to pinpoint the charity’s location in the event of donors who wish to visit the charities or prove their locality existence.
  2. Integration of payment gateways (Mobile money, VISA card etc.)
  3. The application will be able to have a virtual 24/7 chat service between Debtor & Lender.

## 5.5 Recommendations

As the global discourse on poverty alleviation continues to evolve, microfinance stands out as a powerful tool for fostering economic empowerment among underprivileged individuals and businesses. This dissertation has undertaken the ambitious task of developing a Centralized Loan Application Management System (CLAMS) tailored to the specific context of microfinance institutions (MFIs) in Ndola, Zambia. Having navigated through the intricacies of microfinance and the nuances of designing a system to enhance financial accessibility, this section focuses on pivotal recommendations.

These recommendations serve as a roadmap for the effective implementation, sustainable growth, and widespread acceptance of the proposed CLAMS, acknowledging the collaborative efforts required to bridge the gap between financial services and those in need. Addressing key aspects such as collaborative integration, continuous monitoring, user education, scalability, cybersecurity, and community engagement, these recommendations lay the groundwork for a transformative leap in the landscape of microfinance in Zambia and beyond.

### 5.5.1 Operational Recommendations

To address inefficiencies, it is recommended to implement a phased training program for microfinance institution staff, ensuring a smooth transition to the new system. Additionally, establishing a dedicated support hotline and online knowledge base will provide ongoing assistance.

### 5.5.2 Policy Recommendations

Policy recommendations include advocating for the revision of data protection laws to facilitate the collection of comprehensive client information within ethical boundaries. Collaboration with regulatory bodies is recommended to align policies with technological advancements in the microfinance sector.

### 5.5.3 User Training and Support

To enhance user training, the implementation of interactive e-learning modules and regular refresher courses is recommended. Ongoing user support will involve the establishment of a responsive support team and community forums to encourage knowledge sharing among users.

### 5.5.4 Collaboration Opportunities

Recommendations include exploring collaborative opportunities with government agencies to extend the system's reach to remote areas. Collaboration with local universities and technology providers is also recommended to foster innovation and ensure continuous improvement of the Centralized Loan Application Management

# References

[1] A. Bakker, J. Schaveling, and A. Nijhof, “Governance and microfinance institutions,” *Corp. Gov.*, vol. 14, no. 5, pp. 637–652, 2014.

[2] I. García-Pérez, M. Á. Fernández-Izquierdo, and M. J. Muñoz-Torres, “Microfinance institutions fostering sustainable development by region,” *Sustainability*, vol. 12, no. 7, p. 2682, 2020.

[3] C. Munsaka and F. Chileshe, “The Challenges of Accessing Microfinance in Rural Areas: A Case Study of Zambia.,” *2020*.

[4] Michael Mumba and B. Mwape, “The Use of Physical Loan Application Forms in Microfinance: A Case Study of Zambia.,” *2020*.

[5] R. Emekter, Y. Tu, B. Jirasakuldech, and M. Lu, “Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending,” *Appl. Econ.*, vol. 47, no. 1, pp. 54–70, Jan. 2015, doi: 10.1080/00036846.2014.962222.

[6] S. Sonia and Saha, “Centralization of Microfinance,” *2014*.

[7] D. Kariuki and J. Mutau, “A Review of Software Systems for Comparing Loan Rates Across Multiple Microfinance Institutions.,” *2022*.

[8] A. K. Kar and R. B. Swain, “Interest Rates and Financial Performance of Microfinance Institutions: Recent Global Evidence,” *Eur. J. Dev. Res.*, vol. 26, no. 1, pp. 87–106, Jan. 2014, doi: 10.1057/ejdr.2013.33.

[9] C. Basten and M. Mariathasan, “How banks respond to negative interest rates: Evidence from the Swiss exemption threshold,” 2018, Accessed: Nov. 06, 2023. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=3164780

[10] J. Nwachukwu, “Interest Rates, Target Markets and Sustainability in Microfinance,” *Oxf. Dev. Stud.*, vol. 42, no. 1, pp. 86–110, Jan. 2014, doi: 10.1080/13600818.2013.827164.

[11] S. Khan, S. Khan, and M. Aftab, “Digitization and its impact on economy,” *Int. J. Digit. Libr. Serv.*, vol. 5, no. 2, pp. 138–149, 2015.

