



National University of Computer and Emerging Sciences



Credit Risk Liaison (CRL)

FYP Team

Muaz Nadeem.....19L-0928
Muhammad Abdullah.....19L-0924
Ali Idrees.....19L-1141

Supervised by

Saifullah Tanvir

FAST School of Computing

National University of Computer and Emerging Sciences

Lahore, Pakistan

March 2023

Anti-Plagiarism Declaration

This is to declare that the above publication produced under the:

Title: Credit Risk Liaison (CRL)

is the sole contribution of the author(s) and no part hereof has been reproduced on **as it is** basis (cut and paste) which can be considered as **Plagiarism**. All referenced parts have been used to argue the idea and have been cited properly. I/We will be responsible and liable for any consequence if violation of this declaration is determined.

Date: 10/03/23

Student 1

Name: _____

Signature: _____

Student 2

Name: _____

Signature: _____

Student 3

Name: _____

Signature: _____

Student 4

Name: _____

Signature: _____

Authors' Declaration

This states Authors' declaration that the work presented in the report is their own, and has not been submitted/presented previously to any other institution or organization.

Abstract

Our project aims to develop a web-based platform using MERN Stack technology for the wholesale and retail industry. The platform provides a marketplace for retailers to connect with local wholesalers and allows them to place orders directly. The platform also features a credit scoring system to evaluate retailers based on their past order history and creditworthiness, allowing wholesalers to make informed decisions when accepting or rejecting orders. The system's non-functional requirements include performance, scalability, and security, ensuring the platform can handle many users and transactions securely. The project's successful completion will provide an innovative solution to streamline the wholesale and retail industry, benefiting both retailers and wholesalers by making the ordering process efficient and cost-effective.

Executive Summary

Our project aims to provide a blockchain-based platform for small retailers and wholesalers to connect and conduct transactions in a secure and efficient manner. The platform will leverage the MERN stack to provide a user-friendly web-based interface for the users. The platform will include features such as inventory management, order tracking, and a credit score system to incentivize good behavior.

The primary beneficiaries of our platform will be small retailers and wholesalers who currently face challenges in accessing credit and conducting transactions in a secure manner. By leveraging blockchain technology, our platform will provide a transparent and secure system for conducting transactions, which will improve trust between parties and make it easier to access credit.

We have identified several key risks, such as technical challenges and regulatory issues, but we have put in place measures to mitigate these risks. Our project team consists of experienced developers who will use an agile development methodology to ensure that the project is delivered on time and within budget.

Overall, our project represents a significant opportunity to disrupt the retail and wholesale industry and provide a much-needed solution for small retailers and wholesalers. We believe that our platform has the potential to revolutionize the way small businesses conduct transactions, and we are excited to bring this project to fruition.

Table of Contents

Table of Contents	i
List of Tables	iv
List of Figures	v
Chapter 1: Introduction	1
This section provides a basic go through on what, for whom this document is.	1
1.1 Purpose of this Document.....	1
1.2 Intended Audience	1
1.3 Definitions, Acronyms, and Abbreviations	1
Chapter 2: Project Vision.....	2
2.1 Problem Domain Overview	2
2.2 Problem Statement	2
2.3 Problem Elaboration	2
2.4 Goals and Objectives	2
2.5 Project Scope	3
2.6 Sustainable Development Goal (SDG)	3
2.7 Constraints	3
2.8 Business Opportunity.....	4
2.9 Stakeholders Description/ User Characteristics.....	4
2.9.1 Stakeholders Summary	4
2.9.2 Key High-Level Goals and Problems of Stakeholders	5
Chapter 3: Related Work	6
3.1 Explainable AI in Credit Risk Management.....	6
3.1.1 Summary of the study	6
3.1.2 Relation to our work	6
3.2 ANN Approach for Credit Risk Management	6
3.2.1 Summary of the study	6
3.2.2 Relation to our work	6
3.3 Machine learning and AI for Risk Management.....	7
3.3.1 Summary of the study	7
3.3.2 Relation to our work	7
3.4 Provenir	7
3.4.1 Summary of the study	7
3.4.2 Relation to our work	7
3.5 Zest AI	7
3.5.1 Summary of the study	7
3.5.2 Relation to our work	7
3.6 Conclusion	8
Chapter 4: Software Requirement Specifications	9
4.1 List of Features	9
4.2 Functional Requirements	9
4.3 Quality Attributes.....	9
4.4 Non-Functional Requirements	10
4.5 Assumptions.....	10
4.6 Hardware and Software Requirements	11
4.6.1 Software Requirements	11
4.7 Use Cases	11
4.7.1 Login	11
4.7.2 Sign Up	12
4.7.3 Log out	12

4.7.4 View Client Options	13
4.7.5 Add Local Retailer/Whole seller	13
4.7.6 Generate Credit Score	14
4.7.7 Agree upon T&Cs.	14
4.7.8 Verify consignment status.....	15
4.7.9 View Analytics.....	16
4.7.10 View Transactions	16
4.7.11 Compare Retailers.....	17
4.7.12 Get Best Suggestions	17
4.8 Graphical User Interface	18
4.8.1 Main Page	18
4.8.2 Log In.....	19
4.8.3 Sign Up	19
4.8.4 Analytics	20
4.8.5 Terms & Conditions.....	20
4.8.6 Getting Credit Scores	21
4.8.7 Client Options	21
4.9 Database Design.....	22
4.9.1 ER Diagram	22
4.9.2 Data Dictionary	22
4.10 Risk Analysis	23
Chapter 5: High-Level and Low-Level Design	24
5.1 System Overview	24
5.1.1 Web-Based Application	24
5.1.2 AI Modeling.....	24
5.1.3 Blockchain-based Database	25
5.2 Design Considerations	25
5.2.1 Assumptions and Dependencies	26
5.2.2 General Constraints.....	26
5.2.3 Goals and Guidelines	27
5.2.4 Development Methods	28
5.3 System Architecture.....	29
5.3.1 Subsystem Architecture	30
5.4 Architectural Strategies.....	31
5.4.1 Front-end Framework	31
5.4.2 Backend Stack.....	32
5.4.3 Microservices Architecture	32
5.5 Class Diagram	33
5.6 Sequence Diagram	34
5.6.1 Authentication.....	34
5.6.2 Searching and Storing Vendor Details.....	35
5.6.3 Storing Lender Document in Database	36
5.6.4 Retrieving Lender Document from Database	37
5.6.5 Generating Credit Score and Establishing Connection.....	38
5.6.6 Transaction Verification	39
5.6.7 Chatbot.....	40
5.7 Policies and Tactics.....	40
5.7.1 Coding Guidelines and Conventions	40
5.7.2 Testing Framework	40
5.7.3 Git Branching Strategy	41

