



ALBUKHARY INTERNATIONAL UNIVERSITY

SCHOOL OF COMPUTING AND INFORMATICS

SEMESTER 1 2023/2024

CCC2143

SOFTWARE ENGINEERING

GROUP PROJECT ASSIGNMENT (20%)

PROJECT TITLE	Smart Microfinance System for Micro and Small Entrepreneurs	
GROUP LEADER	Abdullahi Adewole Zakariyah	AIU22102360
GROUP MEMBERS	Areyan Muhemed Ali Hussein	AIU21102179
	Umar Muhammad	AIU22102266
	Nima Yoezer	AIU22102221
	Ugyen Tshering	AIU22102222

For Examiner's use only

ITEMS	MARKS
REPORT - 40	
PRESENTATION - 20	
TOTAL (60)	

TABLE OF CONTENTS

1. INTRODUCTION	Page 3
2. SYSTEM PROJECT OBJECTIVES.....	Page 3
a. Mission	
b. Vision	
3. SYSTEM REQUIREMENT ANALYSIS.....	Page 4-6
a. User Requirements	
b. System Requirements	
c. Functional Requirements	
d. Non-Functional REquirements	
4. SUITABLE SOFTWARE PROCESS MODEL.....	Page 7-8
5. SYSTEM MODEL AND ARCHITECTURE.....	Page 9-16
a. System Models	
b. System Architecture Design	
6. COMPONENT-BASED MODEL.....	Page 17-19
a. Loan Calculator Component	
b. User Authentication and Authorization Component	
c. Notification and Communication Component	
7. SYSTEM TESTING STRATEGY.....	Page 19-22
a. Development Testing	
b. Release Testing	
c. User Testing	
d. Test Driven Development	
8. ROLES AND RESPONSIBILITIES OF PROJECT MEMBERS.....	Page 23
9. CONCLUSION.....	Page 23
10. REFERENCES.....	Page 24-25

1. INTRODUCTION

A Smart Microfinance System is, like a toolbox designed to assist individuals with businesses or limited income in effectively managing their finances and accessing financial assistance when needed. It operates online. Provides services such as granting small loans aiding in accounting tracking fees managing customer information and ensuring compliance with regulations. These systems are specifically tailored for individuals who have limited funds or small scale enterprises. They prioritize offering loans, establishing savings accounts, providing insurance options and facilitating money transfers. By utilizing these services people can launch businesses. Enhance their lives if they have limited initial capital.

With Smart microfinance systems individuals can easily apply for loans through user digital platforms. The simplicity of the process makes it accessible to those who may have previously felt overwhelmed by the loan application process. Smart microfinance systems go beyond providing services. They act as a catalyst for change promoting economic growth, innovation and empowerment within communities. By investing in the aspirations of micro and small entrepreneurs these systems contribute to creating a future.

2. SYSTEM PROJECT OBJECTIVES

This micro financing system will outline the services needed by the micro entrepreneurs for their business venture. It is developed so that the micro entrepreneurs can grow in numbers and stimulate their interests more into micro financing.

Vision : “The vision of this system is to empower the micro entrepreneurs to take greater control on their business ventures and significantly improve their skills in financial management.”

Mission: “Our mission is to provide micro entrepreneurs a system where they can manage their financial resources effectively at a low cost with greater functionality. ”

3. SYSTEM REQUIREMENT ANALYSIS

a. User Requirement

The 'Smart Microfinance System' is designed to empower micro and small entrepreneurs by providing them with a user platform that helps them manage their finances efficiently. It simplifies the loan application process, allows users to track transactions transparently and provides time insights. This system also facilitates communication between users and financial institutions giving them access to microfinance services and improving the overall sustainability of their businesses.

b. System Requirements

Register as a Client: To get started the system will have a registration feature that allows micro and small entrepreneurs to create accounts effortlessly. During this process users will provide business information that will be securely stored to establish their profiles.

Apply for a Loan: Within the system there will be a module for loan applications. Once registered, clients can use this module to submit loan requests by providing details such as the purpose of the loan and the desired amount.

Make Payment: The system ensures convenience and security by offering payment methods through a payment gateway. This empowers clients to make loan repayments easily without any worries.

Manage Account: Micro and small entrepreneurs can take advantage of robust account management features within the system. They can use these tools to update and maintain their personal as business information associated with their accounts. Additionally administrators will have access to a module within the system, for efficient management. This module gives administrators the ability to handle user accounts, monitor system activities and perform tasks, for optimal system operation.

Review Loan Application: There will be a module within the system specifically designed for loan professionals. Within this module loan professionals can. Thoroughly review loan applications submitted by clients.

Credit Check: In order to evaluate the creditworthiness of loan applicants the system will integrate with credit check services. This integration ensures an evaluation during the loan application process.

Evaluate Collateral: As part of the review process for loan applications the system will provide functionality that allows loan professionals to assess and evaluate the collateral offered by applicants.

Approve Loan: Building upon the evaluation of credit checks and collateral the system will facilitate the approval of loans. This functionality empowers loan professionals to make decisions based on assessments.

Generate Financial Statement: The system will include a financial reporting module. This module generates statements that provide clients with a comprehensive overview of their financial transactions and status.

Configure System: To ensure adaptability, to changing business requirements there will be configuration options provided by the system. Administrators can utilize these options to customize system settings according to needs.