[12] A. Bakhrun, Suhardi, and L. Ramadani, “IT value implementation at microfinance sector,” in *2014 International Conference on Information Technology Systems and Innovation (ICITSI)*, Nov. 2014, pp. 219–223. doi: 10.1109/ICITSI.2014.7048267.

[13] J. Pytkowska and P. Korynski, “Digitalizing microfinance in Europe,” *Microfinance Cent.*, pp. 1–12, 2017.

[14] S. Y. Sohn, D. H. Kim, and J. H. Yoon, “Technology credit scoring model with fuzzy logistic regression,” *Appl. Soft Comput.*, vol. 43, pp. 150–158, 2016.

[15] D. Karlan, M. Morten, and J. Zinman, “A personal touch in text messaging can improve microloan repayment,” *2020*.

[16] S. Barua and D. Gavandi, “Swindle: Predicting the Probability of Loan Defaults using CatBoost Algorithm,” *2021*.

[17] Y. E. Orgler, “A Credit Scoring Model for Commercial Loans,” *2017*.

[18] Y.-C. Chang, K.-H. Chang, and G.-J. Wu, “Application of eXtreme gradient boosting trees in the construction of credit risk assessment models for financial institutions,” *Appl. Soft Comput.*, vol. 73, pp. 914–920, 2018.

[19] “Zamcash Zambia | Get a loan in Zambia in 5 minutes.” Accessed: Nov. 06, 2023. [Online]. Available: https://www.zamcash.com/

[20] “Izwe Zambia – Shape your financial future with flexible products.” Accessed: Nov. 06, 2023. [Online]. Available: https://www.izwezambia.com/

[21] “Loan Application | FINCA Zambia.” Accessed: Nov. 06, 2023. [Online]. Available: https://finca.co.zm/loan-application/

[22] M. Alexandrou, “Rapid Application Development (RAD) Methodology,” *Retrieved Febr.*, vol. 2, 2010.

[23] K. Dimeska and S. Savoska, “Model of software development using RAD methods and standard ISO/IEC 12207,” in *Proceedings/8 th International conference on applied internet and information technologies*, “St Kliment Ohridski” University-Bitola, Faculty of Information and …, 2018, pp. 48–51. Accessed: Nov. 06, 2023. [Online]. Available: https://www.researchgate.net/profile/Marijana-Dragicevic/publication/332041337\_Business\_Intelligence\_in\_Power\_Distribution\_System/links/5e7360bc92851c93e0ad6d1a/Business-Intelligence-in-Power-Distribution-System.pdf#page=66

[24] R. Malan and D. Bredemeyer, “Functional requirements and use cases,” *Bredemeyer Consult.*, 2001, Accessed: Nov. 06, 2023. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=aceaa41855c38aebe7c823e60e94b39506a92b99

[25] M. Glinz, “On non-functional requirements,” in *15th IEEE international requirements engineering conference (RE 2007)*, IEEE, 2007, pp. 21–26. Accessed: Nov. 06, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4384163/

[26] H. Gomaa, “Advances in Software Design Methods for Concurrent, Real-Time and Distributed Applications,” in *2008 The Third International Conference on Software Engineering Advances*, Oct. 2008, pp. 451–456. doi: 10.1109/ICSEA.2008.78.

[27] H. Koç, A. M. Erdoğan, Y. Barjakly, and S. Peker, “UML diagrams in software engineering research: a systematic literature review,” in *Proceedings*, MDPI, 2021, p. 13. Accessed: Nov. 06, 2023. [Online]. Available: https://www.mdpi.com/2504-3900/74/1/13

[28] Y. Wang *et al.*, “CoCoSum: Contextual Code Summarization with Multi-Relational Graph Neural Network.” arXiv, Jul. 05, 2021. Accessed: Nov. 06, 2023. [Online]. Available: http://arxiv.org/abs/2107.01933

[29] A. Gemino and D. Parker, “Use case diagrams in support of use case modeling: Deriving understanding from the picture,” *J. Database Manag. JDM*, vol. 20, no. 1, pp. 1–24, 2009.