5.7.4 Dependency Management.....	41
5.7.5 Error Handling	41
Chapter 6: Implementation And Test Cases	42
6.1 Implementation	42
6.1.1 Prototype Description	42
6.1.2 Sign Up Page.....	42
6.1.3 Login Page	42
6.1.4 Retailer – Whole-Seller Interface	42
Chapter 7: Conclusion And Future Work	43
References.....	44

List of Tables

Table 1: Data Dictionary.....	22
-------------------------------	----

List of Figures

Figure 1: Sustainable Development Goals.....	3
Figure 2: Main Page	18
Figure 3: Login Page.....	19
Figure 4: Sign Up Page	19
Figure 5: Analytics Page	20
Figure 6: Terms and Condition Page	20
Figure 7: Credit Scores Page.....	21
Figure 8: Client Options.....	21
Figure 9: ER Diagram.....	22
Figure 10: System Architecture	29
Figure 11:Admin Module	30
Figure 12: Retailer Component.....	31
Figure 13: Wholesaler Component	31
Figure 14: Class Diagram	33
Figure 15: Authentication Sequence Diagrams	34
Figure 16: Storing Vendor Details Sequence Diagram.....	35
Figure 17: Storing Lender Documents Sequence Diagram	36
Figure 18: Retrieving Lender Document Sequence Diagram.....	37
Figure 19: Generating Credit Card Score Sequence Diagram	38
Figure 20: Transaction Verification Sequence Diagram	39
Figure 21: Chatbot Sequence Diagram	40

Chapter 1: Introduction

This section provides a basic go through on what, for whom this document is.

1.1 Purpose of this Document

The purpose of this document is to present the design and development of a web-based platform for connecting local retailers and whole-sellers. This platform aims to streamline the buying and selling process, increase efficiency and transparency, and provide a more convenient solution for retailers and whole-sellers for on-credit trading of goods. This document will provide a comprehensive overview of the platform, including the project scope, requirements, design, and implementation details. The document is intended for stakeholders, developers, and project managers involved in the design and development of the platform, as well as potential users who are interested in understanding how the platform works.

1.2 Intended Audience

The intended audience for the document includes our university evaluators who will be grading our project, as well as any stakeholders who may be interested in learning about our project, such as potential investors or industry professionals. It is important that the document is clear and concise, and that technical terms are explained in a way that is accessible to a broad audience.

1.3 Definitions, Acronyms, and Abbreviations

List all important definitions, the acronyms and abbreviations used in this document.

SDG: Sustainable Development Goal

MERN: MongoDB, ExpressJS, ReactJS, NodeJS

Chapter 2: Project Vision

This section provides the complete vision we have for Credit Risk Management. It includes the problem that exists, the solution our project provides, its goals and objectives, the project scope, and its constraints.

2.1 Problem Domain Overview

Our problem domain is the lending and borrowing of goods and services on credit. In traditional lending systems, there is a lack of trust between the parties involved, leading to delayed payments, or even defaulting on payments. Additionally, there is a risk of fraud and manipulation in the lending process. Our system aims to address these issues by using blockchain technology to create a transparent and secure lending platform.

2.2 Problem Statement

The traditional process of lending and borrowing between parties can be complicated and risky, particularly when conducted on credit. There is a lack of transparency and trust between parties, resulting in potential disputes and loss of financial resources.

2.3 Problem Elaboration

The problem we aim to solve is the inefficiencies and risks present in the traditional wholesale market, specifically in the payment and delivery processes. Our solution is to leverage blockchain technology to create a platform that enables secure and efficient transactions between wholesalers and buyers.

One of the main sub-problems we will tackle is the lack of trust between wholesalers and buyers due to the high risk of fraud and default in traditional payment methods. We will address this by implementing smart contracts that use AI models to determine the likelihood of default, and trigger bank transfers to release funds to the appropriate parties based on the terms of the contract.

Another sub-problem we will address is the lack of transparency and traceability in traditional transactions. By recording all transactional data on the blockchain, we will provide a clear and immutable record of all transactions, allowing for easy tracking and auditing.

Furthermore, we will focus on the sub-problem of inefficient delivery processes. By implementing a secure and trackable delivery process on our platform, we will reduce the risk of lost or delayed deliveries and ensure that buyers receive their goods in a timely and efficient manner.

Overall, our platform aims to streamline and secure the wholesale transaction process, improving trust, efficiency, and transparency in the industry.

2.4 Goals and Objectives

The primary goal of this project is to develop a blockchain-based platform that facilitates secure and efficient wholesale transactions between buyers and sellers. This platform will leverage smart contracts and AI models to enable automated payment release and transaction verification, reducing the need for intermediaries and improving trust between parties. Our objectives include designing a user-friendly interface, integrating with existing bank transfer systems to trigger smart contract events, and recording transactional data on the blockchain to ensure transparency and immutability. Additionally, we aim to explore the potential of machine learning models to provide insights into market trends and optimize transaction processes.

2.5 Project Scope

Our project aims to develop a web-based application using React as the front-end technology that will enable wholesalers and retailers to conduct their business transactions seamlessly without the need for intermediaries like payment gateways. The platform will utilize smart contracts and blockchain technology to ensure secure and transparent transactions between parties. The scope of the project includes the development of the platform's front-end and back-end, implementing AI models to facilitate payment verifications, and integrating blockchain technology to record transactional data. The application will support bank transfers, which will trigger the smart contracts to release payments to the appropriate party based on the pre-agreed conditions. The platform will also enable parties to track their transactions and view their transaction history on the blockchain ledger.

2.6 Sustainable Development Goal (SDG)

Decent Work and Economic Growth. Our project aims to create a platform that connects retailers and wholesalers, promoting economic growth by facilitating transactions and promoting fair trade. By providing a platform that simplifies the purchasing process, our project also promotes the creation of decent work by enabling retailers and wholesalers to expand their businesses, potentially leading to job creation.



Figure 1: Sustainable Development Goals
This figure represents all the SDG's that can be target of a FYP

2.7 Constraints

Some potential constraints of the project could include:

- Regulatory compliance: As the platform will be handling financial transactions, there may be legal or regulatory requirements that need to be followed in order to ensure compliance.
- User adoption: The success of the platform will depend on whether users are willing to adapt it and use it for their business needs. It all depends on how we pitch it into the market.