Keep Record and Notify the Client for Information: The system will maintain a record of all transactions and interactions. Furthermore it should include notification functionalities to ensure that customers are kept updated regarding the progress of their loan and other important details, within a timeframe.

c. Functional Requirements

- a. *User Registration:* Allow individuals to sign up as clients by providing the information and make sure to validate and verify the accuracy of the details they provide.
- b. *Loan Application:* Simplify the process for clients when applying for a loan. Gather details such as loan amount, purpose and duration while also implementing checks to ensure all required information is provided.
- c. *Payment Management:* Enable clients to make payments using methods. Integrate a payment gateway that's user friendly and provides real time confirmations.
- d. *Account Management:* Provide clients with a dashboard where they can manage their accounts, view transactions and update their information.
- e. *Loan Application Review:* Create a workflow for staff members to review and process loan applications efficiently. This workflow should include features that enable communication with clients.

- f. *Credit Check*: Securely integrate mechanisms for credit checks to assess the creditworthiness of loan applicants.
- g. *Collateral Evaluation*: Support staff members in evaluating and documenting the value and viability of collateral submitted by clients.
- h. *Loan Approval*: Design a workflow for approving loan applications based on criteria ensuring that the process is secure and can be audited if needed.
- i. *Financial Statement Generation*: Develop a reporting module that generates statements for clients including summaries of transactions and payment histories.
- j. *System Configuration*: Provide administrators with a user interface that allows them to configure system settings easily ensuring adaptability as, per their requirements.
- k. *Record Keeping*: Set up a system for storing client information, including encryption and strict access controls.
- l. *Client Notification*: Develop an automated system for notifying clients about their account status, loan approvals and payment reminders. This system should also provide clients with the option to customize their notification preferences.

d. Non-Functional Requirements

- A. Security; Implement security measures to safeguard user data and financial transactions ensuring confidentiality and integrity.
- B. Efficiency; Streamline all processes from registration to loan approval to ensure response times and efficient execution.
- C. Accessibility; Guarantee that users of all abilities can access and utilize all features of the system fostering inclusivity.
- D. Accuracy; Ensure the precision of credit checks, collateral evaluation, financial statement generation and other crucial procedures.
- E. Flexibility; Create a system that permits administrators to adjust settings according to evolving requirements.

4. SUITABLE SOFTWARE PROCESS MODEL

Agile method is a project management approach where it prioritizes cross-functional collaboration and continuous improvement. It is known as rapid development where the projects are managed and continuously improved. It also features a nature of adaptability where the changes are quickly made and adapted.

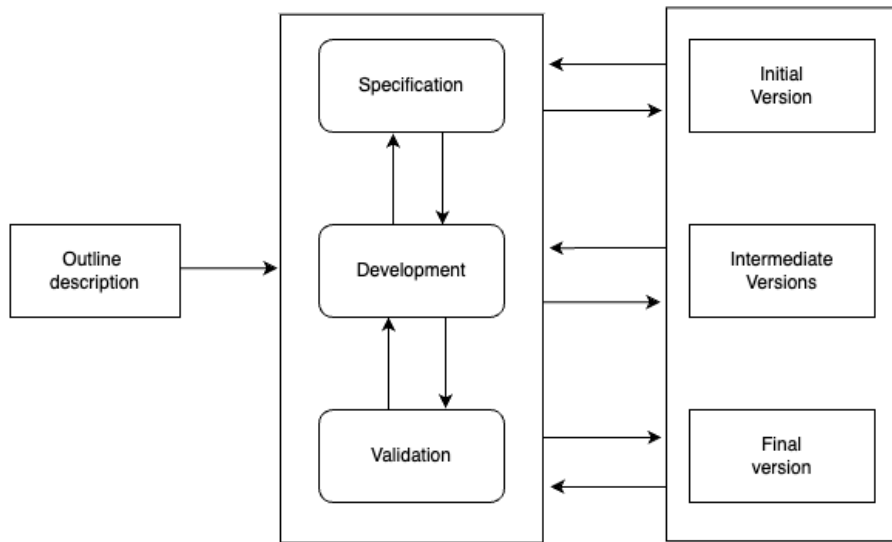


Figure 1.0: Incremental Development Process

Incremental development is a software development approach where the software system is developed and evolved until the required system is developed through feedback from users and others. Specification, development and validation are interleaved instead of separate. The specification is done at initial version, development at intermediate and validation comes as a final version. These three activities are concurrent meaning simultaneous. In an agile approach, the early increments are identified and the later increments are developed based on the progress and user's priorities.

Incremental method has a lot of advantages and suitable to our project with the following reasons:

1. Adaptability to evolving requirements: The micro finance is inevitably exposed to changes and needs of development every now and then. So the flexible nature of incremental methods can adapt to changes and requirements needed for the system.

2. Feedback: Through this method, the users can provide feedback on system features so that it can lead to the development needed for its effective performance.

3. Delivery of essential features: The core functionality of the system is placed early during the development by focusing on essential features first therefore providing subsequent enhancements. So it can contribute to the timely delivery of financial services to entrepreneurs.

4. Risk mitigation: The incremental agile methods can help in alleviating the risks since the development process can be broken down into smaller and manageable increments. Through this process, the risks and development needed to be done can be identified and addressed early in project development.

5. Collaboration: Agile method highlights the collaboration and communication between the users and development team. Through this, they can have a better understanding of the projects, align their goals and deliver good products.