[30] S. Al-Fedaghi, “UML Sequence Diagram: An Alternative Model.” arXiv, May 31, 2021. Accessed: Nov. 06, 2023. [Online]. Available: http://arxiv.org/abs/2105.15152

[31] N. Chapin, “Flowchart,” in *Encyclopedia of computer science*, 2003, pp. 714–716. Accessed: Nov. 06, 2023. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/1074100.1074406

[32] “React Native · Learn once, write anywhere.” Accessed: Nov. 06, 2023. [Online]. Available: https://reactnative.dev/

[33] “Node.js,” Node.js. Accessed: Nov. 06, 2023. [Online]. Available: https://nodejs.org/en

[34] “MongoDB Atlas Database | Multi-Cloud Database Service | MongoDB.” Accessed: Nov. 06, 2023. [Online]. Available: https://www.mongodb.com/atlas/database

[35] “Visual Studio Code - Code Editing. Redefined.” Accessed: Nov. 06, 2023. [Online]. Available: https://code.visualstudio.com/

[36] “Git.” Accessed: Nov. 06, 2023. [Online]. Available: https://git-scm.com/

[37] “Build software better, together,” GitHub. Accessed: Nov. 06, 2023. [Online]. Available: https://github.com

[38] “npm | Home.” Accessed: Nov. 06, 2023. [Online]. Available: https://www.npmjs.com/

[39] “Adobe XD Learn & Support.” Accessed: Nov. 06, 2023. [Online]. Available: https://helpx.adobe.com/content/help/en/support/xd.html

[40] S. Putman, *Using prototyping in instructional design*. Association for Training and Development, 2014. Accessed: Nov. 06, 2023. [Online]. Available: https://books.google.com/books?hl=en&lr=&id=K1aWBQAAQBAJ&oi=fnd&pg=PA1&dq=wireframe+prototype&ots=A7iNhDqZD3&sig=A9wuKmYcENc3\_YMFqBvh53y72n8

[41] C. Andrews *et al.*, “A New Method in User-Centered Design: Collaborative Prototype Design Process (CPDP),” *J. Tech. Writ. Commun.*, vol. 42, no. 2, pp. 123–142, Apr. 2012, doi: 10.2190/TW.42.2.c.

[42] “Patents and Companies Registration Agency | Business Starts Here.” Accessed: Nov. 06, 2023. [Online]. Available: https://www.pacra.org.zm/

[43] “Dashboard,” Expo. Accessed: Nov. 06, 2023. [Online]. Available: https://expo.dev/

[44] “Express - Node.js web application framework.” Accessed: Nov. 06, 2023. [Online]. Available: https://expressjs.com/

[45] “Mongoose ODM v8.0.0.” Accessed: Nov. 06, 2023. [Online]. Available: https://mongoosejs.com/

[46] E. Daka and G. Fraser, “A survey on unit testing practices and problems,” in *2014 IEEE 25th International Symposium on Software Reliability Engineering*, IEEE, 2014, pp. 201–211. Accessed: Nov. 06, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6982627/

[47] H. K. Leung and L. White, “A study of integration testing and software regression at the integration level,” in *Proceedings. Conference on Software Maintenance 1990*, IEEE, 1990, pp. 290–301. Accessed: Nov. 06, 2023. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/131377/

[48] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, “The oracle problem in software testing: A survey,” *IEEE Trans. Softw. Eng.*, vol. 41, no. 5, pp. 507–525, 2014.

[49] H. Taherdoost, “A review of technology acceptance and adoption models and theories,” *Procedia Manuf.*, vol. 22, pp. 960–967, 2018.

# Appendix

Installation Guide

1. React Native Expo Managed Project:

React Native with Expo provides a streamlined development workflow for building cross-platform mobile applications. Expo manages much of the configuration, making it easier for developers to focus on the app's logic and user interface.

* 1. Installation Guide:
     1. Install Node.js and npm:
     2. Visit the official Node.js website (https://nodejs.org/).
     3. Download and install the latest LTS version.
     4. npm (Node Package Manager) is included with Node.js.
     5. Install Expo CLI:
        1. Open a terminal or command prompt.
        2. Run the following command to install Expo CLI globally:
           1. npm install -g expo-cli
        3. Create a new Expo project:
           1. Run the following commands to create a new Expo project:

expo init YourProjectName

cd YourProjectName

* + - 1. Start the Expo development server:
         1. Navigate into your project folder and run:

expo start

* + - * 1. This will start the Expo development server, and you'll see a QR code in the terminal.
      1. Run on your device:
         1. Install the Expo Go app on your iOS or Android device.
         2. Use the Expo Go app to scan the QR code from the terminal.
         3. Your app will be compiled and launched on your device.