2.8 Business Opportunity

Our platform presents a unique business opportunity as it aims to solve the long-standing problems in the wholesale market industry by providing a secure, transparent, and efficient platform for conducting business transactions. With the increasing demand for global trade and the digitization of transactions, our platform will provide a competitive advantage to wholesale businesses by reducing transaction costs, improving efficiency, and increasing trust between parties.

By leveraging AI models and blockchain technology, our platform ensures accurate and secure recording of transactional data and payment history. This not only reduces the risk of fraud and errors but also enables parties to make informed decisions based on the historical data. Moreover, our platform offers a seamless payment process where lenders can receive payments directly through their bank accounts triggered by smart contracts, eliminating the need for traditional escrow services.

In addition, our platform offers flexibility and customization options to meet the varying needs of different wholesale businesses. This allows our platform to cater to a broad range of wholesale businesses and expand its market reach. With our user-friendly and accessible web-based application with React as its front-end, we aim to provide a seamless experience to users. Therefore, our platform presents a unique business opportunity for wholesale businesses to improve their efficiency, reduce transaction costs, and increase trust in their business transactions.

2.9 Stakeholders Description/ User Characteristics

The users of the system will be divided into two main groups: lenders and borrowers. Lenders are individuals or companies who provide funding to borrowers in exchange for a return on their investment. Borrowers are individuals or companies who require funding for their business operations.

The lenders will be able to browse through different loan requests submitted by the borrowers, review the terms and conditions of each loan request and determine whether they want to invest or not. On the other hand, borrowers will be able to create a loan request, set the terms and conditions, and wait for lenders to invest in their loan.

Other users of the system will include administrators who will manage the overall operations of the platform, including handling user accounts, ensuring regulatory compliance, and maintaining the platform's security.

2.9.1 Stakeholders Summary

Stakeholders of the proposed system include borrowers, lenders, and the platform operator. The borrowers are individuals or businesses who require funds to purchase goods from suppliers, while the lenders are investors who are willing to finance these purchases. The platform operator is responsible for providing the web-based platform that connects borrowers and lenders and ensures the smooth operation of the system.

Borrowers will create profiles on the platform and upload information about their business, including their creditworthiness and the types of goods they intend to purchase. Lenders will also create profiles and specify the types of borrowers and goods they are willing to finance. The platform operator will be responsible for verifying the identities of borrowers and lenders and ensuring compliance with regulatory requirements.

Overall, the stakeholders' roles in the system are interconnected, and the success of the platform depends on the cooperation and trust between borrowers, lenders, and the operator. The

platform's success will be measured by its ability to attract and retain a large user base, facilitate timely and secure transactions, and ensure the transparency and fairness of the lending process.

2.9.2 Key High-Level Goals and Problems of Stakeholders

As per the stakeholder's summary, the key high-level goals and problems of the stakeholders can be summarized as follows:

1. Lenders: The lenders are looking for a reliable platform where they can lend their money and earn interest on it. They are concerned about the security and transparency of the lending process, as well as the timely repayments by the borrowers.
2. Borrowers: The borrowers are looking for a hassle-free and convenient way to borrow money. They want a platform that offers flexible repayment terms and low interest rates. They are also concerned about the security and privacy of their personal and financial information.
3. Platform Owners: The platform owners are looking to provide a trustworthy and efficient platform for lenders and borrowers to conduct business. They want to create a sustainable business model that generates revenue through fees and commissions on transactions.
4. Regulators: Regulators are concerned about the legality and compliance of the platform. They want to ensure that the platform operates in accordance with the laws and regulations governing lending and financial transactions.
5. Society: Society benefits from the platform by having access to affordable credit options, which can lead to increased economic activity and job creation. The platform can also help to reduce the prevalence of loan sharks and other predatory lending practices.

Overall, the key high-level goals of the stakeholders are to create a safe, reliable, and efficient lending platform that benefits all parties involved. The problems they face include ensuring security, transparency, legality, and compliance while generating revenue and providing affordable credit options to borrowers.

Chapter 3: Related Work

This section covers the work, studies, and development done in the past by other teams that is related to or somewhat overlaps with the work we will be conducting in this project.

3.1 Explainable AI in Credit Risk Management

This is a study conducted by students at Columbia University and Zurich University of Applied Sciences, which was published in 2021. The researchers were investigating the prospects of using AI and ML for developing a model for credit management.

3.1.1 Summary of the study

In this study, the researchers applied Local Interpretable Model Agnostic Explanations (LIME) and SHapley Additive explanations (SHAP) modelling techniques to a massive dataset offered by a major United States lending body, called the Lenders Club. They explain the results obtained by the afore-mentioned analysis algorithms, and their findings include important attributes from profiles of credit borrowers, which show trends that juxtapose their ability to meet their credit commitments. The research concluded that there are attributes of financial profiles that can be very good candidates for training a ML algorithm and latter be used for predicting likelihood of other entities meeting their credit-commitments.

3.1.2 Relation to our work

This research highlights that there are trends in financial profiles of entities which can be shown as a mapping between indicators within the profile against their ability to meet their credit liabilities. This backs our working logic as our model relies on the possibility of reliably predicting ability to meet credit by statistical data from financial profiles and trade records. They also highlighted two algorithms (Namely LIME and SHAP) which can be used to parse a huge dataset and spot out trends and correlations in seemingly unrelated data. These algorithms will be helpful when picking out correlated attributes from the dataset which we will be using to train our ML algorithm.

3.2 ANN Approach for Credit Risk Management

This research by Cornell University analyzes performance of multiple Artificial Neural Networks for forecasting the credit risk of a list of Italian companies.

3.2.1 Summary of the study

This used data from 2004, which consisted of financial records of an array of Italian manufacturing companies. They tested the performance of two different ANNs and compared the results with the actual outcomes. The study also helps determine the construction of ANN (such as the number of hidden layers) and the effects of changes in construction of the ANN on the accuracy of results.

3.2.2 Relation to our work

The results from this study can be used to further weigh the merits of using AI techniques for Credit Risk Management. It also shows us the possibility of using ANN or other perceptron-reliant techniques to serve as the backbone of our ML algorithm. It also hints at strategies that can be used to optimize the ANN if it is opted, in order to maximize the reliability of the predictions it makes.

3.3 Machine learning and AI for Risk Management

A study published by Rennes school of business in France, discusses the possible approaches for employing AI to determine credit and default risks.

3.3.1 Summary of the study

This study uses applied an array of techniques to a uniform dataset and evaluates the results and performance for each of these techniques. The tested techniques included Perceptron learning, Regression, Support vectors, Decision trees, and Clustering etc.