With all these reasons and features available, we choose the agile incremental methodology so we can deliver good products and have effective interactive developments.

5. SYSTEM MODEL AND ARCHITECTURE

a. System Models

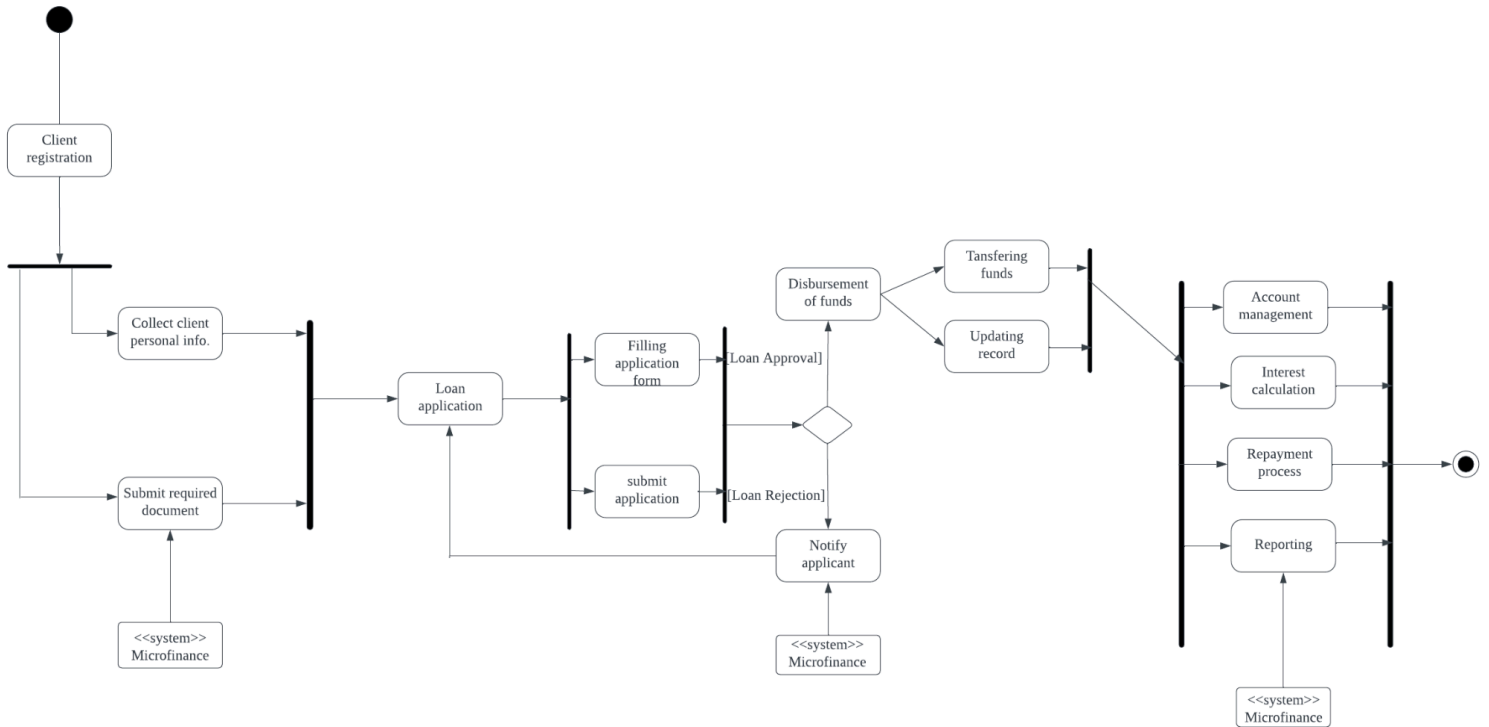


Figure 1.1: Activity Diagram



Figure 1.2: Use case Diagram.