1. Node.js and Express.js:

Node.js is a runtime for executing JavaScript code server-side, while Express.js is a web application framework for Node.js, simplifying the process of building robust and scalable web applications.

* 1. Installation Guide:
     1. Install Node.js:
        1. Same as the React Native Expo installation, download and install the latest LTS version of Node.js from the official website.
        2. Create a Node.js project:
        3. Create a new folder for your Node.js project.
     2. Open a terminal, navigate to the project folder, and run:
        1. npm init -y
        2. Install Express.js:
     3. Run the following command to install Express.js:
        1. npm install express
     4. Create an Express.js server:
     5. Run the Express.js server:
        1. In the terminal, run:
           1. node server.js

Your Express.js server is now running, and you can access it at http://localhost:3000 in your browser.

These steps should help you set up a React Native Expo managed project along with a Node.js and Express.js server. Adjust the details according to your project requirements.

Research Questionnaires

Section 1: General Information

* 1. Name (Optional)
  2. Have you ever or do you know anyone who has applied for a loan with a microfinance institution?

1. Yes
2. No

2.2 How do you typically find and compare interest rates when applying for a loan? \*

1. Visiting multiple institutions in person
2. Using online resources
3. Relying on recommendations from friends/family

2.3 Would you find value in a mobile app that allows you to compare interest rates between different microfinance institutions?\*

1. Yes
2. No

Section 3: Digitizing the Loan Application Process

3.1 How do you currently apply for loans?\*

1. In person at a physical location
2. Over the phone
3. Using an online portal or website

3.2 What challenges, if any, have you encountered while applying for loans through these methods?

3.3 Would you be interested in using a mobile app to digitize the loan application process?\*

1. Yes
2. No

Section 4: Payment Reminders and Credit Scoring

4.1 How do you typically receive payment reminders for loans or debts?\*

1. Phone calls
2. SMS/text messages
3. Email or Postal mail

4.2 Do you think automated payment reminders would be helpful in ensuring timely payments?\*

1. Yes
2. No

4.3 Are you aware of credit scoring systems that evaluate your creditworthiness when applying for loans?\*

1. Yes
2. No

4.4 Would you be comfortable with a mobile app using credit scoring to assess your creditworthiness when applying for loans?\*