3.3.2 Relation to our work

The comparison of success rates of different techniques will be helpful when determining the program structure for our Machine Learning Algorithm. It will also help us explore possibilities for expanding our project beyond analytics and transaction assistance – as a portion of this study covers several other domains of finance that can benefit from assistance of reliable AI-based predictive tools with risk management.

3.4 Provenir

Provenir is a company based in New Jersey that uses AI based predictive models to assist businesses with their decision-making.

3.4.1 Summary of the study

Provenir has an AI-powered platform that assists lenders with managing credit risk. Their services include risk assessment, fraud detection, compliance management, and decision analytics. The company uses undisclosed AI and ML techniques to evaluate borrower credibility and helps lenders decide who to work with.

3.4.2 Relation to our work

There is a degree of overlap between the prospective services we hope to offer with our project, and what Provenir offers to their customers. Our AI will be generating similar Analytics for the decision suggestions, though we hope to go a step further and assist with the actual transaction too.

3.5 Zest AI

Zest AI is a California based company, which has developed a platform which allows borrowers to get credit risk assessments for individual lenders.

3.5.1 Summary of the study

The platform is advertised as a highly customizable interface that can be tailored for different kinds of lenders and accepts data from a variety of sources including credit bureau authorities, banking records, and other internal data that can be manually input by the lender.

3.5.2 Relation to our work

Zest AI's services overlap with the project we are working on. It also uses AI based predictive techniques on borrower's financial data to make predictions about their ability to fulfill credit liabilities. It also has partnerships with financial institutions which helps them fine-tune their

analytics with authentic and up to date data from several resources – which bears some similarity to the feedback loops and ranking mechanism we plan on implementing for borrowers who are in business with multiple lenders.

3.6 Conclusion

In conclusion, the related work shows that there is significant research and development in the use of AI and machine learning techniques for credit risk management. Various studies have explored different models and algorithms for predicting credit risk and improving decision-making processes for lenders. Companies such as Provenir and Zest AI are already providing AI-based solutions in this area. However, our project is different in that we aim to provide a platform that not only offers AI-based credit risk assessment but also facilitates the actual transaction process between lenders and borrowers. We also plan to incorporate feedback loops and ranking mechanisms that will help lenders make informed decisions and manage their risks better. In essence, our project is a holistic approach that combines both credit risk management and transaction facilitation.

Chapter 4: Software Requirement Specifications

This section includes all the expected requirements of our software for its proper functioning, along with the details of implementation, functional requirements, and the planned use cases and GUI outlook.

4.1 List of Features

1. Creation and management of contracts between lenders and borrowers.
2. Automated payment processing and distribution using smart contracts.
3. AI-powered risk assessment to determine creditworthiness of borrowers.
4. Recording of all transactional data on a blockchain for transparency and security.
5. User-friendly web-based interface with React as the front-end technology.
6. Integration with bank transfer systems for seamless payment processing
7. Notification system for timely reminders and updates on contract status
8. Multi-language support to cater to a global user base.
9. Secure login and authentication system with role-based access control
10. Real-time analytics and reporting for lenders to monitor their investments and track performance.

4.2 Functional Requirements

1. User Authentication - Users should be able to create an account and log in to the system securely. This functionality will be used by all users of the system.
2. Create Listings - Lenders should be able to create new listings for their consignments, specifying all the necessary details like price, quantity, location, and timeline. This functionality will be used by lenders.
3. Search Listings - Borrowers should be able to search for consignments based on various parameters like location, price, and quantity. This functionality will be used by borrowers.
4. Request Listings - Borrowers should be able to request a particular consignment from a lender. This functionality will be used by borrowers.
5. Accept/Reject Requests - Lenders should be able to accept or reject borrower requests based on their own criteria. This functionality will be used by lenders.
6. Payment Processing - The system should be able to process payments from borrowers to lenders securely and transparently. This functionality will be used by both borrowers and lenders.
7. Transaction History - The system should maintain a record of all the transactions that take place on the platform. This functionality will be used by both borrowers and lenders.
8. Smart Contract Integration - The system should be integrated with a smart contract to automate the payment processing and enforce the terms of the transaction. This functionality will be used by the system itself.
9. Communication - The system should provide a communication channel between lenders and borrowers to discuss the details of the consignment and transaction. This functionality will be used by both borrowers and lenders.
10. Reporting and Analytics - The system should provide reports and analytics on various aspects of the platform like transaction history, user behavior, and inventory. This functionality will be used by the system administrators.

4.3 Quality Attributes

One of the most critical quality attributes for our project is reliability. As a web-based platform, users will rely heavily on our system to perform their business transactions smoothly and

securely. Therefore, we must ensure that the system is stable and performs reliably under various load conditions. Another crucial quality attribute is security, as we are dealing with sensitive financial transactions. We need to implement robust security measures to protect user data and prevent unauthorized access to the system. Additionally, we must consider usability as a quality attribute to ensure that the platform is easy to navigate and use for our target audience. Finally, scalability is an important quality attribute to consider as our platform will need to support a growing user base and handle increasing amounts of data as the business grows.

4.4 Non-Functional Requirements

1. Performance: The system should be able to handle many concurrent users and data transactions without significant delays or errors. The system should be able to handle at least quite reasonable simultaneous users.
2. Reliability: The system should be highly reliable and available at all times. The system should have a backup and disaster recovery plan in place in case of system failures or data loss.
3. Security: The system should be designed with robust security features to ensure that all user data is secure and protected from unauthorized access or malicious attacks. The system should use encryption techniques to ensure that all data transmissions are secure.
4. Usability: The system should be user-friendly and easy to navigate for all types of users. The system should provide clear and concise instructions and guidance for users to complete tasks efficiently and effectively.
5. Maintainability: The system should be easily maintainable, and updates or enhancements should be easy to implement. The system should also be designed to be scalable to accommodate future growth or changes in business needs.
6. Reusability: The system should be designed to be modular and reusable, with the ability to be integrated with other systems or applications in the future.
7. Extensibility: The system should be designed to be flexible and easily extensible to accommodate future changes or enhancements to the system. The system should be designed with a clear and well-documented architecture to facilitate future development and maintenance.

4.5 Assumptions

1. The users of the system have a basic understanding of how to use web applications.
2. The users have access to a reliable internet connection and a compatible web browser.
3. The data provided by the users is accurate and up to date.
4. The system will be hosted on a secure and reliable server.
5. The system will be developed and tested on the latest version of the chosen development platform.
6. The users will have the necessary permissions to access and perform all the required functionalities.
7. The system will not be responsible for any fraudulent activities performed by the users.
8. The data provided by the users will be stored securely and will only be accessible to authorized personnel.
9. The system will be developed within the allocated time and budget constraints.
10. The system will be compliant with relevant laws and regulations.