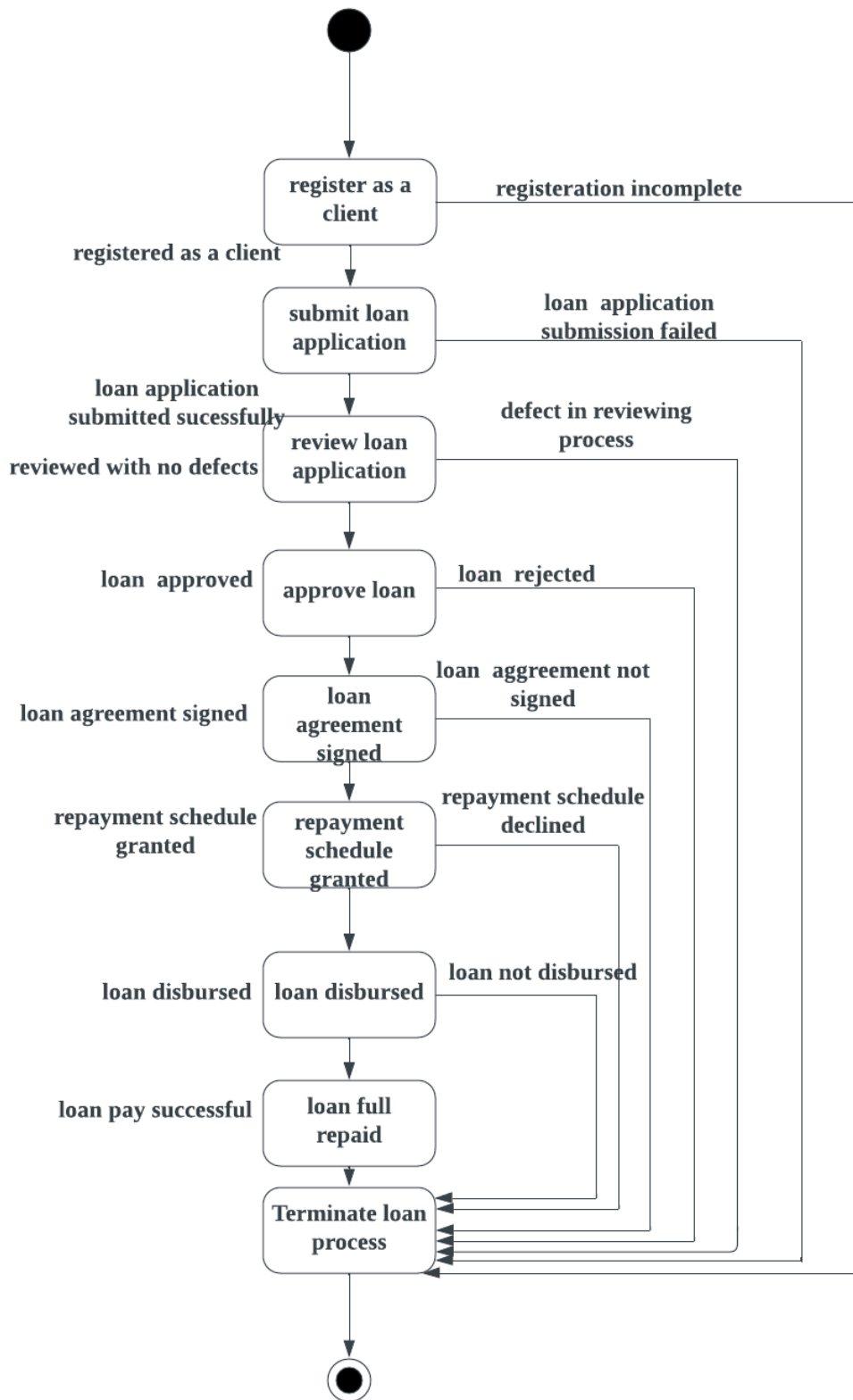


Figure 1.3: *State Diagram.*

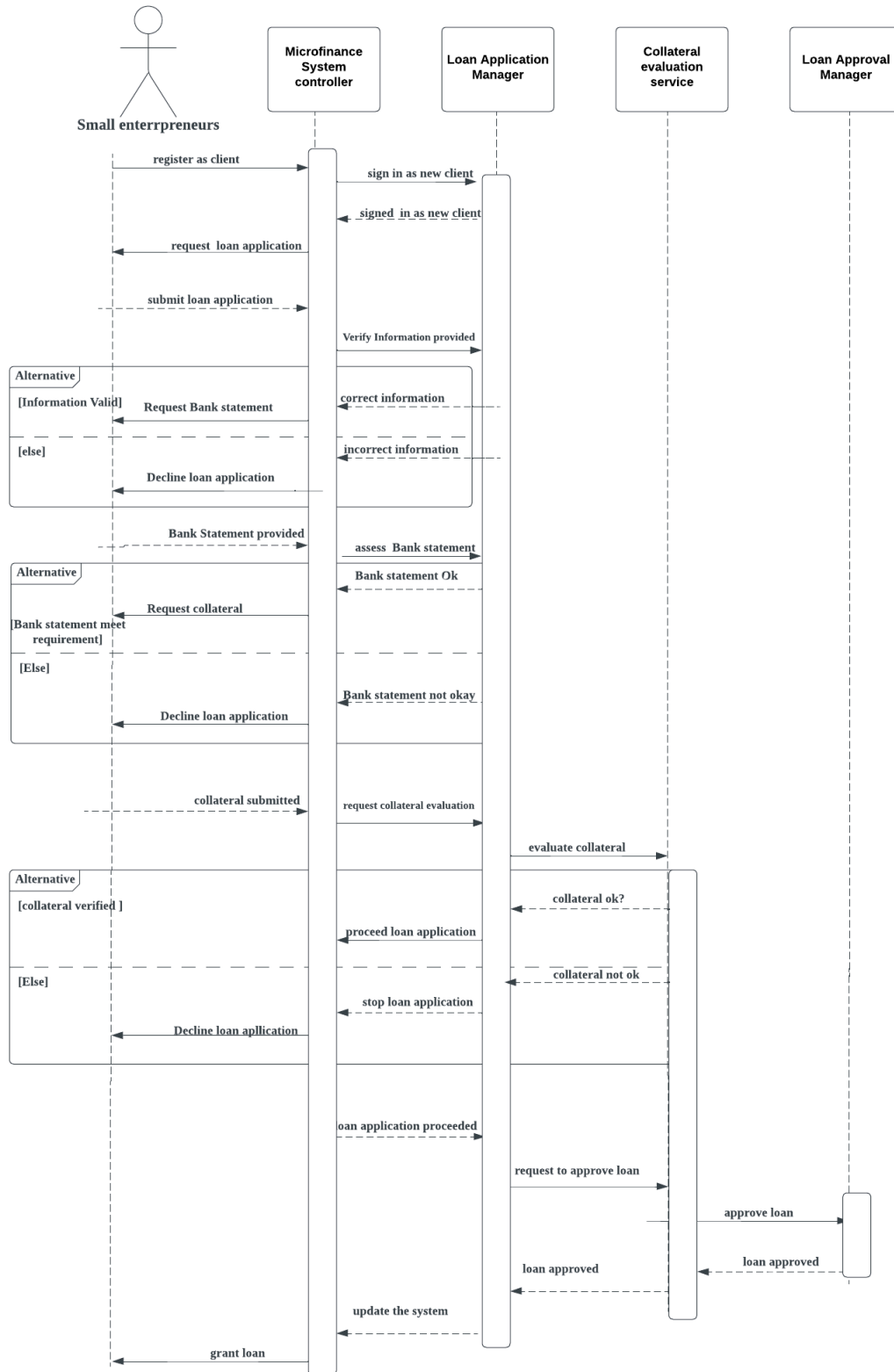


Figure 1.4: Sequence Diagram.

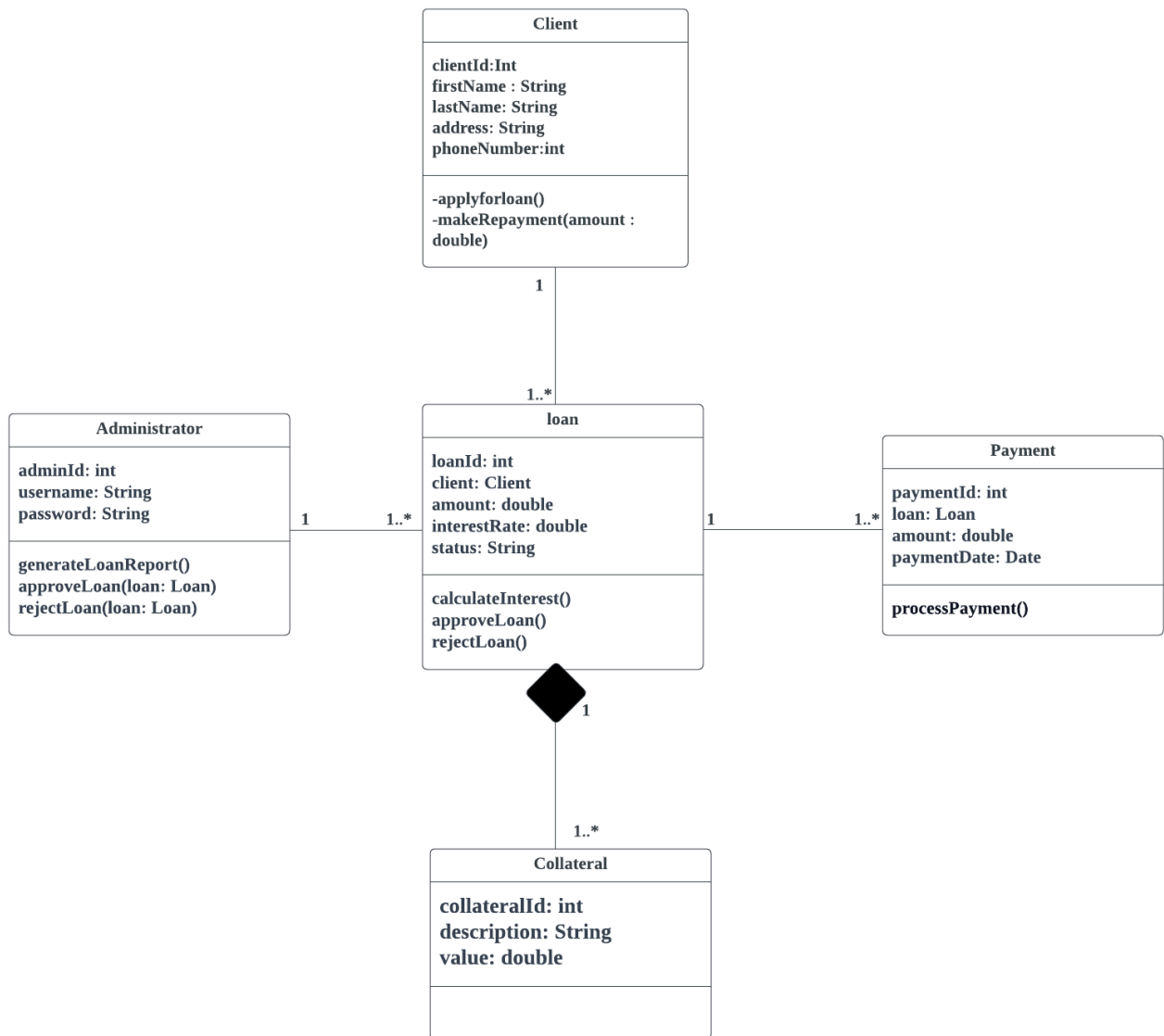


Figure 1.5: *Class Diagram.*

b. System Architecture Design

User Interface



User Interface Management



Configuration Services



Application Services



Integrated Services



Shared Infrastructure Services



Figure 1.6: *Layered Architecture.*

Name	Layered Architecture
Description	The Layered Architecture pattern is another way of achieving separation and independence. Here, the system functionality is organized into separate layers, and each layer only relies on the facilities and services offered by the layer immediately beneath it. This layered approach supports the incremental development of systems. The lowest level layers represent core services that are likely to be used throughout the system.
When used	It is used when building new facilities on top of existing systems. For instance, when the development is spread across several teams with each team responsible for a layer of functionality and when there is a requirement for multilevel security.
Advantages	It allows replacement of entire layers as long as the interface is maintained. Moreover, the redundant facilities (e.g., authentication) can be provided in each layer to increase the dependability of the system.
Disadvantages	In practical terms, providing a clean separation between layers is often difficult, and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it. Also, performance can be greatly reduced since the top layer cannot directly interact with the lowest layer.

