

REPUBLIQUE DU CAMEROON

Paix-Travail-Patrie

Faculte D'ingenierie Et De Technologie



REPUBLIC OF CAMEROON

Peace-Work-Fatherland

Faculty Of Engineering And Technology

UNIVERSITY OF BUEA
FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER ENGINEERING
CEF 349: ANALYSIS AND DESIGN OF AN INFORMATION SYSTEM
COURSE INSTRUCTOR: MR KINGUE

**ANALYSIS AND DESIGN OF A MICRO-FINANCE
MANAGEMENT INFORMATION SYSTEM**

Presented by : **GROUP 6.**

NAME	MATRICLE NUMBER
NGADANG ASSONFACK PASCALE ESTELLE **	FE23A109
TAKU-MBI INYANG*	FE23A153
DJEUGWA WEAH MAXIMILIENNE	FE23A036
NWANTOLY STEPHEN	FE23A127
CHEWACHONG LARRY	FE23A031
CHE RANDY	FE23A030
NYANYOH DIVINE	FE23A128



smoLON

Empowering you, One Loan at a Time

Software Requirement Specification (SRS) Document Template for A Microfinance Management System (SMOLLON MICROFINANCE)

Project Name: SMOLLON MICROFINANCE

Date: 10/01/2025

Version:

By:

Presented by : **GROUP 6.**

NAME	MATRICLE NUMBER
NGADANG ASSONFACK PASCALE ESTELLE **	FE23A109
TAKU-MBI INYANG*	FE23A153
DJEUGWA WEAH MAXIMILIENNE	FE23A036
NWANTOLY STEPHEN	FE23A127
CHEWACHONG LARRY	FE23A031
CHE RANDY	FE23A030
NYANYOH DIVINE	FE23A128

Table of Contents

1. INTRODUCTION AND COMPETITIVE ANALYSIS	1
1.1 Purpose:	1
1.2. Document Conventions	1
1.3. Definition of the System	1
1.4. Subject Matter	1
1.5. Notable Examples and Work Done by Competition	1
1.6. Areas Where Notable Rivals Failed	2
1.7. Overview of the System	2
1.8. Differentiating Factors	2
1.9. Intended Audience	3
2. Requirements Gathering and Data Analysis	4
2.1. Requirements Gathering	4
2.2. Data Analysis and Visualization	6
3. OVERALL DESCRIPTION	11
3.1. Product Perspective	11
3.2. Product Functions	11
3.3. User Characteristics	12
3.4. General Constraints	12
3.5. Assumptions and Dependencies	13
4. External Interface Requirements	14
4.1. User Interface	14
4.2. Hardware Interfaces	14
4.3. Software Interfaces	14
5. DESIGN CONSTRAINTS	16
5.1. Standards	16
5.2. Hardware Limitations	16
5.3. Software Limitations	16
6. ADDITIONAL REQUIREMENTS	18
6.1. Timeline and Duration	18
6.2. Project Cost and Worth	18
6.3. List of Staff Members	19

7. DESIGN THINKING PROCESS.....	21
EMPATHIZE.....	21
7.2. DEFINE: Framing the problem	22
7.3. IDEATE: Generate solutions.....	22
7.4. PROTOTYPE DEVELOPING MODEL	23
7.5. TEST.....	23
8. SYSTEM AND ARCHITECTURAL DESIGNS	24
8.1. SYSTEM OVERVIEW	24
8.2. ARCHITECTURAL SYLE/PATTERN	24
8.3. DEPLOYMENT DIAGRAM.....	26
8.4. SYSTEM DECOMPOSITION.....	26
9. UI/UX DESIGN	27
9.1. User interface design	27
9.2. USER EXPERIENCE DESIGN	36
10. USABIITY DESIGN	37
10.1. USABILITY PRINCIPLES	38
10.2. ACCESSIBILITY CONSTRAINTS	38
10.3. USABILITY Testing PLAN.....	38
11. DETAILED DESIGN	39
11.1. DATA DESIGN	39
11.2. SECURITY DESIGN	41
12. DESIGN METHODOLOGIES	44
12.1. Object oriented modeling design.....	44
12.2. Event-driven design.....	47
12.3. Data flow diagrams.....	50
13. Non functional requirements.....	51

1. INTRODUCTION AND COMPETITIVE ANALYSIS

1.1 Purpose: The purpose of this Software Requirements Specification (SRS) document is to outline the functional and non-functional requirements for the Microfinance Information System (MIS). This document serves as a comprehensive guide for stakeholders, including developers, testers, project managers, and investors, to ensure that the system meets the needs of microfinance institutions.

1.2. Document Conventions

- **Functional Requirement:** A specific behavior or function of the system.
- **Non-Functional Requirement:** Attributes such as performance, usability, and reliability
- **User Stories:** Short descriptions of a feature from the perspective of the end user.
- **Use Case:** A detailed description of how users will interact with the system.

1.3. Definition of the System

The Microfinance Information System (MIS) aims to streamline the operations of microfinance institutions by providing a comprehensive platform for managing client information, loan processing, and financial reporting. The system will enhance efficiency, improve customer service, and facilitate better decision-making.

1.4. Subject Matter

The system addresses the microfinance industry, which provides financial services to underserved populations. According to the Microfinance Gateway, a good number of adults worldwide remain unbanked, highlighting the need for accessible financial services. The global microfinance market is expected to grow significantly, driven by increasing demand for financial inclusion.

1.5. Notable Examples and Work Done by Competition

- **Successes:**
 - Pres Microfinance Institution Ltd. (Pres MFI) has achieved success by effectively addressing the financial needs of underserved populations in Buea, Cameroon. Its focus on providing tailored micro-loans and promoting financial literacy empowers local entrepreneurs to start and grow their businesses. Additionally, its commitment to community engagement and customer service fosters trust and loyalty among clients, enabling Pres MFI to create a positive impact on the local economy through job creation and sustainable development.

- Unics Microfinance is successful due to its targeted approach in addressing the financial needs of low-income individuals and small businesses in Cameroon. By offering tailored micro-loans and accessible savings products, along with comprehensive financial education, Unics empowers clients to improve their economic situations. Its strong community engagement and commitment to customer service foster trust and loyalty, enabling the institution to make a meaningful impact on local entrepreneurship and economic development.
- **Failures:**
 - Fame Microfinance in Buea, Cameroon, failed primarily due to poor management practices and inadequate risk assessment, which led to high loan default rates and financial losses. The institution's failure to comply with regulatory requirements resulted in penalties, further eroding its credibility. Consequently, clients lost trust, leading to a decline in new business and increased withdrawals, ultimately contributing to its collapse.

1.6. Areas Where Notable Rivals Failed

Analysis of competitors reveals several shortcomings:

- **Poor User Experience:** Complicated interfaces that deter users from engaging fully with the platform.
- **Insufficient Customer Support:** Delayed responses and inadequate assistance for users facing issues
- The community isn't well educated concerning the aspect of microfinance.

1.7. Overview of the System

The MIS will provide a suite of tools for managing client data, processing loans, tracking repayments, and generating comprehensive reports. Key objectives include improving operational efficiency, ensuring data integrity, and enhancing user experience.

1.8. Differentiating Factors

- **User-Centric Design:** An intuitive interface that simplifies navigation and increases user engagement.
- **Advanced Analytics:** Real-time data analysis tools to support decision-making and identify trends
- **Mobile Accessibility:** A mobile-friendly platform enabling users to access services anytime, anywhere.

1.9. Intended Audience

This SRS document is intended for: - Developers: To understand the technical requirements and system architecture.

- Testers: To develop test cases based on the defined requirements
- Project Managers: To ensure project alignment with stakeholder expectations.
- Stakeholders: To evaluate the project's feasibility and alignment with business goals.

2. Requirements Gathering and Data Analysis

2.1. Requirements Gathering

This section outlines the approach used to gather requirements for the Microfinance Management System.

- **Techniques Used:**

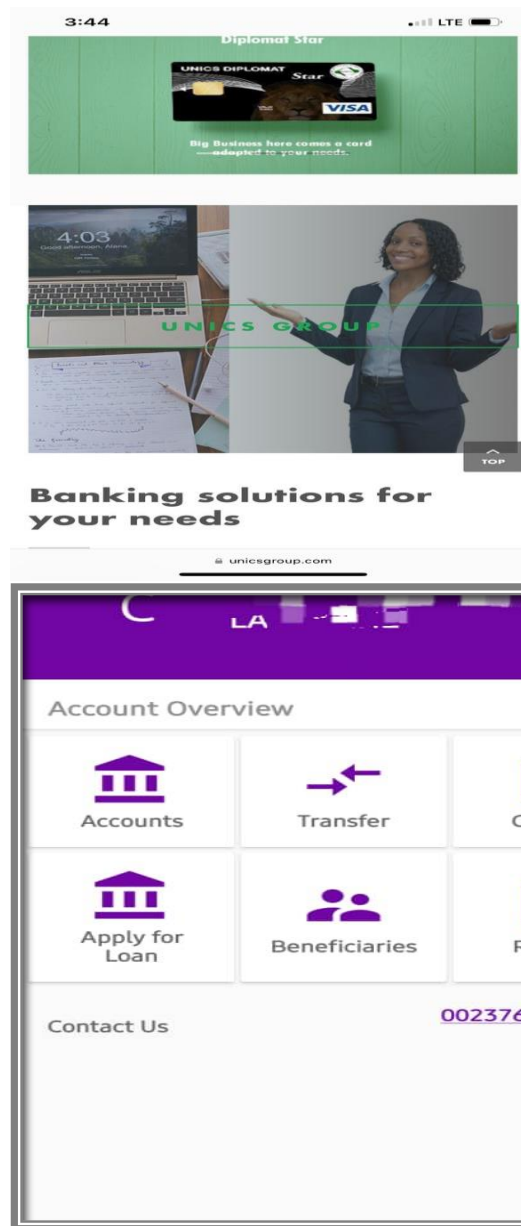
1. **Interviews:** Conducted structured interviews with stakeholders, including clients, loan officers, to understand their expectations and pain points.