4.6 Hardware and Software Requirements

4.6.1 Software Requirements

- Operating System: Windows, macOS, or Linux
- Web Browser: Chrome, Firefox, or Safari
- Node.js: version 14.x or later
- NPM (Node Package Manager): version 6.x or later
- React: version 16.x or later
- Ethereum blockchain client: Geth or Parity
- Solidity: version 0.5.x or later

Note: These software requirements may be subject to change based on the development progress and specific implementation details.

4.7 Use Cases

This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large architectural coverage—they exercise many architectural elements or if they stress or illustrate a specific, delicate point of the architecture.

4.7.1 Login

Name	Log-in		
Actors	Whole-seller, Retailer		
Summary	The user shall provide their email and password on the login form and after successful verification, redirect the user to the home page.		
Pre-Conditions	The user must be in the database records either added by any of the authorized users or added manually by a developer. The user must not already be logged in.		
Post-Conditions	The user's session is successfully established and shall be redirected to the home page.		
Special Requirements	None		
Basic Flow			
Actor Action	System Response		
1 The user opens the login page.	2 The login page is displayed asking for email and password.		
3 The user enters valid email and password.	4 The system verifies the email and password, establishes a session for the user and redirects the user to the home page.		
Alternative Flow			
3 The user enters invalid email or password.	4-A The system responds with an error message: <i>Incorrect email or password entered.</i>		

4.7.2 Sign Up

Name	Sign Up		
Actors	Whole-seller, Retailer		
Summary	The user shall provide their name, email, date of birth, CNIC, age and password on the sign in form and after successful verification, redirect the user to the payment method page.		
Pre-Conditions	The client must not be in the database records.		
Post-Conditions	The user's session is successfully established and shall be redirected to the payment method page.		
Special Requirements	None		
Basic Flow			
Actor Action	System Response		
1 The user opens the sign-up page.	2 The sign-up page is displayed asking for name, email, date of birth, CNIC, age and password.		
3 The user enters valid name, email, date of birth, CNIC, age and password.	4 The system verifies that the name, email, date of birth, CNIC, age and password does not already exist in the database, establishes a session for the client and redirects the client to the payment method page.		
Alternative Flow			
3 The user enters the name, email, date of birth, CNIC, age and password that already exists in the database	4-A The system responds with an error message: <i>Already a user.</i>		

4.7.3 Log out

Name	Log out		
Actors	Whole-seller, Retailer		
Summary	The user shall click on the log out button and after successful verification, redirect the user to the payment method page.		
Pre-Conditions	The user must be logged in the system.		
Post-Conditions	The client's session is successfully deleted and shall be redirected to the login page.		
Special Requirements	None		
Basic Flow			
Actor Action	System Response		
1 The client clicks on the log out button	2 The system logs out the client and redirect to the login page.		

4.7.4 View Client Options

Name	View Client Options		
Actors	Registered Retailer/Wholesaler		
Summary	This use case allows the user to view the various options available to them as a retailer or wholesaler on our platform		
Pre-Conditions	The user must be a registered retailer or wholesaler. The user must have logged in to the system. The user must have the necessary privileges to view retailer/wholesaler options.		
Post-Conditions	The user is able to view the available retailer/wholesaler options. The user can select an option to proceed further.		
Special Requirements	None		
Basic Flow			
Actor Action	System Response		
1 The user navigates to the retailer/wholesaler options page.	2 The system displays a list of available retailer/wholesaler options.		
3 The user selects an option to proceed further.	4 The system navigates the user to the selected option.		
Alternative Flow			
3 If the user is not logged in	4-A The system prompts them to log in first.		

4.7.5 Add Local Retailer/Whole seller

Name	Add Local Retailer/Whole seller		
Actors	Retailer/Lender		
Summary	The user can search for account information of other users by entering their ID.		
Pre-Conditions	The actor is authenticated and authorized as a user. The system has user data stored in its database.		
Post-Conditions	The user has found the account information of the user they searched for.		
Special Requirements	The system should have a robust search function that allows for easy and accurate retrieval of user data. The user ID should be unique and easily identifiable to the user. The system should enforce appropriate access control to prevent unauthorized access to sensitive user data.		
Basic Flow			

Actor Action		System Response	
1	The user navigates to the search page.	2	The system prompts the user to enter the ID of the user they want to search.
3	The user enters the ID of the user they want to search	4	The system displays detailed information about the user's account, such as their name, contact information, credit score, and other relevant details.
Alternative Flow			
3	The user enters invalid ID	4-A	The system responds with an error message: <i>The entered ID does not exist</i>
3	The user sends the other user a link to make account	4-B	The system then links the accounts

4.7.6 Generate Credit Score

Name	Generate Credit Score		
Actors	Admin, Business Owner, Store Manager		
Summary	Users can generate a credit score based on their transaction history, which is used to determine their creditworthiness for future transactions on the platform.		
Pre-Conditions	The user must be in the database records either added by any of the authorized users or added manually by a developer. The user must not already be logged in.		
Post-Conditions	The user's session is successfully established and shall be redirected to the home page.		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	The user logs into the system and accesses the credit score generation feature.	2	The System asks the user to upload the required documents.
3	The user uploads the required documents.	4	The system displays the generated credit score to the user
Alternative Flow			
3	The retailer hasn't uploaded the documents.	4-A	The system responds with an error message: <i>Documents not uploaded yet</i>

4.7.7 Agree upon T&Cs.

Name	Mutual terms and conditions
-------------	-----------------------------

Actors	Retailer/Lender		
Summary	The user can search for account information of other users by entering their ID.		
Pre-Conditions	<p>Lender and borrower have created their accounts and logged in to the credit risk management system.</p> <p>Lender and borrower have initiated a loan request.</p>		
Post-Conditions	<p>The lender and borrower have agreed upon the final loan terms and conditions.</p> <p>The credit risk management system has generated a loan agreement with the final terms and conditions that have been electronically signed by the lender and borrower.</p>		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	Lender and borrower review the loan details including loan amount, interest rate, repayment schedule, and collateral requirements.	2	The system displays the loan details for review
3	Lender and borrower agree upon the final loan terms and conditions.	4	The credit risk management system generates a loan agreement with the final terms and conditions.
Alternative Flow			
3	The borrower is unable to pay the agreed upon amount in time	4-A	The system reschedules the date to pay back and deducts the credit score

4.7.8 Verify consignment status

Name	Verify consignment status		
Actors	Registered whole seller/ retailer.		
Summary	The user must verify the delivery and dispatchment of the consignment from their ends respectively.		
Pre-Conditions	The credit score generated suggests that the party won't default. T&Cs should be agreed upon both parties.		
Post-Conditions	Transactions and payments will be done and viewed.		
Special Requirements	None		
Basic Flow			
Actor Action		System Response	
1	The whole seller verifies that he has dispatched the consignment.	2	The system notifies the retailer and asks him to verify back
3	Retailer verifies about the delivered consignment.	4	The system verifies both sides and lets the users start the next step.