1. Yes
2. No

Essential Modules Source Code

1. MFI Centralization on the Platform source.
   1. **SOURCE CODE HERE…**

**app.get("/lenders", async (req, res) => {**

**console.log("fetching lender data...");**

**try {**

**// Use the find method to retrieve all lender documents from the database**

**const lenders = await Lender.find();**

**// Check if there are lenders in the database**

**if (lenders.length === 0) {**

**return res.status(404).json({ error: "No lenders found" });**

**}**

**// console.log(lenders);**

**// Split lenders into three sets based on status**

**const approvedLenders = lenders.filter(**

**(lender) => lender.LenderStatus.status === "approved"**

**);**

**const rejectedLenders = lenders.filter(**

**(lender) => lender.LenderStatus.status === "rejected"**

**);**

**const pendingLenders = lenders.filter(**

**(lender) => lender.LenderStatus.status === "pending"**

**);**

**const readyLenders = lenders.filter(**

**(lender) => lender.LenderStatus.status === "ready"**

**);**

**// Determine the response format based on the client's request**

**const acceptHeader = req.get("Accept");**

**if (acceptHeader && acceptHeader.includes("application/json")) {**

**// Respond with JSON if the client accepts JSON**

**res.status(200).json({**

**approved: approvedLenders,**

**ready: readyLenders,**

**rejected: rejectedLenders,**

**pending: pendingLenders,**

**lenders: lenders,**

**});**

**} else {**

**// Render the "lenders.ejs" view for other requests**

**res.render("admin-dashboard", {**

**lenders,**

**approvedLenders,**

**rejectedLenders,**

**pendingLenders,**

**readyLenders**

**});**

**}**

**} catch (error) {**

**console.error(error);**

**res.status(500).json({ error: "Internal server error" });**

**}**

**});**

**{approvedLenders.map((lender) => (**

**<TouchableOpacity**

**key={lender.\_id} // Assuming each lender has a unique ID**

**onPress={() => goToNextScreen(lender)} // Pass the lender ID to your next screen**

**style={{**

**flexDirection: "row",**

**height: 110,**

**// padding: 5,**

**alignItems: "center",**

**justifyContent: "space-between",**

**borderRadius: 5,**

**borderWidth: 0.3,**

**borderColor: 'lightgray',**

**marginVertical: 5,**

**}}**

**>**

**{/\* Render the lender information within your card \*/}**

**<View**

**style={{**

**width: 80,**

**height: 30,**

**marginRight: 15,**

**flex: 1,**

**marginLeft: 15,**

**}}**

**>**

**{lender && (**

**<Image**

**style={{ width: "100%", height: "100%", flex: 1 }} // Set the aspect ratio as needed**

**source={{**

**uri: `http://172.20.10.3:2010${lender.LenderAdvert.logo}`,**

**}}**

**resizeMode="contain" // Use "contain" to fit the image within the container**

**/>**

**)}**

**</View>**

**<View style={{ flex: 3, paddingHorizontal: 10 }}>**

**<Text style={{ fontSize: 22, fontWeight: "bold" }}>**

**{lender.LenderCompanyInfo.companyName}**

**</Text>**

**<Text**

**style={{**

**fontSize: 10,**

**color: colors.secondary,**

**marginVertical: 3,**

**}}**

**>**

**{lender.LenderAdvert.briefDesc}**

**</Text>**

**<View style={{ flexDirection: "row", alignItems: "center" }}>**

**<View>**

**<Image source={smallIcons.locationIcon} />**

**</View>**

**<Text**

**style={{**

**fontSize: 10,**

**color: colors.secondary,**

**marginLeft: 5,**

**}}**

**>**

**{lender.LenderCompanyInfo.businessOperationalAddress}**

**</Text>**

**</View>**

**</View>**

**</TouchableOpacity>**

**))}**

1. Digitized Loan Application Process
   1. **SOURCE CODE HERE…**

**app.post("/apply-unsecured-loan", async (req, res) => {**

**try {**

**const authToken = req.headers.authorization;**

**const secretKey = process.env.JWT\_SECRET;**

**const debtorId = verifyToken(authToken, secretKey);**

**console.log("Sending Data to MFI: ", debtorId);**

**const debtor = await Debtor.findOne({ \_id: debtorId });**

**const debtorData = extractDebtorData(debtor);**

**// console.log('Credit score data: ', debtorData)**

**const creditScore = calculateCreditScore(debtorData);**

**console.log(**

**`${debtor.basicInformation.firstname} Credit Score: ${creditScore}`**

**);**

**const loanApplicationData = {**

**debtorId,**

**lenderId: req.body.lenderId,**

**amountRequested: req.body.amountRequested,**

**loanRepaymentDate: req.body.loanRepaymentDate,**

**creditScore,**

**status: "pending",**

**};**

**await createLoanApplication(loanApplicationData);**

**await updateDebtorCreditScore(debtorId, creditScore);**

**console.log("Loan Applied Successfully");**

**//SEND ALERTS TO MFIs ON THE NEW LOANS APPLIED**

**res**

**.status(201)**

**.json({ message: "Unsecured loan application submitted successfully" });**

**sendAlert(**

**debtor.basicInformation.phoneNumberPrimary,**

**"katongobupe@hotmail.com",**

**"Unsecured loan application submitted successfully" + loanApplicationData**

**);**

**} catch (error) {**

**console.error("Error:", error);**

**res.status(500).json({ error: "Internal Server Error" });**

**}**

**function extractDebtorData(debtor) {**

**return {**

**occupation: debtor.basicInformation.occupation,**

**outstandingLoans:**

**debtor.creditScoresVars.currentLoans == null**

**? 