Google Forms Surveys: Distributed online forms to collect quantitative and qualitative data from a wider audience, such as rural clients and field staff.

A screenshot of a Google Form titled 'Section 2 of 3'. The form contains a definition of a Microfinance Information System and a multiple-choice question about age. The definition text is: 'What is a Microfinance Information System? A micro finance information system is a software solution that helps micro finance institutions manage their operations, including client information, loan processing, and financial reporting. It enhances efficiency and decision-making in micro finance services. Untitled Section'. Below this is a description field labeled 'Description (optional)'. The multiple-choice question is 'What is your age?' with options: 15-25, 26-35, 36-45, 46-55, and 55+.

2. **Online Research:** Studied existing microfinance systems and industry best practices to ensure alignment with standard processes and user needs.



- **Stakeholders Involved:**
 - **Clients:** Individuals borrowing or saving money with the institution.

- **Field Staff:** Loan officers and customer service representatives managing day-to-day client interactions.
- **Locations Covered:**
 - Data was collected from both urban and rural branch locations to account for differences in client demographics, access to resources, and financial practices such as UNICS a category 2 microfinance, MC2 , MITACULL Buea

2.2. Data Analysis and Visualization

This section explains the data analysis process to refine the requirements and insights for the Microfinance Management System.

- **Data Analysis Requirements:**
 - **Data Sources:**
 - Responses from Google Forms (e.g., demographic data, client feedback).
 - Notes and transcripts from stakeholder interviews.
 - Industry data from reports and online research.

Timestamp	Email address	What is your age?	What is your gender	What is your level of education ?	What is your source of income?	How familiar are you with micro finance
06/12/2024 08:07:14		18	Female	Undergraduate	From my parents	70k
06/12/2024 08:46:59		18	Male		Parents	+50,000fcfa
06/12/2024 11:07:27		21	Male	Undergraduate	My parents	Between 30000frs to 35000frs
09/12/2024 06:34:41		18	Female	Undergraduate	Parents	60k
09/12/2024 06:39:01		50	Male	Master's degree	Thanks giving	
09/12/2024 06:43:02			Female	Bachelor's degree	Not fixe	Like 30k
09/12/2024 06:45:06		35	Female	Secondary education	My work	150000fcfa
09/12/2024 06:58:31		19	Female	Bachelor's degree	None	
09/12/2024 07:31:09		24	Male	Undergraduate	Teaching	
09/12/2024 08:10:17		19	Male	Bachelor's degree	Parents	
09/12/2024 08:57:32		25	Male	Bachelor's degree	Deft employed	30k

- **Data Formats:**
 - Structured data: Numerical and categorical survey responses.
- **Data Cleaning:**
 - Survey data was screened for duplicates, incomplete responses, and inconsistencies.
 - Qualitative responses were categorized and summarized for thematic analysis

Timestamp	Email address	What is your age?	What is your gender	What is your level of education ?	What is your source of income?	How familiar are you with micro finance
06/12/2024 08:07:14		18	Female	Undergraduate	From my parents	70k
06/12/2024 08:46:59		18	Male		Parents	+50,000fcfa
06/12/2024 11:07:27		21	Male	Undergraduate	My parents	Between 30000frs to 35000frs
09/12/2024 06:34:41		18	Female	Undergraduate	Parents	60k
09/12/2024 06:39:01		50	Male	Master's degree	Thanks giving	
09/12/2024 06:43:02			Female	Bachelor's degree	Not fixe	Like 30k
09/12/2024 06:45:06		35	Female	Secondary education	My work	150000fcfa
09/12/2024 06:58:31		19	Female	Bachelor's degree	None	
09/12/2024 07:31:09		24	Male	Undergraduate	Teaching	
09/12/2024 08:10:17		19	Male	Bachelor's degree	Parents	
09/12/2024 08:57:32		25	Male	Bachelor's degree	Deft employed	30k

The above is the uncleansed data collected. After treatment the table below was obtained

Micro finance management system (Responses)						
Timestamp	Email address	What is your age?	What is your gender	What is your level of education ?	What is your source of income?	How familiar are you with micro finance infor
06/12/2024 08:07:14		18-25	Female	Undergraduate	Parents	70,000
06/12/2024 08:46:59		18-25	Male	Undergraduate	Parents	50,000
06/12/2024 11:07:27		18-25	Male	Undergraduate	Parents	35,000
09/12/2024 06:34:41		18-25	Female	Undergraduate	Parents	60,000
09/12/2024 06:39:01		46-55	Male	Master's degree	Thanks giving	30,000
09/12/2024 06:43:02		18-25	Female	Bachelor's degree	others	30,000
09/12/2024 06:45:06		26-35	Female	Secondary education	salary	150,000
09/12/2024 06:58:31		18-25	Female	Bachelor's degree	salary	50,000
09/12/2024 07:31:09		18-25	Male	Undergraduate	civil servants	100,000
09/12/2024 08:10:17		18-25	Male	Bachelor's degree	Parents	40,000
09/12/2024 08:57:32		18-25	Male	Bachelor's degree	others	30,000
09/12/2024 09:53:35		46-55	Female	Bachelor's degree	civil servants	240,000
09/12/2024 10:47:24		18-25	Female	Bachelor's degree	Parents	20,000
09/12/2024 10:49:44		46-55	Female	Bachelor's degree	civil servant	100,000
09/12/2024 10:49:56		26-35	Male	Bachelor's degree	salary	120,000
09/12/2024 10:53:24		36-25	Male	Bachelor's degree	Salary	0
09/12/2024 15:21:38		18-25	Female	Bachelor's degree	salary	30,000
09/12/2024 21:22:19		46-55	Male	Bachelor's degree	Business	0

- **Analysis Techniques:**

1. **Descriptive Statistics:** Used to identify trends in client demographics, loan purposes, and repayment challenges.

What is your age?

☐ 15-25

☐ 26-35

☐ 36-45

☐ 46-55

☐ 55+

What is your gender

☐ Male

☐ Female

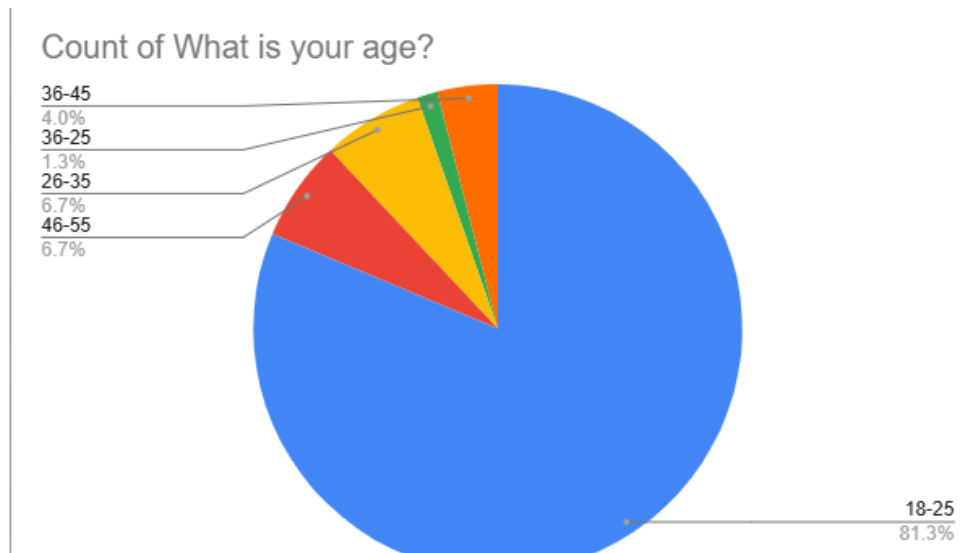
2. **Visualization Tools:** Employed tools like Excel to create user-friendly charts and graphs.

- **Visualization and Statistics:**

The following key statistics were obtained through data visualization:

- **Client Demographics:**

- 60% female, 40% male.
- Age distribution: 81% (18-25), 4.0% (26-45), 6.7% (46+).



- **Loan Purposes:**

- 40% business expansion, 30% farming, 20% education, 10% personal needs.

- **Repayment Issues:**

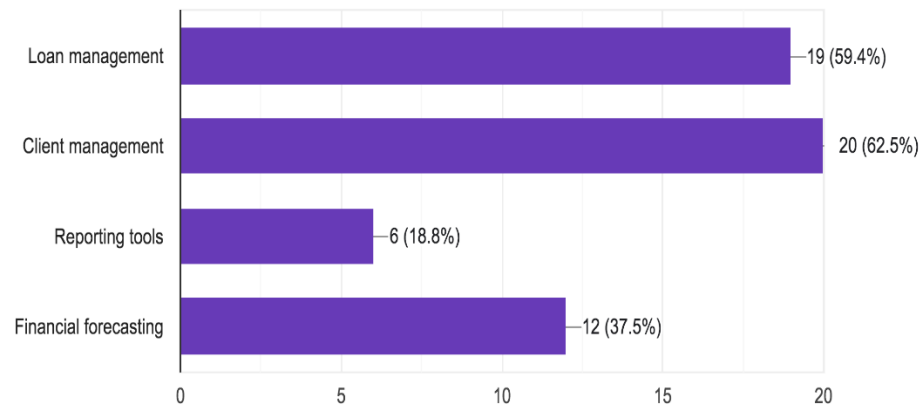
- 70% reported delays due to income instability.
- 20% attributed delays to a lack of financial literacy.

- **System Feature Preferences:**

- 59.4% of clients showed interest in automated loan tracking.

Which features do you think are most important in a micro finance information system? (Select all that apply)

32 responses



The requirements gathering and analysis process provided a comprehensive understanding of stakeholder needs. Insights from data analysis highlighted critical areas for system improvement, including the introduction of real-time loan tracking, streamlined loan applications, and financial literacy tools. Visualizations helped prioritize features that address the most significant challenges faced by clients and staff.