Alternative Flow			
3	Any user doesn't verify from their end.	4-A	The system notifies both the parties and asks them to verify to start the transactions.

4.7.9 View Analytics

Name	View Analytics		
Actors	Admin, Wholesalers/Retailor		
Summary	Allows the user to view their financial analytics over a period of time		
Pre-Conditions	The user will have all the data from the start point where he starts the business on the app, this data will be used in analysis.		
Post-Conditions	The user gets the analytics based on the performance online.		
Special Requirements	The user's record is all maintained on the app by the app itself and is preparing the analytics.		
Basic Flow			
Actor Action		System Response	
1	The user goes on to the analytics part.	2	The system shows the overall performance
3	The user has performed some activity on the app i.e., made some business.	4	The system uses this activity to generate the analytics and further depict it as the overall performance.
Alternative Flow			
3	The user has not performed any activity till now.	4-A	The system will not be able to provide any relevant information as nothing's there to analyze.

4.7.10 View Transactions

Name	View Transactions		
Actors	Admin, Wholesaler/Retailor		
Summary	All the activity performed by the user is recorded onto the app and all the transactions are recorded and these transactions can be viewed under the transactions part.		
Pre-Conditions	The user performs activity so that it can generate Transactions.		
Post-Conditions	All the transactions by the activities within the app are stored in the Transaction part.		
Special Requirements	The activity needs to be performed.		
Basic Flow			
Actor Action		System Response	
1	The user opens the Transaction page.	2	The system shows the transaction record of

			all the activities in the past.
3	The user performs the activities to add them into transactions.	4	The system adds them into the user's transactions

Alternative Flow

3	The activity is being performed without taking app into account.	4-A	The system will not be able to generate any of such Transaction history where any deal was done outside the app.
---	--	-----	--

4.7.11 Compare Retailers

Name	Compare Retailors
Actors	Retailer/wholesaler, System.
Summary	The User Can select for the retailors to be compared for a better understanding of the best possible options.
Pre-Conditions	User must be logged into the system and have selected the retailors to be compared.
Post-Conditions	The system generates the result giving the pros and cons in both of the options selected.
Special Requirements	The user must select the retailors and then select the compare option.

Basic Flow

Actor Action		System Response	
1	The user gets on to the Comparison page	2	The System takes the user to the Comparison page giving options of retailors the user wants to compare.
3	The user selects the best of retailors as per his/her wish.	4	The system brings up the result of comparison, depicting a clear picture of the retailer's actual market position.

Alternative Flow

3	The user selects only one retailer	4-A	The system gives an error as a person can't be compared with himself.
---	------------------------------------	-----	---

4.7.12 Get Best Suggestions

Name	Get Best Suggestions
Actors	Wholesaler
Summary	The user shall be provided with the best of suggestions among retailers once the credit ratings of particular selected retailer are under the mark.
Pre-Conditions	User must be logged into the system and must be assigned credit score.
Post-Conditions	User receives suggestions based on the credit scores.

Special Requirements	The system must provide the credit rating in order to suggest the retailer of the best available choices.		
Basic Flow			
Actor Action		System Response	
1	The user gets on the suggestion page	2	The suggestion page with all its categories appears.
3	User click on get best suggestion button.	4	Retrieves the credit scores and matches with corresponding suggestions.
Alternative Flow			
3	The user goes with 'already have a customer' option.	4-A	The system takes it directly to the 'search by ID' page as the user has already got the customer.

4.8 Graphical User Interface

This section gives the GUI dumps of each screen, with reference to the users. The navigation flow of each user is also required, and each GUI should mark the functionality/use case that it covers.

4.8.1 Main Page

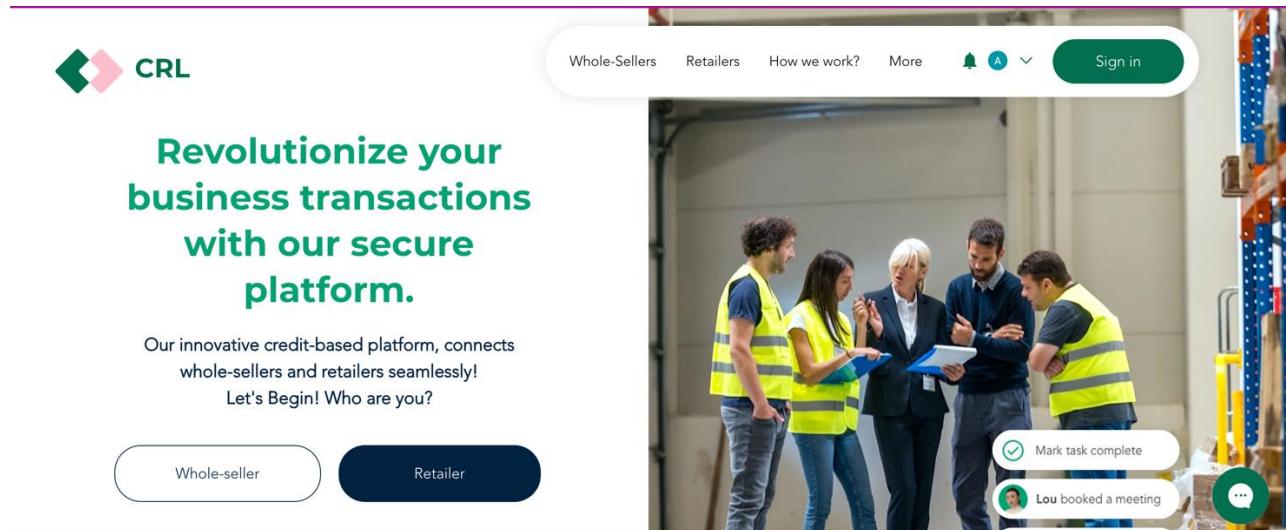


Figure 2: Main Page
This figure depicts the main page of our application

4.8.2 Log In

The image shows the login page of an application. At the top right is a close button (X). Below it is a large 'Log In' heading. Underneath the heading is a subtext 'New to this site? [Sign Up](#)'. There are two input fields: 'Email' and 'Password'. Below the password field is a 'Forgot password?' link. A prominent blue rectangular button contains the white text 'Log In'. Below this button is a horizontal line with the text 'or log in with' followed by icons for Facebook and Google.