0**

**: debtor.creditScoresVars.currentLoans,**

**clearedLoans:**

**debtor.creditScoresVars.clearedLoans == null**

**? 0**

**: debtor.creditScoresVars.clearedLoans,**

**defaultedLoans:**

**debtor.creditScoresVars.defaultedLoans == null**

**? 0**

**: debtor.creditScoresVars.defaultedLoans,**

**};**

**}**

**async function createLoanApplication(loanApplicationData) {**

**const loanApplication = new UnsecuredLoanApplication(loanApplicationData);**

**await loanApplication.save();**

**}**

**async function updateDebtorCreditScore(debtorId, creditScore) {**

**return Debtor.findOneAndUpdate(**

**{ \_id: debtorId },**

**{**

**$set: {**

**"creditScoresVars.currentLoans": +1,**

**"creditScoresVars.creditScore": creditScore,**

**},**

**},**

**{ new: true }**

**);**

**}**

**});**

**app.get("/user-loan-data", async (req, res) => {**

**const authToken = req.headers.authorization;**

**const secretKey = process.env.JWT\_SECRET; // Replace with your actual secret key**

**const userId = verifyToken(authToken, secretKey);**

**console.log("Fetching User Loan Data: ", userId);**

**//SEND ALERTS TO MFIs ON THE NEW LOANS APPLIED**

**try {**

**const securedLoanData = await SecuredLoanApplication.find({**

**debtorId: userId,**

**});**

**const unsecuredLoanData = await UnsecuredLoanApplication.find({**

**debtorId: userId,**

**});**

**if (securedLoanData && unsecuredLoanData) {**

**// Respond with the user's loan data**

**console.log(**

**"User Loan Data Fetched: ",**

**securedLoanData,**

**unsecuredLoanData**

**);**

**res.json({ unsecured: unsecuredLoanData, secured: securedLoanData });**

**}**

**} catch (error) {**

**console.log("Unauthorized user!");**

**// Handle token verification or data fetching errors**

**res.status(401).json({ error: "Unauthorized" });**

**}**

**});**

**app.get("/mfi-loan-data", async (req, res) => {**

**const authToken = req.headers.authorization;**

**const secretKey = process.env.JWT\_SECRET; // Replace with your actual secret key**

**const userId = verifyToken(authToken, secretKey);**

**console.log("Fetching MFI Loan Data: ", userId);**

**try {**

**const securedLoanData = await SecuredLoanApplication.find({**

**lenderId: userId,**

**});**

**const unsecuredLoanData = await UnsecuredLoanApplication.find({**

**lenderId: userId,**

**});**

**if (securedLoanData && unsecuredLoanData) {**

**// Respond with the user's loan data**

**console.log(**

**"User Loan Data Fetched: ",**

**securedLoanData,**

**unsecuredLoanData**

**);**

**res.json({ unsecured: unsecuredLoanData, secured: securedLoanData });**

**}**

**} catch (error) {**

**console.log("Unauthorized user!");**

**// Handle token verification or data fetching errors**

**res.status(401).json({ error: "Unauthorized" });**

**}**

**});**

1. Automated Payment Reminders and Credit Score Alogorithm
   1. **SOURCE CODE HERE…**

**async function automatedReminder() {**

**console.log("Preparing Reminder...");**

**try {**

**console.log("fetching loans...");**

**const securedLoans = await SecuredLoanApplication.find();**

**console.log("Secured loans Fetched!", securedLoans);**

**const unsecuredLoans = await UnsecuredLoanApplication.find();**

**console.log("UnsecuredLoans loans Fetched!", securedLoans);**

**console.log("Fetching Debtors Next...");**

**const debtors = await Debtor.find();**

**console.log("Debtors loans Fetched!");**

**if (securedLoans.length > 0) {**

**for (let i = 0; i < securedLoans.length; i++) {**

**// Access the specific loan's repayment date using securedLoans[i].loanRepaymentDate**

**if (calculateDaysDifference(Date.now(), new Date(securedLoans[i].loanRepaymentDate)) <= 5) {**

**for (let j = 0; j < debtors.length; j++) {**

**// Access the debtor using debtors[j] instead of debtors.\_id**

**if (debtors[j].\_id == securedLoans[i].debtorId) {**

**if(calculateDaysDifference(Date.now(), new Date(securedLoans[i].loanRepaymentDate) < 0)){**

**console.log('Remind ', debtors[j].basicInformation.firstname, ' Loan was Due ', calculateDaysDifference(Date.now(), new Date(securedLoans[i].loanRepaymentDate)), 'Days Ago!', JSON.stringify(securedLoans[i]));**

**}else{**

**console.log('Remind ', debtors[j].basicInformation.firstname, ' Loan was is Due in', calculateDaysDifference(Date.now(), new Date(securedLoans[i].loanRepaymentDate)), 'Days ', JSON.stringify(securedLoans[i]));**