3. OVERALL DESCRIPTION

3.1. Product Perspective

The Microfinance Information System (MIS) is an integral part of a larger ecosystem of financial management software designed to streamline financial operations for microfinance institutions (MFIs). It interacts with the following components:

- **Core Banking Systems:** Facilitates loan issuance, savings, and payments by integrating with other banking platforms.
- **Customer Relationship Management (CRM):** Helps manage customer data for personalized services.
- **Accounting Systems:** Automates bookkeeping, transaction records, and financial reporting.
- **Third-party APIs:** Integrates with APIs for regulatory compliance, mobile money, and payment gateways.
- The MIS serves as a standalone system or can be deployed within a cloud-hosted architecture to facilitate accessibility.

3.2. Product Functions

The MIS will perform the following core functions:

a. Customer and Account Management:

- Register customers and maintain detailed profiles.
- Link customers with specific accounts, loans, or saving plans.

b. Loan Management:

- Automate loan disbursement workflows (approval, tracking, repayment schedules).
- Calculate interest rates and penalties for late payments.

c. Savings Management:

- Enable users to open saving accounts with interest accrual and withdrawal limits.

d. Reporting and Analytics:

- Generate financial statements (profit/loss, loan portfolio performance).
- Visualize data through dashboards and graphs.

e. Payment Processing:

- Process payments via mobile money, credit/debit cards, and wire transfers.

- Track and audit all payment transactions.

f. Compliance and Regulation:

- Track legal compliance for financial transactions.
- Generate compliance reports for regulatory authorities.

3.3. User Characteristics

The target users of the system can be categorized as follows:

a. Field Agents:

- Responsible for onboarding customers in remote areas.
- Basic computer knowledge and experience with mobile devices.
- **User Story:** “As a field agent, I need to register customer data and loan requests through a mobile application, even when offline.”

b. Branch Managers:

- Oversee loan disbursements, savings accounts, and reporting for specific branches.
- Intermediate knowledge of financial management tools.
- **User Story:** “As a branch manager, I need to monitor real-time customer payments and generate branch performance reports.”

c. Customers:

- Borrowers or savers interacting with the system via mobile applications or SMS services.
- Minimal technical skills but familiarity with mobile wallets and USSD codes.
- **User Story:** “As a customer, I want to check my account balance or payment due date from my phone.”

d. System Administrators:

- Install, maintain, and troubleshoot the software.
- Advanced knowledge of database management and server configuration.
- **User Story:** “As a system administrator, I need access to detailed logs to debug performance issues.”

3.4. General Constraints

The system will operate under the following constraints:

1. Performance Requirements:

- The system should handle 1,000+ concurrent users with minimal latency (<1 second for most operations).

- Data integrity and consistency must be ensured during transactions.

2. Security Requirements:

- Comply with industry security standards such as ISO 27001 and GDPR.
- Provide secure authentication (e.g., two-factor authentication and encrypted connections).

3. Hardware/Software Limitations:

- Optimized to run on low-end devices for field agents and customers in remote areas.
- Must support offline functionality with automatic syncing once online.

4. Mobile Support:

- Fully responsive web interface and dedicated mobile apps for Android/iOS.

5. Regulatory Compliance:

- Must align with local financial laws (e.g., anti-money laundering requirements).

3.5. Assumptions and Dependencies

a. Assumptions:

- Field agents and branch offices will have reliable internet access for periodic syncing.
- Customers are familiar with using mobile wallets or have mobile network coverage.
- Integration with third-party APIs (e.g., mobile money providers) is stable and secure.

b. Dependencies:

- Availability of regulatory compliance updates (e.g., tax codes or financial reporting rules).
- Dependence on external APIs for critical features such as payment processing and credit checks.
- Continuous software updates and support to keep the system compatible with newer OS platforms and devices.

4. External Interface Requirements

The external interface requirements represent the interfaces or the layout through which users will interact with the Microfinance Management System (MFMS). The goal is to define how the system will communicate with users, other systems, and hardware devices.

4.1. User Interface

This section describes the user interfaces of the MFMS, including:

Client Portal: A web-based portal where clients can access their account information, view transaction history, and perform other tasks.

Staff Portal: A web-based portal where MFI staff can manage client information, process loan applications, and perform other tasks.

Reporting Dashboard: A web-based dashboard where credit managers and other stakeholders can access reports and analytics on client behavior and loan performance.

4.2. Hardware Interfaces

This section describes any hardware interfaces required by the MFMS. Hardware interfaces may include:

Printers: For printing loan documents, receipts, and other reports.

Scanners: For scanning loan documents and other paperwork

Sensors: For tracking and monitoring equipment or inventory.

4.3. Software Interfaces

This section describes the software interfaces required by the MFMS. Software interfaces may include:

Accounting System Integration: The MFMS will integrate with existing accounting systems to exchange financial data.

Payment Gateway Integration: The MFMS will integrate with payment gateways to facilitate loan repayments and savings deposits.

Data Analytics Integration: The MFMS will integrate with data analytics tools to provide insights and reports on client behavior and loan performance.

These software interfaces will enable the MFMS to communicate with other systems, exchange data, and provide a user experience

5. DESIGN CONSTRAINTS

5.1. Standards

- a. **Software Quality:** The system must work well, be reliable, secure, and easy to use.
- b. **Payment Security:** If payments are made through the system, customer information must be protected using global payment security rules.
- c. **Accounting Rules:** The system must follow local financial reporting rules used in Cameroon.
- d. **Cameroon Laws:** The system must respect local laws about data privacy and cyber security.
- e. **User Friendliness:** The system should be simple and easy to use, keeping in mind the skills of users in Buea.

5.2. Hardware Limitations

- a. **Limited Server Power:** The computers or servers used may not have a lot of storage, memory, or processing speed, so the system must run efficiently.
- b. **Unstable Electricity:** Power cuts are common, so the system should work with backup power or save data to avoid loss.
- c. **Basic Computers:** Staff may use simple computers with limited speed and memory.
- d. **Mobile Devices:** The system should also work on cheap smartphones with limited storage and memory.
- e. **Internet Issues:** Internet might be slow or unreliable, so the system should work offline when needed and sync when the connection is restored.

5.3. Software Limitations

1. Operating Systems:

- The system must work on Windows computers in offices and Android phones for field agents.
- This limits the tools we can use to create the system.

2. Database:

- The database must be lightweight and affordable (e.g., MySQL or PostgreSQL) to save costs and work well on limited hardware.

3. Development Tools for Each Phase:

- Gathering Requirements: Use tools like Google Docs to write down what the system should do.
- Why: It's easy to use and allows collaboration.
- Designing: Use tools like Lucid Chart to draw system diagrams (e.g., workflows, database designs).
- Why: It helps explain how the system will work.
- Building: Use coding tools like Visual Studio Code and frameworks like Django (Python) or Spring Boot (Java).
- Why: These tools make it easy to develop systems that can run on both computers and mobile phones.
- Testing: Use tools like Postman for testing how the system handles data or Selenium for testing the user interface.
- Why: These tools ensure the system works properly without needing too much manual testing.
- Deployment: Use Docker (to package the system) and host it on a cloud platform like AWS.
- Why: Docker makes it easier to install the system anywhere, and AWS ensures it can scale up when needed.

4. Growth Limitations:

- At first, the system will only handle a small number of users and transactions, but it should be easy to upgrade later.

5. Security:

- While the system may not use very advanced security features due to hardware limits, basic protections like data encryption will still be included.

6. ADDITIONAL REQUIREMENTS

6.1. Timeline and Duration

Project Timeline:

- **Week 1-2:** Planning and requirement gathering
- **Week 3-6:** System design (user interface, database structure, and workflows)
- **Week 7-12:** Development of core features (account management, loan processing, transaction recording)
- **Week 13-15:** Testing and debugging
- **Week 16:** Deployment and training

Total Time Duration: Approximately 4 months (16 weeks)

6.2. Project Cost and Worth

Estimated Costs:

1. Software Development

- Design, coding, testing, deployment: 3,000,000 FCFA

2. Infrastructure

- Hosting, servers, and security: 1,000,000 FCFA

3. Training and Support

- Staff training and ongoing support: 500,000 FCFA

4. Miscellaneous

- Equipment, licenses, and unforeseen expenses: 500,000 FCFA

Total Estimated Cost: 5,000,000 FCFA

Potential Financial Value (2-3 years)

- **Revenue from clients:** Estimated at 2,000,000 FCFA per year.
- **Cost savings:** By automating processes, it may save about 1,000,000 FCFA annually.

Estimated Value Over 3 Years: 9,000,000 FCFA

Profit Over 3 Years: 4,000,000 FCFA (9,000,000 - 5,000,000 FCFA)

6.3. List of Staff Members

1. **Project Manager:** Oversees the project and ensures timelines are met.
2. **System Analyst:** Identifies the system requirements and ensures alignment with business needs.
3. **Developers:** Build the system (both backend and frontend).
4. **Database Administrator:** Manages the database structure and data security.
5. **Tester/Quality Assurance:** Ensures the system functions as expected without errors.
6. **Trainer/Support Staff:** Provides user training and ongoing technical support.

REFERENCES

- **Microfinance Gateway:** Provides insights into global microfinance trends, including the importance of financial inclusion.

Link: <https://www.findevgateway.org>

- **CGAP (Consultative Group to Assist the Poor):** Explores microfinance best practices and industry failures.

Link: <https://www.cgap.org>

- **IEEE 29148-2018:** International standard for Software Requirements Specifications (used for structuring SRS documents).

Link: <https://ieeexplore.ieee.org>

- **LucidChart:** Commonly used for designing system diagrams, workflows, and database relationships as referenced in the SRS.

Link: <https://www.lucidchart.com>

- **Pres Microfinance:** Referenced as a success case for microfinance practices in Cameroon.