User Interface: The first layer provides user interface facilities, which involves presenting information to the users and collecting user inputs for activities related to loan application, transactions and many more.

- a. Web or app interface allows users to interact with the system

User Interface Management: This layer manages user interactions and experience, as well as provides authentication and authorization.

- a. Session Management: It manages the users activity in the application

- b. Interface Creation: It creates an appropriate interface for the users
- c. Interface Delivery: Helps to deliver the best interface interaction and experience
- d. Log in: Options given to the user for authentication and authorization

Configuration Services: This layer includes several components that incorporate knowledge of the installed applications and provide configuration functionality to users.

- a. User Interface Configuration: Supports user interface functionality
- b. Application Configuration: Provides overall functionality of the application
- c. Security Configuration: Used to enhance the overall security of the application
- d. Configuration API: Provides access to configuration data

Application Services: This layer includes the components concerned with the application functionality and utility components used by other application components.

- a. Customer Management: Manages customer related operations
- b. Loan Processing: Handles loan applications and approvals
- c. Risk Analysis: Evaluates credit risk
- d. Transaction Handling: Manages at-all financial transactions
- e. Account Management: Helps to manage users accounts

Integrated Services: These include complementary services integrated into the system to improve its overall interactions and functionality.

- a. Service Integration: Manages service communications
- b. User Analytics: Helps to produce user analytics
- c. Authentication and Authorization: Validates user's identity and manage access rights

Shared Infrastructure Interface: This the lowest layer which includes system support software, typically, database and operating system support.

- a. Logging and Monitoring: Trackers system activities
- b. Application Interfacing: Application's interactions
- c. Shared Database: Centralized data storage

6. COMPONENT-BASED MODEL

a. Loan Calculator Component

In our smart microfinance system, we have got a cool Loan Estimation Tool. Just plug in your loan amount, interest rate and how long you want to repay and voila it spits out your monthly repayment amount, total repayment, and other handy details, making it easy peasy for micro and small enterprises to plan their finances.

In our microfinance System for Small Businesses, The loan calculator is independently providing precise details about your loan. It follows rules that everything trusts for financial calculation, ensuring consistency and reliability. In the microfinance world, the loan calculator is your go-to ally making loan management adaptable, easy to integrate, well documented, independent, and in line with established financial standards. Small businesses can rely on it to navigate their financial journeys with confidence and ease.

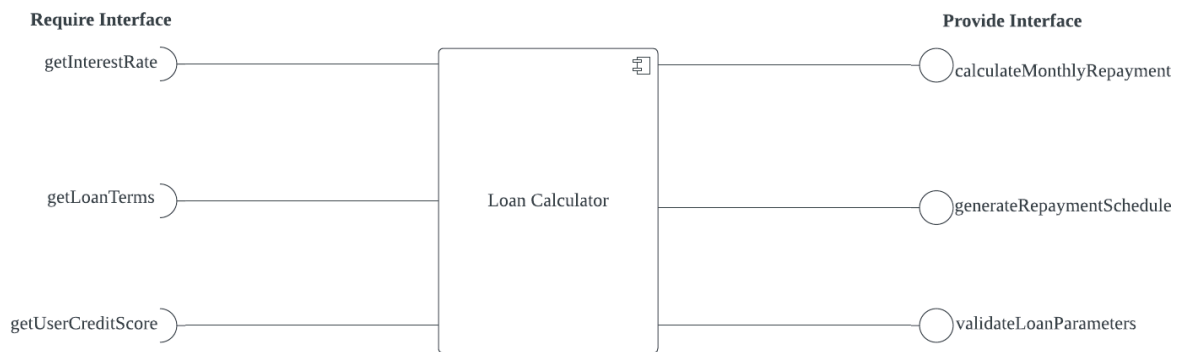


Figure 1.7: *Component Based Diagram for Loan Calculator.*

b. User Authentication and Authorization Component

- a. User Registration: When you sign up, share your name, email and create a password. We will make sure your email and password are strong and keep your information safe
- b. Login to log in type in your email and password we will check if they match what you gave us when signing up if they match you are good to go
- c. Password Recovery: Tell us if you forgot your password, and we will send you a special email. Click the link in the email to set a new password your password is updated
- d. Access Control: we have different roles, admins can do everything, while regular users can do something we decide who can do what in the system.