**}**

**}**

**}**

**}**

**}**

**}**

**if (unsecuredLoans.length > 0) {**

**for (let i = 0; i < unsecuredLoans.length; i++) {**

**// Access the specific loan's repayment date using securedLoans[i].loanRepaymentDate**

**if (calculateDaysDifference(Date.now(), new Date(unsecuredLoans[i].loanRepaymentDate)) <= 5) {**

**for (let j = 0; j < debtors.length; j++) {**

**// Access the debtor using debtors[j] instead of debtors.\_id**

**if (debtors[j].\_id == unsecuredLoans[i].debtorId) {**

**console.log('Remind ', debtors[j].basicInformation.firstname, ' Loan is Due in ', calculateDaysDifference(Date.now(), new Date(unsecuredLoans[i].loanRepaymentDate)), 'Days ', JSON.stringify(unsecuredLoans[i]));**

**}**

**}**

**}**

**}**

**}**

**} catch (error) {**

**console.error(error);**

**}**

**}**

**const connectWithRetry = () => {**

**mongoose**

**.connect(mongoDBConnection, {**

**useNewUrlParser: true,**

**useUnifiedTopology: true,**

**})**

**.then(() => {**

**console.log("Connected to MongoDB");**

**app.listen(port, () => {**

**console.log(`Server is running on port ${port}`);**

**//\*\*\*\*\*\*\*CALL THIS FUCNTION THREE TIMES A DAY (7hrs, 14hrs, 16hrs) \*\*\*\*\*\*\*\*\* \*/**

**cron.schedule('0 7,12,17 \* \* \*', () => {**

**automatedReminder();**

**//IF LOAN IS DUE ADD INTEREST**

**addInterest(debtorId, loanApplicationId, interest);**

**});**

**});**

**})**

**.catch((error) => {**

**console.error("Error connecting to MongoDB:", error);**

**console.log(**

**"Retrying in 5 seconds..."**

**);**

**setTimeout(connectWithRetry, 5000); // Retry after 5 seconds**

**});**

**};**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCTION CALLS ON START SERVER\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**connectWithRetry();**