- **Unics Microfinance:** Also highlighted as a success in addressing financial needs.

Link: <http://unicameroon.net>

- **Google Forms for Data Collection:**

Link: <https://forms.google.com>

- **Visual Studio Code:** <https://code.visualstudio.com>
- **Spring Boot:** <https://spring.io/projects/spring-boot>
- **AWS for Cloud Deployment:** Used for hosting and scaling MIS applications.

Link: <https://aws.amazon.com>

DESIGN OF OUR MICROFINANCE MANAGEMENT SYTEM

7. DESIGN THINKING PROCESS

Design thinking is a user-centered problem-solving approach that focuses on understanding users' needs, brainstorming creative solutions, and testing ideas before final implementation. This report outlines the application of design thinking in developing a microfinance management system named 'SMOLLON'.

The design thinking process consists of five iterative stages: Empathize, Define, Ideate, Prototype, and Test. By following this structured methodology, we ensure that the system meets the financial and technological needs of its users while fostering financial inclusion and efficiency.

7.1. DESIGN THINKING PROCESS OVERVIEW

EMPATHIZE

To build a system that effectively serves microfinance users, we conducted extensive user research through surveys, interviews, and observations. The key stakeholders include:

Customers: individuals seeking for loans, owning accounts for business or growth

Administrators: system manager overseeing transactions and compliance

During our requirement gathering we discovered that customer found it difficult and complex to perform certain task at micro finance such as account creation, loan application and payment processes and also had to wait for long queues at microfinance institution before having a service done.

Also, administrators need and efficient and automated tracking system for loan and repayment, using papers and books becomes tiring to search for customer history



7.2. DEFINE: Framing the problem

After our research we came out with the following problem statement
“How might we design a simple, easy to use, and accessible microfinance system that simplifies microfinance services?”

- **Complex loan processing:** Lengthy documentation and approval times.
- **Limited accessibility:** Borrowers in rural areas lack technical proficiency.
- **Lack of financial awareness:** Users struggle with repayment tracking and financial planning.

7.3. IDEATE: Generate solutions

A brainstorming session was conducted to explore innovative solutions.
The key ideas included:

- A mobile-friendly interface for easy access to microfinance services.
- Simplified loan application forms with easily understandable words
- Automated notifications for loan approval, repayment reminders, and due dates.
- A user dashboard to track loans, payments, and financial history.

7.4. PROTOTYPE DEVELOPING MODEL

Interactive prototype was built using figma to test and validate different UI layout and functionalities.

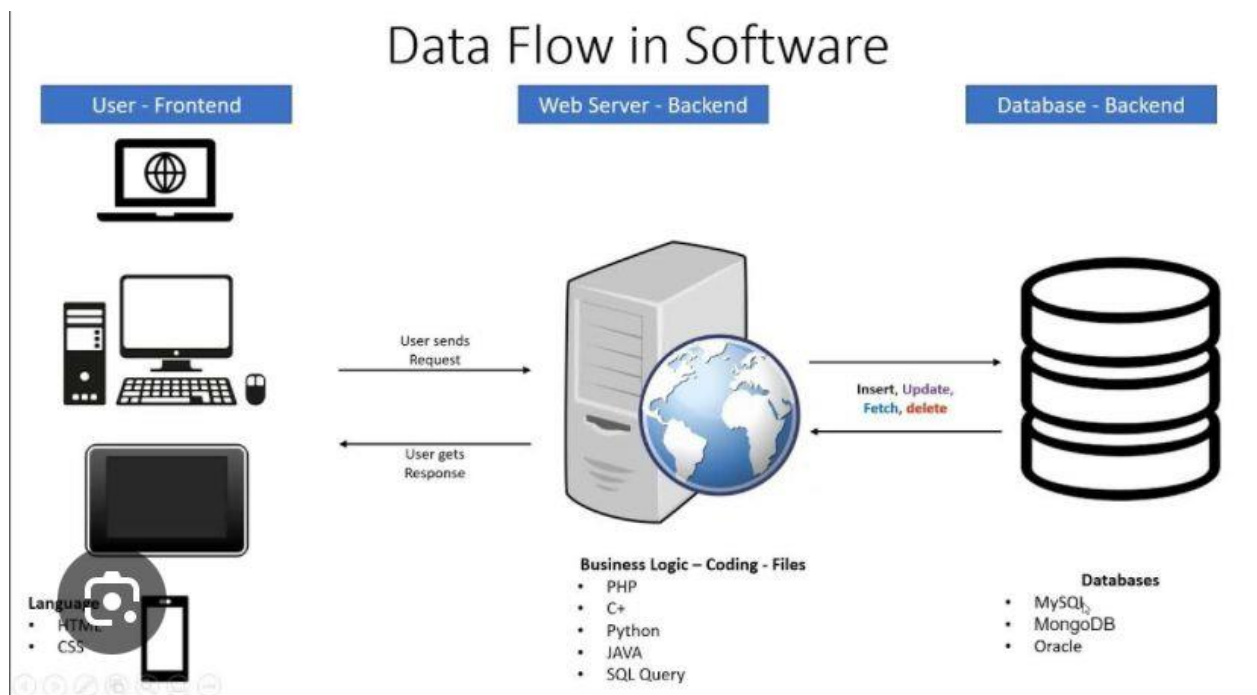
7.5. TEST

User testing was conducted on the prototype to gather insights on usability and navigation. Based on result, adjustments was done to color, buttons.

8. SYSTEM AND ARCHITECTURAL DESIGNS

8.1. SYSTEM OVERVIEW

The system consist of a frontend UI (React js), a backend API (Node js with express) and a MySQL database for data storage. The application allows the user to register, apply for loan, track payment, manage transactions



8.2. ARCHITECTURAL SYLE/PATTERN

A layered is useful for a microfinance management system because it organizes the software into separate layers, improving scalability, maintainability, security and flexibility

High-Level Structure of Microfinance Management System

1. User Interface (UI)

Web Portal

- Customer Dashboard
- Loan Application Forms
- Payment Portal

- Account Management

Admin Dashboard

- User Management
- Loan Management
- Reports and Analytics

2. Backend Services

Authentication Service

- User registration and login
- Role-based access control

Loan Management Service

- Loan application processing
- Approval workflows
- Disbursement and repayment tracking

Customer Relationship Management (CRM) Service

- Customer interaction tracking
- Feedback management

Reporting Service

- Financial reports
- Transaction history
- Compliance reports

3. Database Layer

User Database

- Stores user profiles, authentication details, and user credentials.

Loan Database

- Stores loan applications, statuses, and payment schedules

Transaction Database

- Tracks all financial transactions and history

Analytics Database

- Stores data for reporting and analytics purposes

4. API Layer

RESTful API

- Facilitates communication between the frontend and backend services;
- Contains Endpoints for user operations, loan management, and reporting

5. Security Layer

- **Encryption:** Secure transfer of data (SSL/TLS)
- **Access Control:** Role-based access management
- **Audit Logs:** Tracking user actions for compliance

6. Integration Layer

Payment Gateway Integration

- For processing payments and disbursements

Third-party Services

- Integration with credit bureaus, SMS gateways, etc.

8.3. DEPLOYMENT DIAGRAM

8.4. SYSTEM DECOMPOSITION

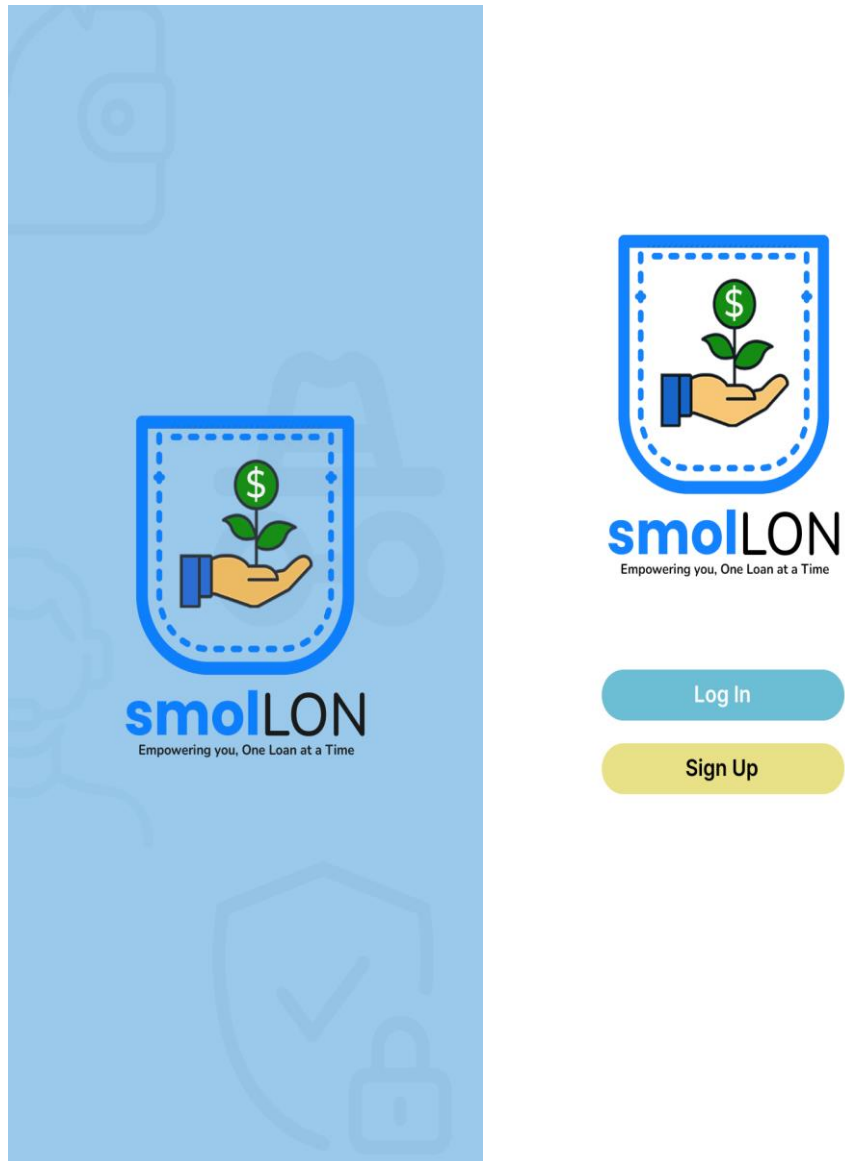
Component and module design for our MFMS

- Breaking down the system to smaller modules and assigning their responsibilities:
 1. **Authentication module:** Handles user login, registration, password management, and access control
 2. **Client management module:** Manages client information (personal details, contact info)
 3. **Saving management module:** Manages savings accounts, deposits, withdrawals, and interest calculations
 4. **Loan origination module:** Handles loan applications, loan approval, and disbursement. (**This module is solely responsible for handling loan applications**)
 5. **Loan portfolio management Module:** tracks loan repayments, calculate interests, generate loan reports (**while this module tracks the loan performance**).
 6. **SMS/Email notification module:** Handles automated notification and reminders to clients.