Figure 3: Login Page
This page shows the login page of our application.

4.8.3 Sign Up

The image shows the sign-up page of an application. At the top center is a purple header with the text 'Let's Connect' and a large 'Sign Up' button. Below the header is a subtext 'We are this close to you. You mean a lot to us.' The main content area is divided into three columns. The left column contains icons for location ('Address: 500 Terry Francois St, San Francisco, CA 94158'), WhatsApp ('WhatsApp: +44 7459477824'), and email ('Email: info@crl.com'). The middle column contains an icon for social media and links to various platforms. The right column contains a form with fields for 'First Name *' and 'Last Name', 'Email *', 'Phone *' (with a dropdown for 'PK +92'), and a 'Message' text area. At the bottom is a reCAPTCHA checkbox labeled 'I'm not a robot' and a blue 'Sign Up' button.

Figure 4: Sign Up Page
This is the sign-up page of our application.

4.8.4 Analytics

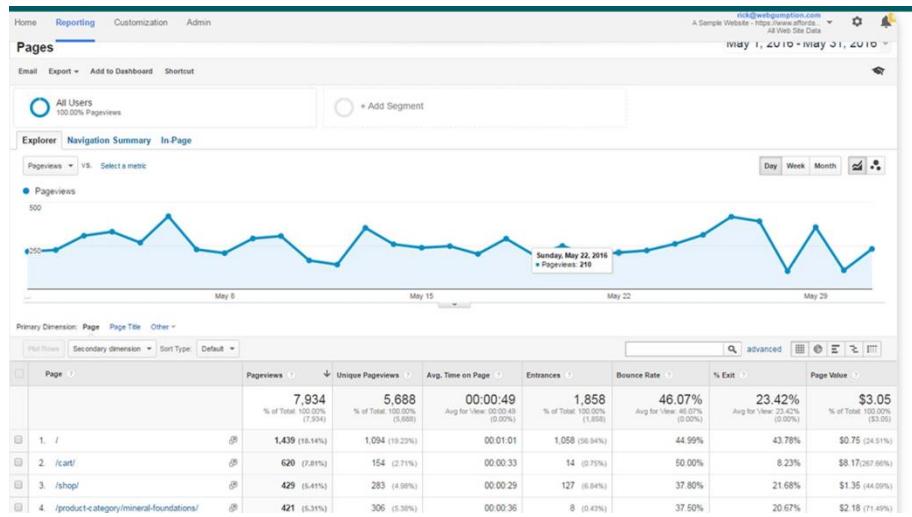


Figure 5: Analytics Page
This is the analysis page of our application.

4.8.5 Terms & Conditions

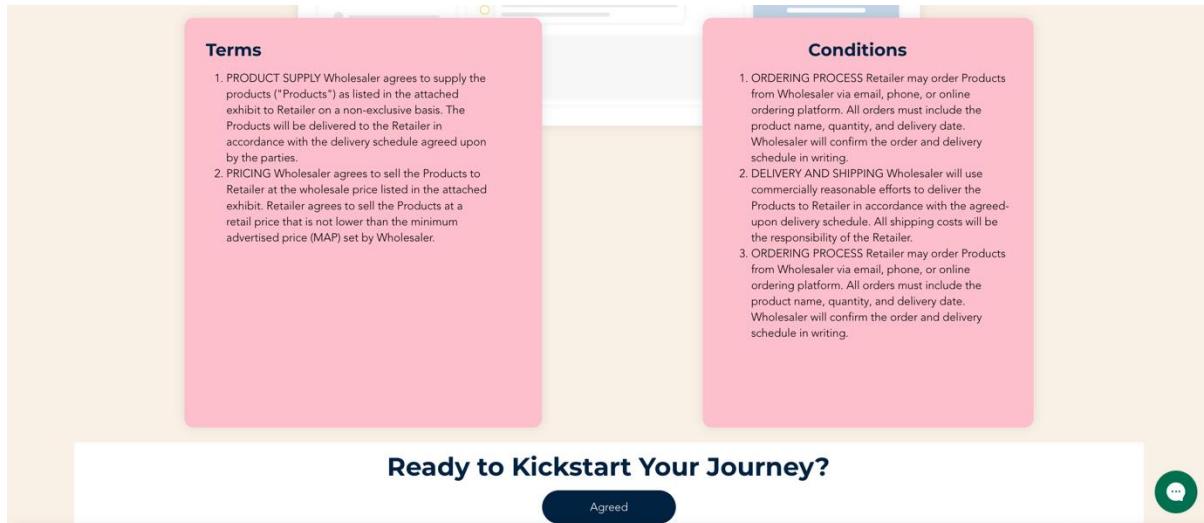


Figure 6: Terms and Condition Page
This shows the policies of our website.

4.8.6 Getting Credit Scores

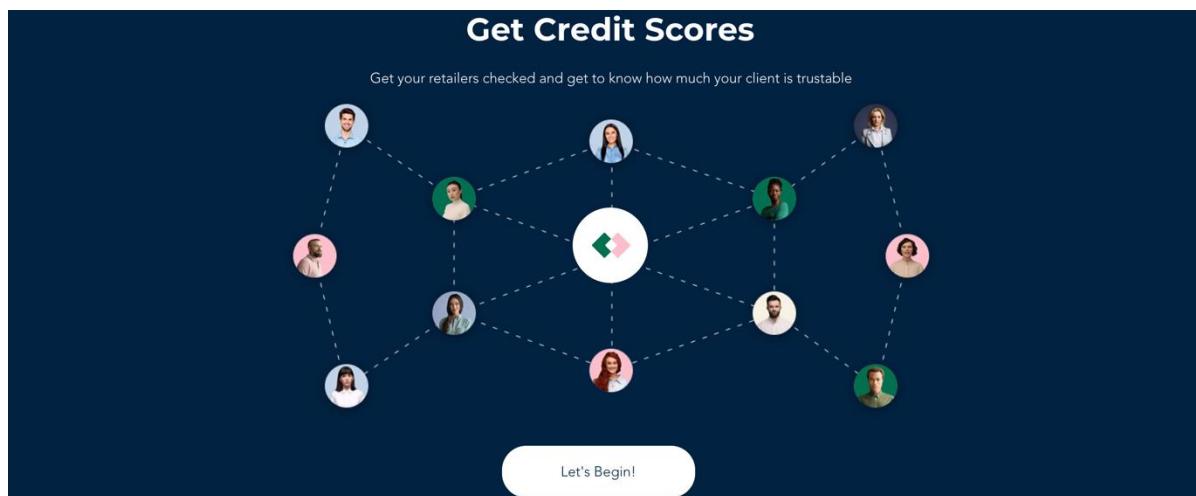


Figure 7: Credit Scores Page
This page shows the credit scores page of our application.