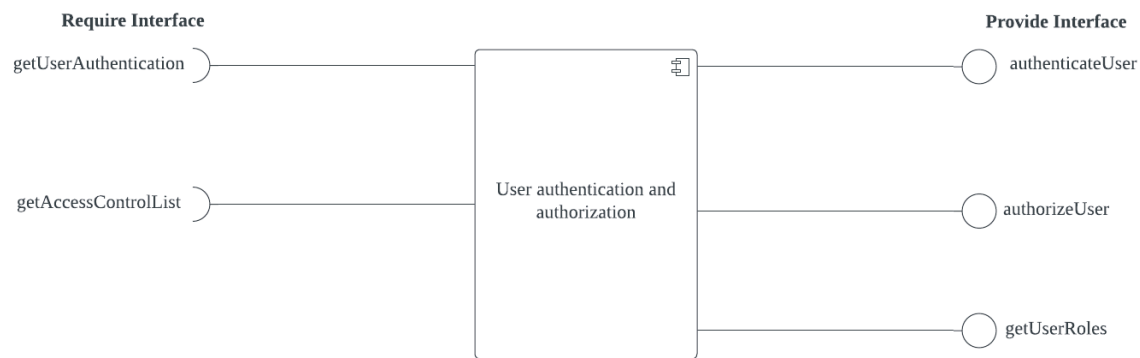


Figure 1.8: *Component Based Diagram for User Authentication and authorization*

c. Notification and Communication Component

Our smart microfinance system boasts a robust Communication Center. The component connects with users through email, SMS, and in-app notifications. It is designed to send vital updates like loan approvals, repayment reminders, and other essential information. The Communication Center ensures clients stay engaged and well-informed about their financial activities. It's the

go-to feature for fostering effective communication, enhancing user engagement and providing smooth experience for individuals managing their finances within our microfinance system

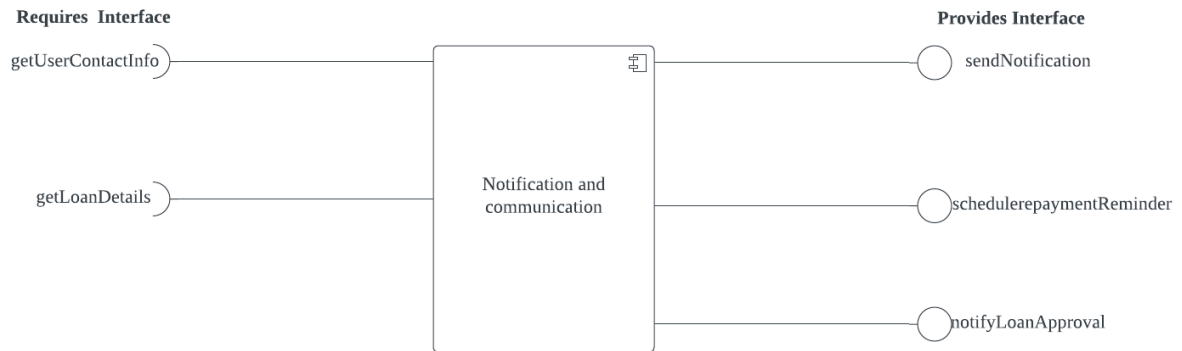


Figure 1.9: Component Based Diagram for Notification and communication.

7. SYSTEM TESTING STRATEGY

a. Development Testing:

Loan Approval
SubmitLoanApplication()
VerifyInfo()
RequestBankStatement()
SubmitBankStatement()
AccessBankStatement()
SubmitCollateral()
EvaluateCollateral()
ApproveLoan()

Unit testing is the process of testing program components, such as methods or object classes. Unit includes Individual functions or methods within an object.

When testing object classes, test design is crucial to provide coverage of all of the features of the object. It should test all operations associated with the object; set and check the value of all attributes associated with the object; and put the object into all possible states.

Figure 2.0: Unit Testing Diagram

Use Case testing

Because of its focus on interactions, use case-based testing is an effective approach to system testing. Several components or objects normally implement each use case in the system. Testing the use case forces these interactions to occur. If you have developed a sequence diagram to

model the use case implementation, you can see the objects or components that are involved in the interaction. We can use the sequence diagram (Figure 1.4: *Sequence Diagram*) to identify operations that will be tested and to help design the test cases to execute the tests. Therefore issuing a request for a report will result in the execution of the following thread of methods:

Microfinance System Controller:submit loan application → Loan Application Manager:proceed → Collateral Evaluation Service:request to approve load→ Loan Approval Manager:approve loan

b. Release Testing:

Required based testing

Requirement: if a client is known not to be aware of how the loan processing is going in the system.

1. Notification should be sent automatically to clients for all the transactions they made in the system. Including deposit, withdrawal, loan disbursement, and repayment.
2. Summarizing the financial activities carried out by the client over a period of time.
3. Provide timely updates to clients regarding the loan application status, notify the client for loan approval, disbursement, and changes in repayment schedules.
4. Send automated reminders to clients in advance of upcoming loan repayment.
5. Allow clients to access their notification history through the system.

Scenario testing

I. Applying for loan in Microfinance bank system

1. Account logging or Registration as client

System features tested:

- a) User authentication : scenarios that may occur include: wrong password or username.
- b) Registration : scenarios that may occur include: wrong gmail address, wrong phone number.