9. UI/UX DESIGN

9.1. User interface design

9.1.1. Screen mockups and wireframes



Welcome

Log In

SignUp

Username or Email

example@example.com



Password

Log In

[Forgot Password?](#)

Sign Up

Or Sign Up with



Create Account

Log In

SignUp

Full Name

NAPE

Email

example@example.com

Mobile Number

+237 699 999 999

Password

Confirm Password

By continuing, you agree to [Terms of Use and Privacy Policy.](#)

Sign Up

Already have an account? [Log In](#)

Welcome,
NAPE



Loans

Available Balance

200,000.00 XAF



Request Loan



Withdraw



Top - Up

Current Balance

200,000.00 XAF



Open Loans



Close Loans

Recent Transactions

See All >



Refund
02/10/2024

20,000 XAF



Student Loan
02/03/2024

20,000 XAF



Default Penalty
07/09/2023

500 XAF



Refund
03/02/2023

10,000 XAF



Student Loan
20/09/2022

10,000 XAF

CW1032658-56
20,000 XAF



CW7895432-06
20,000 XAF



CW1025455-04
10,000 XAF



Home



Loans



Transactions



Account



Home



Loans



Transactions



Account

My Account



Show Dashboard Account Balance



Enable Notifications



Top-Up Account



Apply Loan



Email Notification



Contact Support



Privacy Policy



Reset Password



Sign Out

Transactions

Transaction Details



Refund

02/10/2024

20,000 XAF



Student Loan

02/03/2024

20,000 XAF



Default Penalty

07/09/2023

500 XAF



Refund

03/02/2023

10,000 XAF



Student Loan

29/09/2022

10,000 XAF



Student Loan

02/03/2024

20,000 XAF



Default Penalty

07/09/2023

500 XAF



Refund

03/02/2023

10,000 XAF



Home



Loans



Transactions



Account



Home



Loans



Transactions



Account

< **Top-Up Account**



How much do you want to Deposit?

Phone Number

Top-Up

Go Phone app and Dial *126# for
MTN Mobile Money or #150# for
Orange Money

< Apply for a Loan

How much do you want to borrow?

How long do you want the loan for?

APPLY

< Personal Information

Title

Gender

First Name

Last Name

Date of Birth

Continue

< Employment Information

Place of Work

Bank Name

Bank Account

Continue

< Employment Information

Place of Work

Ba

Ba



Completed!

You have successfully applied for a loan. A response will be sent shortly.

< Loan Breakdown

Amount	Duration
20,000 XAF	7 Months
Monthly Repayment	Loan Release Date
5,000 XAF	02/03/2024
Due Date	Interest per day
02/10/2024	0.25%

By clicking Continue, you agree to our
[Terms of Service](#) and [Privacy Policy](#)

Continue

CW1032658-56

< 200,000.00 XAF >

Amount	Duration
20,000 XAF	7 Months
Monthly Repayment	Loan Release Date
5,000 XAF	02/03/2024
Due Date	Balance
02/10/2024	15,000 XAF

Loan Status

Active

68% paid



Repayment History

-  March Repayment 20,000 XAF
02/03/2024
-  October Repayment 20,000 XAF
02/10/2024
-  September Repayment 10,000 XAF
07/09/2023
-  February Repayment 10,000 XAF
03/02/2023



Home



Loans



Transactions



Account

9.1.2. USER INTERACTION FLOW

When the user launch the mobile app, he is asked to sign up if he does not have an account. To sign up the user have to enter some credentials that would be save in the database of the system. If the user already had an account, he can then log in with user name/ email and password and credentials validated by the system. If log in successful the user is then brought to the home page with a welcome message, otherwise an error message will be sent. When the user launch the mobile app, he is asked to sign up if he does not have an account. To sign up the user have to enter some credentials that would be save in the database of the system. If the user already had an account, he can then log in with user name/ email and password and credentials validated by the system. If log in successful the user is then brought to the home page with a welcome message, otherwise an error message will be sent.

At the bottom an activity bar with icons for home, loan, transaction, and account Clicking on loan will allow you to view your loan status and repayment history. Also when you click on the transaction icon you can see all your transaction history both for loan payment and deposits. On the account button, you will see a dashboard where you can deposit, apply loan etc.

9.1.3. STYLE GUIDE

The design follows a consistent and meaningful color palette to enhance usability and readability

PRIMARY COLOR

Blue: used for headers and buttons, and progress indicator to represent trust, security, and professionalism

White and light gray: background colors that provide cleanness and purity

SECONDARY COLORS

Mint green used for inputs fields, creating a readable interface

Beige/Yellow used to grab attention as a call to action

ACCENT COLORS

Red used for icons related to actions such as loan request and withdrawals, ensuring important actions and clear visuals

TYPOGRAPHY

The UI uses, readable fonts to ensure clarity and hierarchy:
Large fonts for important financial data ex available balance
Standard size for labels

LAYOUT

The UI include dashboard screen, displays for available balance, navigation bar with icons, toggle buttons to switch from log in to sign up etc.

9.2. USER EXPERIENCE DESIGN

9.2.1. User Personas

“As a customer, I want to check my account balance or payment due date from my phone.”

“As a customer, I will like to apply for a loan from my phone”

“As customer, it will like to receive notifications, to know my loan status”

“As a customer, I will like to make a deposit from my phone”

“AS a customer, I will like a system with more visuals and easy to understand to carry out my transactions”

9.2.2. USE CASES

When the user launch the mobile app, he is asked to sign up if he does not have an account. To sign up the user have to enter some credentials that would be save in the database of the system. If the user already had an account, he can then log in with user name/ email and password and credentials validated by the system. If log in successful the user is then brought to the home page with a welcome message, otherwise an error message will be sent.

At the bottom an activity bar with icons for home, loan, transaction, and account Clicking on loan will allow you to view your loan status and repayment history. Also when you click on the transaction icon you can see all your transaction history both for loan payment and deposits. On the account button, you will see a dashboard where you can deposit, apply loan etc.

To apply for loan, you need to enter the amount you want to borrow and the due date, fill in some information and submit and wait for approval. After you will see if it was validated or not as a notification

More details about the use cases will be given subsequently

9.2.3. INFORMATION ARCHITECTURE

The information is structured into sections;

- Authentication
- Loan management
- Transactions
- User profiles

10. USABILITY DESIGN

10.1. USABILITY PRINCIPLES

1. **Well defined interfaces:** Each module expose a well defined interface for other modules to interact with. For example, the client management module has an interface that allows other modules to:
 - Retrieve information
 - Create new client record
 - Update client details
2. The software is easy to learn with icons, clear buttons and straight to the point instructions

10.2. ACCESSIBILITY CONSTRAINTS

- Easy to read icons to facilitate understanding
- Straight forward instructions to ease understanding to illiterates

10.3. USABILITY Testing PLAN

Implement thorough testing strategies:

- Unit tests i.e. testing individual modules
- Integration test i.e. testing how modules interact with each other.

11.DETAILED DESIGN

11.1. DATA DESIGN

1. Database Schema (Tables, Relationships, Constraints) This scheme holds the outline of the database.

Key Entities

- Clients: People who borrow from or save with the microfinance institution.
- Loans: Records of credit given to clients.
- Payments: Records of loan repayments.
- Staff: Employees managing loans and clients.
- Savings Accounts: accounts for clients who save

Table Structure Clients Table:

- ClientID (Primary Key)
- Name
- ContactInfo (Phone, Address, Email)
- DateOfBirth
- Gender
- NationalID

Loans Table:

- LoanID (Primary Key)
- ClientID (Foreign Key Referencing Clients)
- LoanAmount
- InterestRate
- LoanTerm (Duration)
- LoanStatus (Active, Paid, Defaulted)
- DisbursementDate

Payments Table:

- PaymentID (Primary Key)
- LoanID (Foreign Key Referencing Loans)
- PaymentDate
- PaymentAmount
- PaymentMethod Staff Table:
- StaffID (Primary Key)
- Name
- Role (Loan Officer, Manager, etc.)
- ContactInfo

Savings Table:

- SavingsID (Primary Key)
- ClientID (Foreign Key referencing Clients)

- Balance
- LastTransactionDate

Relationships One-to-Many:

- A client can have multiple loans.
- A loan can have multiple payments.

One-to-One:

- Each client can have one savings account.