1. Database Design
   1. **Debtor Schema**

**const mongoose = require("mongoose");**

**const bcrypt = require("bcrypt");**

**const debtorSchema = new mongoose.Schema({**

**basicInformation: {**

**firstname: String,**

**lastname: String,**

**occupation: String,**

**phoneNumberPrimary: String,**

**phoneNumberSecondary: String,**

**dateOfBirth: Date,**

**},**

**residentialAddress: {**

**houseNumber: String,**

**streetName: String,**

**township: String,**

**province: String,**

**district: String,**

**country: String,**

**},**

**identificationImages: {**

**nrcFrontImage: String,**

**nrcBackImage: String,**

**nrcNumber: String,**

**portraitImage: String,**

**portraitWithNRCImage: String,**

**},**

**completeSetup: {**

**emailAddress: String,**

**password: String,**

**confirmPassword: String,**

**},**

**creditScoresVars: {**

**currentLoans: Number,**

**clearedLoans: Number,**

**defaultedLoans: Number,**

**creditScore: Number**

**}**

**});**

**debtorSchema.pre("save", async function (next) {**

**const salt = await bcrypt.genSalt();**

**this.completeSetup.password = await bcrypt.hash(this.completeSetup.password, salt);**

**next();**

**});**

**// Add a method to compare passwords**

**debtorSchema.methods.comparePassword = async function(candidatePassword) {**

**try {**

**console.log('comparing passwords....');**

**return await bcrypt.compare(candidatePassword, this.completeSetup.password);**

**} catch (error) {**

**throw error;**

**}**

**};**

**const Debtor = mongoose.model("Debtor", debtorSchema);**

**module.exports = Debtor;**

* 1. **Lender Schema**

**const mongoose = require('mongoose');**

**const bcrypt = require('bcrypt');**

**const lenderSchema = new mongoose.Schema({**

**LenderCompanyInfo: {**

**companyName: String,**

**businessOperationalAddress: String,**

**phoneNumberPrimary: String,**

**PhoneNumberSecondary: String,**

**businessWebsite: String,**

**},**

**AuthorizedPersonel: {**

**firstname: String,**

**lastname: String,**

**phoneNumberPrimary: String,**

**emailAddress: String,**

**occupation: String,**

**NRCFilePath: String, // Store the file path for NRC**

**},**

**LenderKYCDocument: {**

**KYCDocumentFilePath: String, // Store the file path for KYCDocument**

**},**

**LenderCompleteSetup: {**

**businessEmailAddress: String,**

**password: String,**

**},**

**LenderStatus: {**

**status: String**

**},**

**LenderAdvert:{**

**briefDesc: String,**

**logo: String,**

**securedLoanInterest: [String],**

**unsecuredLoanInterest: [String],**

**catalogImage: String**

**}**

**});**

**lenderSchema.pre("save", async function (next) {**

**const salt = await bcrypt.genSalt();**

**this.LenderCompleteSetup.password = await bcrypt.hash(this.LenderCompleteSetup.password, salt);**

**next();**

**});**

**// Add a method to compare passwords**

**lenderSchema.methods.comparePassword = async function(candidatePassword) {**

**try {**

**return await bcrypt.compare(candidatePassword, this.LenderCompleteSetup.password);**

**} catch (error) {**

**throw error;**

**}**

**};**

**const Lender = mongoose.model("Lender", lenderSchema);**

**module.exports = Lender;**

* 1. **Admin Schema**

**const mongoose = require('mongoose');**

**// Define the schema for SecuredLoanApplication**

**const securedLoanApplicationSchema = new mongoose.Schema({**

**debtorId: {**

**type: String, // Assuming debtorId is a reference to a User model**

**required: true,**

**},**

**lenderId: {**

**type: String, // Assuming lender is a reference to a User model**

**required: true,**

**},**

**amountRequested: {**

**type: String,**

**required: true,**

**},**

**loanRepaymentDate: {**

**type: String,**

**required: true,**

**},**

**collateralImagePath: {**

**type: String, // Store the file path or URL to the collateral image**

**},**

**supportingDocxPath: {**

**type: String, // Store the file path or URL to the supporting document**

**},**

**status: {**

**type: String,**

**},**

**// Add other fields as needed**

**});**

**// Create a model for SecuredLoanApplication**

**const SecuredLoanApplication = mongoose.model('SecuredLoanApplication', securedLoanApplicationSchema);**

**module.exports = SecuredLoanApplication;**

* 1. **Secured Loans Schema**

**const mongoose = require('mongoose');**

**// Define the schema for SecuredLoanApplication**

**const securedLoanApplicationSchema = new mongoose.Schema({**

**debtorId: {**

**type: String, // Assuming debtorId is a reference to a User model**

**required: true,**

**},**

**lenderId: {**

**type: String, // Assuming lender is a reference to a User model**

**required: true,**

**},**

**amountRequested: {**

**type: String,**

**required: true,**

**},**

**loanRepaymentDate: {**

**type: String,**

**required: true,**

**},**

**collateralImagePath: {**

**type: String, // Store the file path or URL to the collateral image**

**},**

**supportingDocxPath: {**

**type: String, // Store the file path or URL to the supporting document**

**},**

**status: {**

**type: String,**

**},**

**// Add other fields as needed**

**});**

**// Create a model for SecuredLoanApplication**

**const SecuredLoanApplication = mongoose.model('SecuredLoanApplication', securedLoanApplicationSchema);**

**module.exports = SecuredLoanApplication;**

* 1. **Unsecured Loans Schema**

**const mongoose = require("mongoose");**

**// Define the schema for SecuredLoanApplication**

**const unsecuredLoanApplicationSchema = new mongoose.Schema({**

**debtorId: {**

**type: String, // Assuming debtorId is a reference to a User model**

**required: true,**

**},**

**lenderId: {**

**type: String, // Assuming lender is a reference to a User model**

**required: true,**

**},**

**amountRequested: {**

**type: String,**

**required: true,**

**},**

**loanRepaymentDate: {**

**type: Date,**

**required: true,**

**},**

**status: {**

**type: String,**

**},**

**});**

**// Create a model for SecuredLoanApplication**

**const UnsecuredLoanApplication = mongoose.model(**

**"unsecuredLoanApplication",**

**unsecuredLoanApplicationSchema**

**);**

**module.exports = UnsecuredLoanApplication;**