2. Loan application

System features tested:

- a). Attachment of necessary docs: scenarios that may occur include: attaching wrong docs, docs not attached.
- b). Submit application : scenarios that may occur include: application not submitted

3. Loan Disbursement

System features tested:

- a). Transferring fund: scenario that may occur include: fund not transferred, transfer wrong amount of money

4. Repayment:

System features tested :

- a). Repayment process: scenarios that may occur include : repayment not made on time, transaction error in repayment process.

c. User Testing:

User testing is a stage in the testing process in which users or customers provide input and advice on system testing. The reason for this is that influences from the user's working environment have a major effect on the reliability, performance, usability and robustness of a system

- Alpha testing: Bring in users with the development team to test the software at the developer's site.
- Beta testing: A release of the software is made available to users to allow them to experiment and to raise problems that they discover with the system developers.
- Acceptance testing: Customers test a system to decide whether or not it is ready to be accepted from the system developers and deployed in the customer environment

d. Test Driven Development

Test-driven development (TDD) is an approach to program development in which testing and code development are interleaved. As we develop the code incrementally for our microfinance system, we will also perform a set of tests for that increment. We will not start working on the next increment until the code that we have developed passes all of its tests.

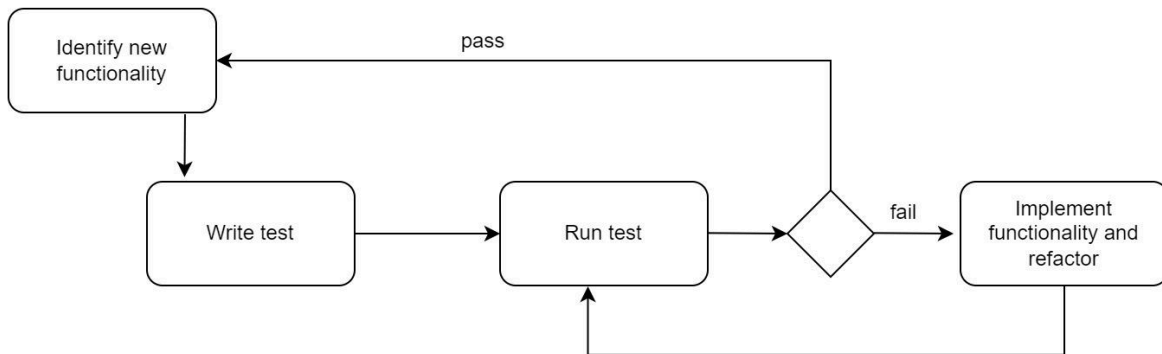


Figure 2.1: Test Driven Strategy Method

The fundamental TDD process are as follows:

1. We start by identifying the increment of functionality that is required.
2. We write a test for this functionality and implement it as an automated test.
3. We then run the test. If it fails, we then implement the functionality and re-run the test. If it passes the test, we move on to implementing the next chunk of functionality

8. ROLES AND RESPONSIBILITIES OF PROJECT MEMBERS

Name	Student ID	Roles
Abdullahi Adewole Zakariyah	AIU22102360	System Models Software Testing
Areyan Muhemed Ali Hussein	AIU21102179	Component Based Development
Umar Muhammad	AIU22102266	Requirement Analysis and conclusion
Nima Yoezer	AIU22102221	Software Process Model
Ugyen Tshering	AIU22102222	System Architecture Design and Software Testing

9. CONCLUSION

In summary the Smart Microfinance System is an achievement in empowering micro and small entrepreneurs. During the development process the main focus was on meeting user needs resulting in a platform that simplifies financial management tasks. Despite challenges like ensuring data security and optimizing system efficiency the project successfully overcome these obstacles. Moving forward there is room for improvement in areas such as enhancing user experience, strengthening security measures and exploring technologies. The Smart Microfinance System not only showcases innovation but also demonstrates a commitment to fostering economic growth and empowering communities.

10. REFERENCES

- Catania, J. T. (2006). Requirements analysis: A review. *Advances in Systems, Computing Sciences and Software Engineering: Proceedings of SCSS05*, 411-418.
- Maiden, N. (2008). User requirements and system requirements. *IEEE Software*, 25(2), 90-91.
- Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). *Non-functional requirements in software engineering* (Vol. 5). Springer Science & Business Media.
- Authen, D. (2021) *What is Agile Methodology in project management?*, *Versatile & Robust Project Management Software*. Available at: <https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/> (Accessed: 28 January 2024).
- Collegenote (2022) *Concept of user and system requirements, Concept of User and System Requirements*. Available at: <https://www.collegenote.net/curriculum/software-engineering-csit/54/309/> (Accessed: 28 January 2024).
- Coronel-Pangol, K. et al. (2023) *Microfinance, an alternative for financing entrepreneurship: Implications and trends-bibliometric analysis*, *MDPI*. Available at: <https://www.mdpi.com/2227-7072/11/3/83> (Accessed: 28 January 2024).
- Kagan, J. (2023) *Microfinance definition: Benefits, history, and how it works*, *Investopedia*. Available at: <https://www.investopedia.com/terms/m/microfinance.asp> (Accessed: 28 January 2024).
- Pajatha, R. (2023) *Software engineering: Classification of software requirements*, *GeeksforGeeks*. Available at: <https://www.geeksforgeeks.org/software-engineering-classification-of-software-requirements/> (Accessed: 28 January 2024).
- Team, A. (2023) *Functional and non-functional requirements: Specification an*, *AltexSoft*. Available at:

<https://www.altexsoft.com/blog/functional-and-non-functional-requirements-specification-and-types/> (Accessed: 28 January 2024).

Team, O. (2023) *Software architecture patterns*, *O'Reilly Online Learning*. Available at: <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html> (Accessed: 28 January 2024).