Many-to-Many:

- Staff may manage multiple clients, and clients can interact with multiple staff members. This requires a junction table.

```
mysql> SELECT * FROM Customers;
+-----+-----+-----+-----+-----+-----+-----+
| CustomerID | FirstName | LastName | DateOfBirth | PhoneNumber | Email | Address |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Estelle | Smith | 1990-01-01 | 123-456-7890 | estelle@example.com | 123 Main St |
| 2 | Randy | Johnson | 1985-05-15 | 234-567-8901 | randy@example.com | 456 Elm St |
| 3 | Max | Thompson | 1992-03-20 | 345-678-9012 | max@example.com | 789 Pine St |
| 4 | Neo | Anderson | 1988-07-30 | 456-789-0123 | neo@example.com | 321 Oak St |
| 5 | Larry | Brown | 1983-12-12 | 567-890-1234 | larry@example.com | 654 Maple St |
| 6 | KDPO | Davis | 1995-11-11 | 678-901-2345 | kdpo@example.com | 987 Cedar St |
| 7 | Rawlings | The Clearant | 1990-09-09 | 789-012-3456 | rawlings@example.com | 159 Birch St |
+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> SELECT
  -> Loans.LoanID,
  -> Loans.LoanType,
  -> Loans.LoanAmount,
  -> Loans.Status,
  -> Customers.FirstName,
  -> Customers.LastName,
  -> Customers.PhoneNumber
  -> FROM
  -> Loans
  -> JOIN
  -> Customers ON Loans.CustomerID = Customers.CustomerID;
Empty set (0.01 sec)

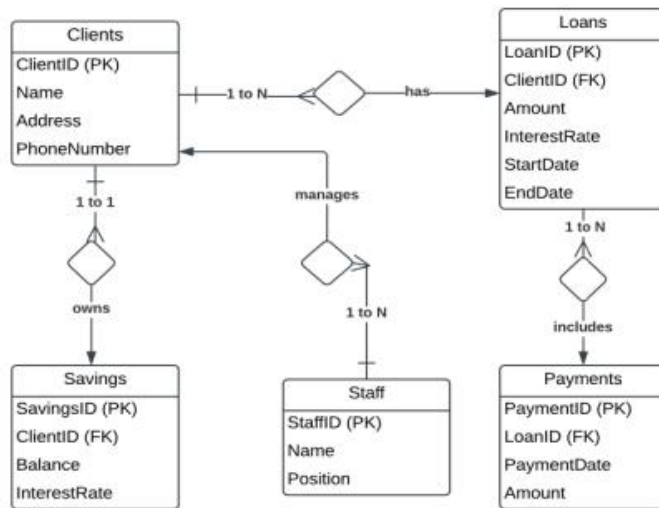
mysql> SELECT
  -> Payments.PaymentID,
  -> Payments.PaymentDate,
  -> Payments.AmountPaid,
  -> Payments.PaymentMethod
  -> FROM
  -> Payments
  -> WHERE
  -> Payments.LoanID = 1; -- Replace with the specific LoanID
Empty set (0.00 sec)

mysql> SELECT
  -> Transactions.TransactionID,
  -> Transactions.TransactionType,
  -> Transactions.Amount,
  -> Transactions.TransactionDate
  -> FROM
  -> Transactions
  -> WHERE
  -> Transactions.CustomerID = 1; -- Replace with the specific CustomerID
Empty set (0.00 sec)
```

2. Entity-Relationship Diagram

ERD visually represents the relationships between entities.

- Clients → Loans:** A client can have multiple loans, but each loan belongs to only one client.
- Loans → Payments:** A loan can have multiple payments, but each payment is associated with only one loan.
- Clients → Savings:** A client can have multiple savings accounts, but each savings account belongs to only one client
- Staff → Clients:** A staff member can manage many clients.



11.2. SECURITY DESIGN

Security Design for our Microfinance System

1. Authentication and Role-Based Access Control (RBAC)

i. Authentication:

- Require users to log in with a password and a second layer of security, like a code sent to their phone or email (two-step login).
- Use fingerprints or face recognition for mobile app users if possible.
- Lock accounts temporarily after too many wrong password attempts to stop hackers.

ii. Role-Based Access:

- Give each person a role based on their job, like Admin, Loan Officer.
- Each role should only access what they need: for example Admin: Access everything, including managing users and reports .Can add new employees to the system, change user roles, and view all customer and financial data.
- Only give people access to the parts of the system they need to do their job.

2. Data Encryption and Secure Communication

i. Encrypt Data:

- Protect sensitive data like customer information and transactions by converting it into unreadable code (encryption).
- Store passwords safely by turning them into secure, one-way codes (hashing).

ii. Secure Communication:

- Use HTTPS (the padlock icon in the browser) to protect data sent between users and the system. When a customer logs into their account using a browser or mobile app, the system uses HTTPS. This ensures that all data they send (like their login credentials) or receive (like their account balance) is encrypted and cannot be intercepted by hackers.
- Encrypt messages, especially for financial transactions or personal updates, so no one can read them during transmission.

iii. Manage Keys:

- Keep the “keys” to unlock encrypted data in a secure place like a cloud service (e.g., AWS or Azure).
- Change these keys regularly to keep them safe.

3. Protect Against System Weaknesses

i. Find Weak Spots:

- Test the system regularly to check for vulnerabilities that hackers could exploit. Use security tools to do this.
- Fix any problems found during these tests quickly.

ii. Check User Input:

- Make sure the system only accepts safe information from users. This prevents hackers from inserting harmful code.

iii. Keep Software Updated:

- Regularly update all software and tools to close security holes.
- Use tools that automatically check for outdated components

iv. Firewalls and Monitoring:

- Use firewalls to block harmful traffic
- Install systems that can detect and stop unusual or dangerous activity.

v. Backups:

- Save copies of all data regularly, so it can be restored if something goes wrong.

- Protect these backups with encryption and store them safely in multiple locations

12.DESIGN METHODOLOGIES

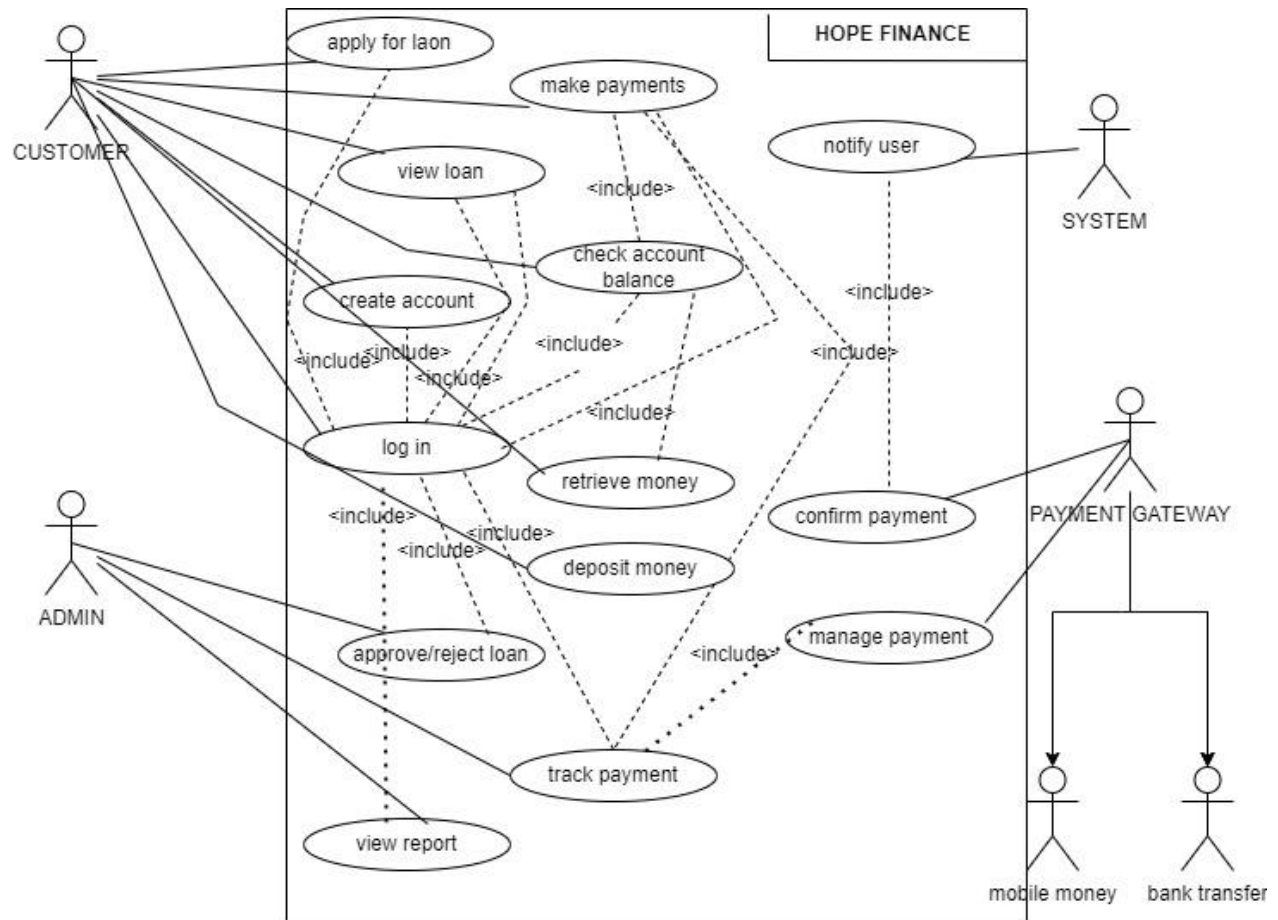
12.1. Object oriented modeling design

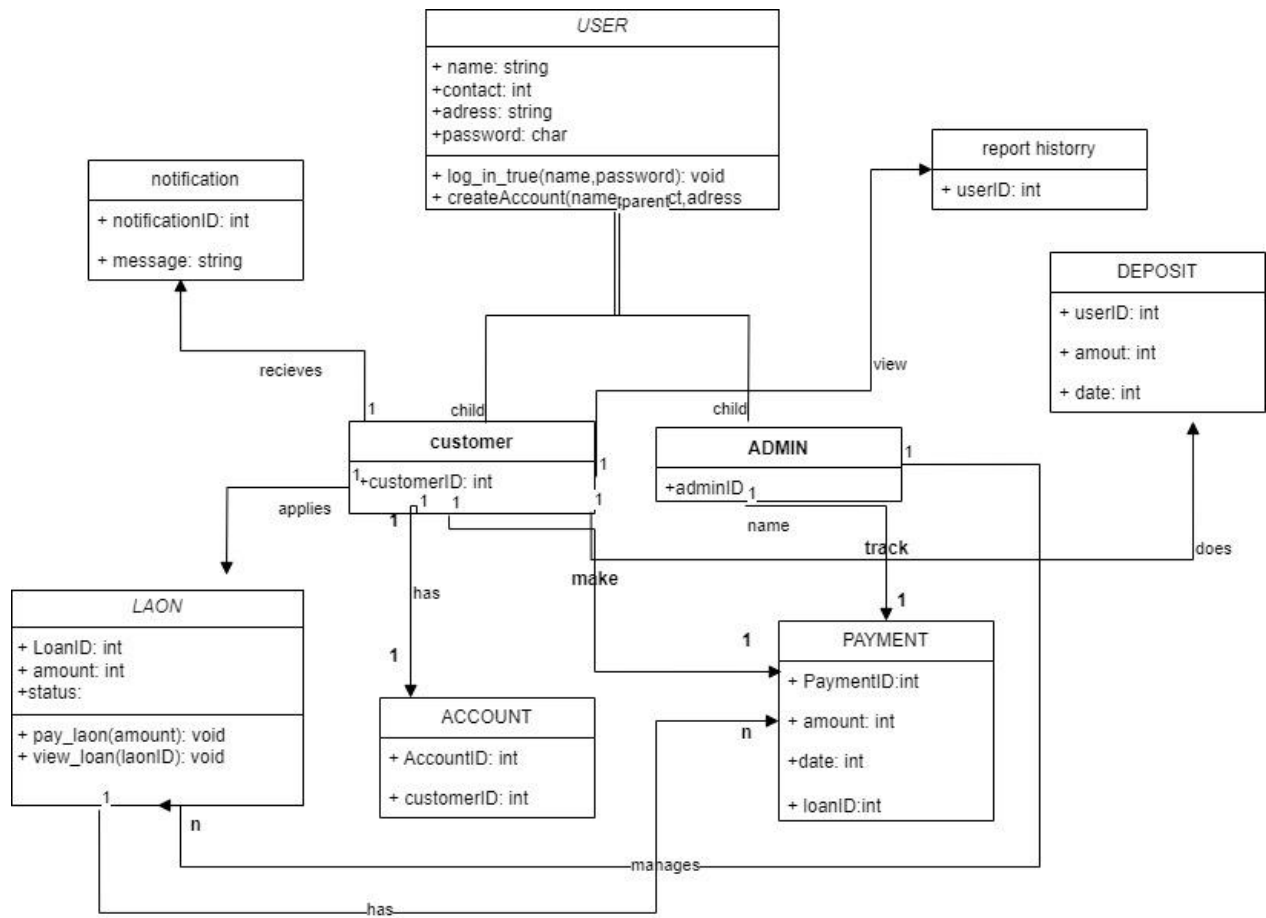
- a. **Modular design:** Each class has a single responsibility which performs a single task. This promotes modularity and maintain the code.
- b. **Inheritance:** each class is open for modification without modifying the existing code
- c. **Dependency inversion principle:** low level classes such as notification depends on high level classes such as loan

Below are some uml diagram of our MFMS

Use case diagram

Class diagram

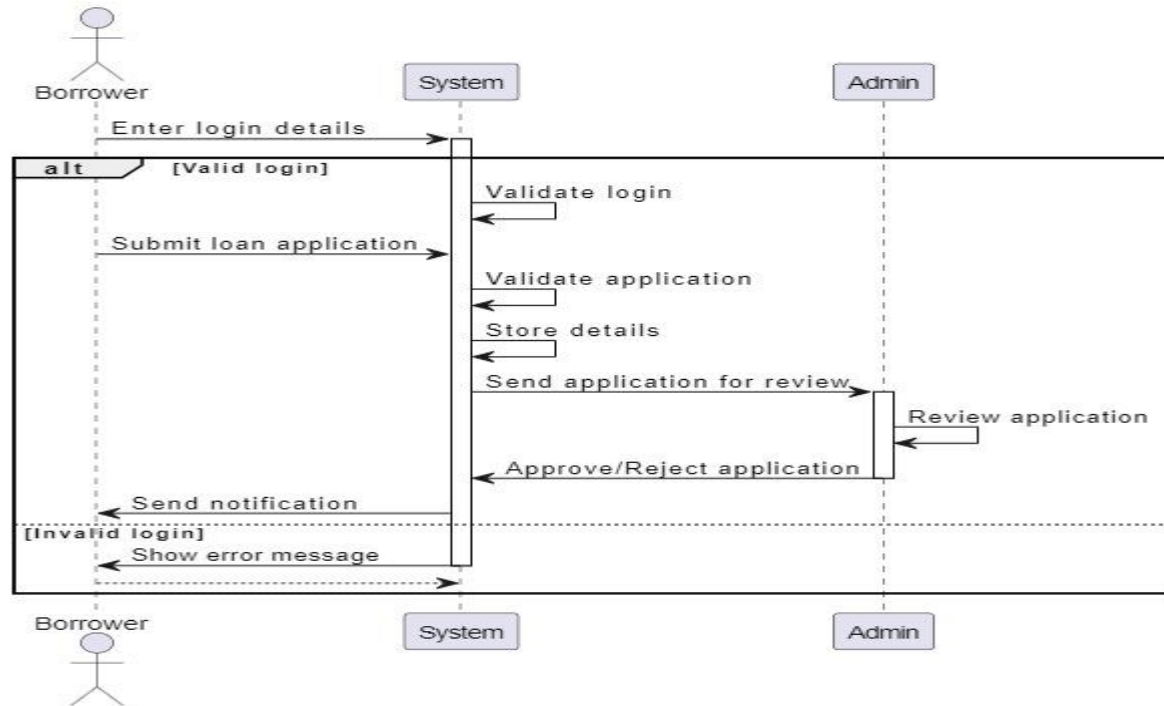




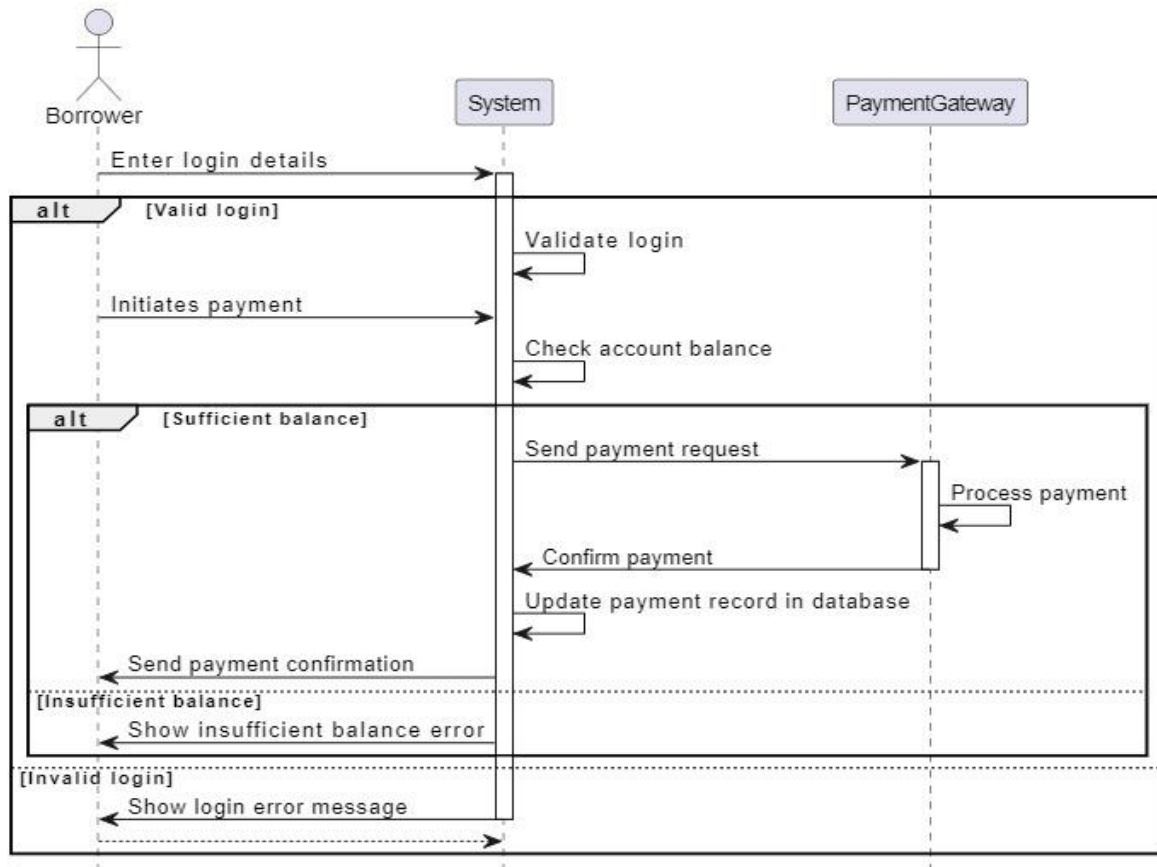
12.2. Event-driven design

Sequence diagrams

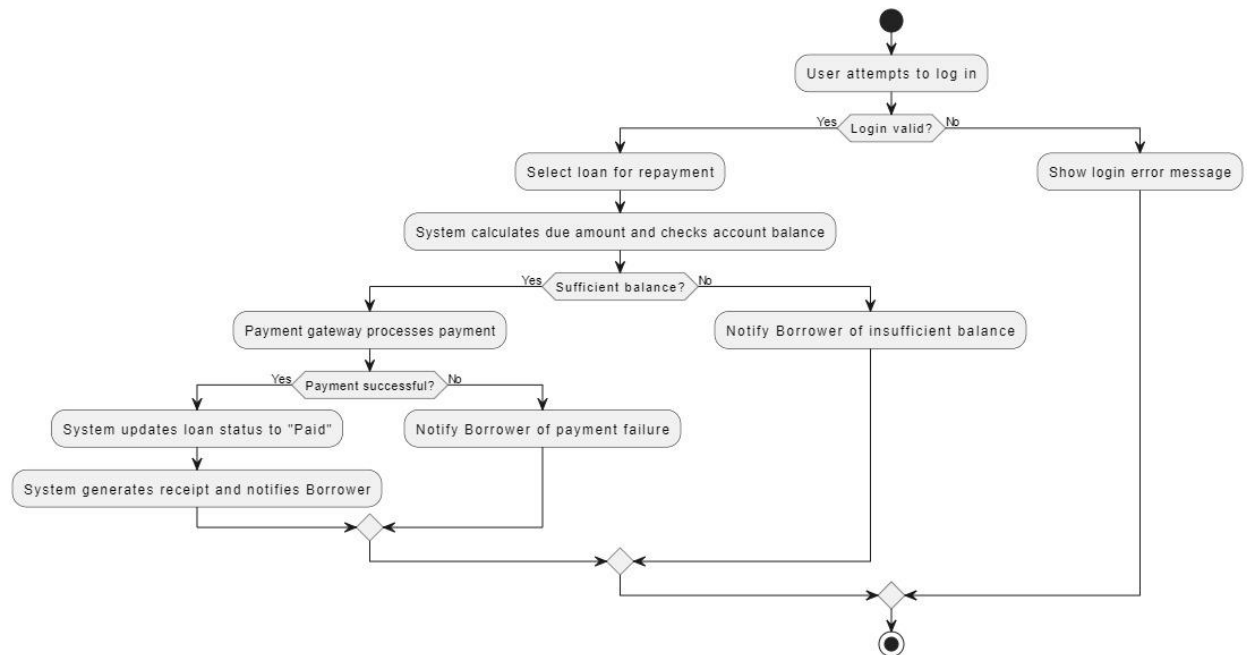
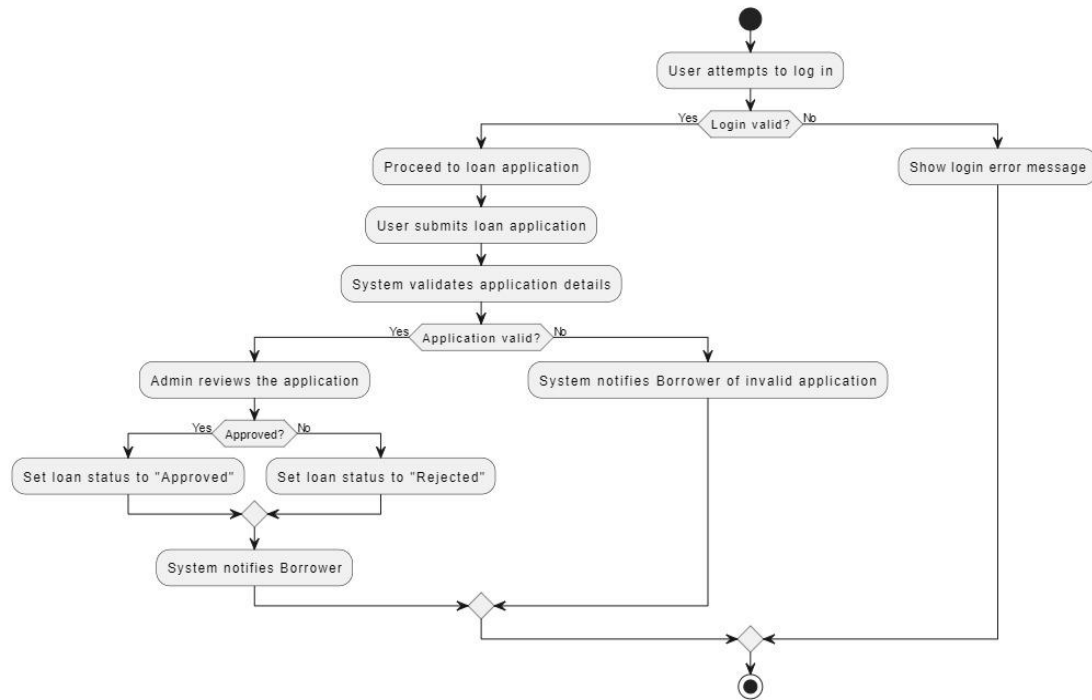
a. Loan application process



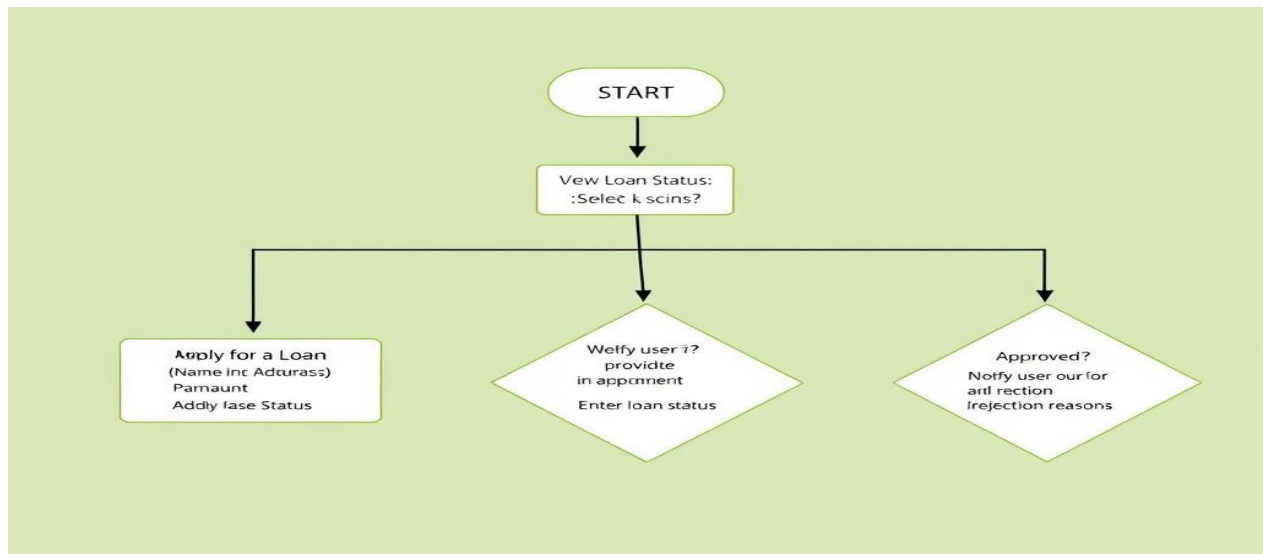
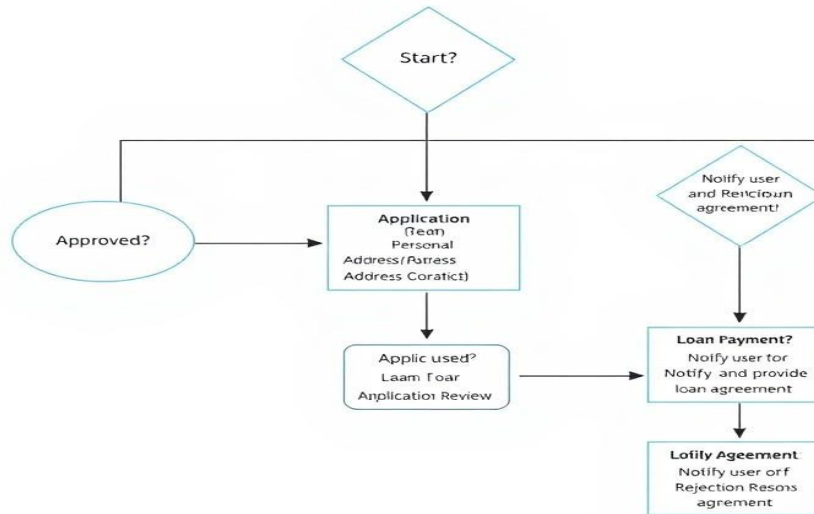
b. Payment process



Activity diagram



12.3. Data flow diagrams



13.Non functional requirements.

1. **Maintainability:** a well structured and modular code that is easier to understand
2. **Scalability:** can start with 1000 users and easily accommodate growth with time due to modularity and cloud hosting.
3. **Usability:** clear icons, buttons, dashboard, activity bar and straight forward instructions to facilitate use
4. **Portability:** due to technology use (JavaScript frameworks) cloud hosting, system can easily run on all android devices and windows devices
5. **Security:** security measures cited above such as encryption, password ensures a secured system

14.DESIGN RATIONALE

14.1 DESIGN CONSTRAINT

The **Microfinance Management System** must comply with financial regulations, ensure **data security** (encryption, authentication), and maintain **audit trails**. It should handle **high transaction loads**, support **multi-platform access**, and be **scalable**. The **UI must be user-friendly**, support multiple languages, and include accessibility features. Budget constraints should favor **cost-effective technologies** while ensuring **performance and security**. Deployment must allow **easy installation, updates, and maintenance** with minimal downtime.

14.2 TRADE – OFFS

A **Microfinance Management System** must balance **security and speed, cost and features**, and **scalability and simplicity**. It also faces trade-offs between **user-friendliness and advanced functions** and choosing **cloud (cheaper, but internet-dependent) or on-premises (costly, but more control)**.

15. MISCELANEOUS

The link to our linkedIn account https://www.linkedin.com/in/estelle-ngadang-6a768b347?utm_source=share&utm_campaign=share_via&utm_content=profile&utm_medium=ios_app

The link to our github account <https://github.com/Estelle-6>