4.8.7 Client Options

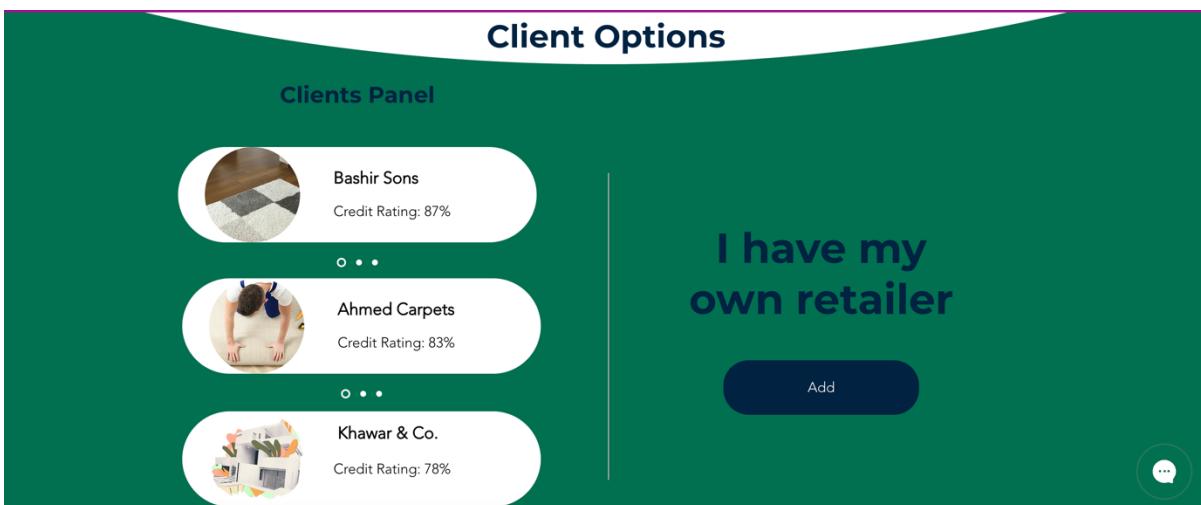


Figure 8: Client Options
This figure depicts the client-side page of our project.

4.9 Database Design

4.9.1 ER Diagram

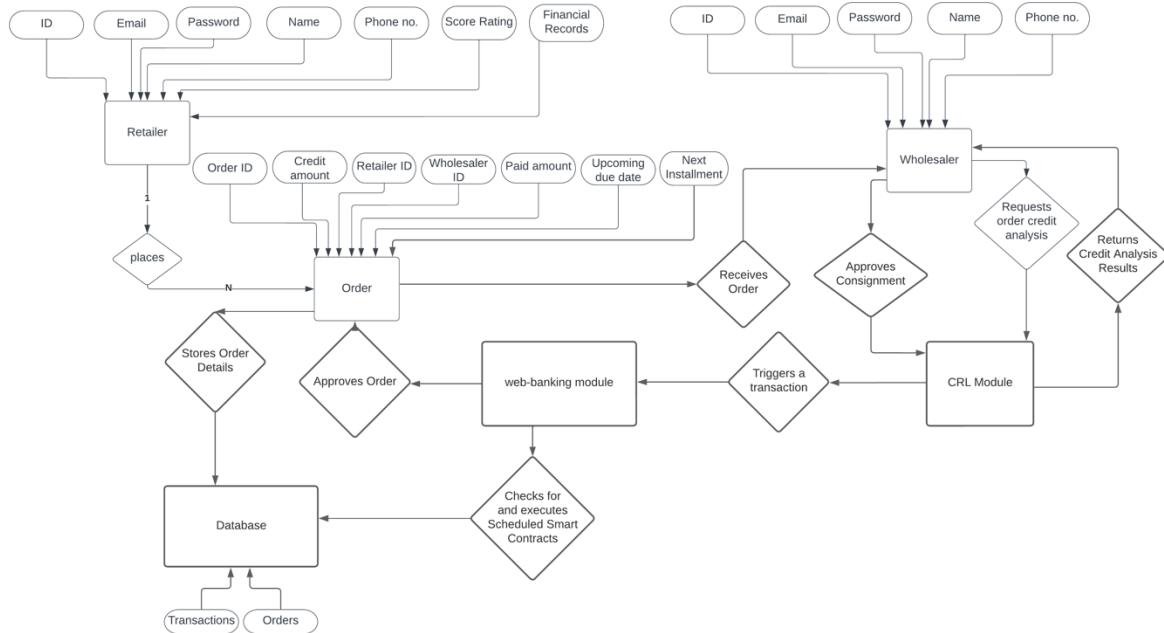


Figure 9: ER Diagram
This figure depicts the ER diagram of our application.

4.9.2 Data Dictionary

Table 1: Data Dictionary
This table gives an overview of our database.

Entity	Attribute	Data Type	Nullable	Description
Retailer	id	int	No	Retailer's unique id (primary key)
	name	string	No	Retailer's name
	email	string	No	Retailer's email address
	password	string	No	Retailer's password
	phoneNo	string	No	Retailer's phone number
	scoreRating	string	Yes	Retailer's credit score
	financialRecords	array	Yes	Retailer's financial records
Whole-Seller	id	int	No	Seller's unique id (primary key)
	name	string	No	Seller's name
	email	string	No	Seller's email
	password	string	No	Seller's password

	phoneNo	string	No	Seller's phone number
Order	orderId	int	No	Order's unique ID
	credit	float	No	Total credit due on the order
	retId	int	No	Retailer (borrower) ID
	whId	int	No	Seller (lender) ID
	amountPaid	float	No	Amount paid so far
	nextPayment	float	No	Amount due for next installment
	nextDate	date	No	Deadline for next installment
Database	transactions	DBArray	yes	Details of all transactions
	orders	DBArray	yes	Details of all orders

4.10 Risk Analysis

1. Technical risks: These are risks associated with the technology used in the project, such as the possibility of software bugs, hardware failures, or security breaches. To mitigate these risks, the project team will need to conduct thorough testing and implement appropriate security measures.
2. Business risks: These are risks associated with the business side of the project, such as the possibility of low demand for the product, competition from other companies, or economic downturns. To mitigate these risks, the project team will need to conduct market research, analyze the competitive landscape, and create a solid business plan.
3. Resource risks: These are risks associated with the availability of resources required for the project, such as human resources, equipment, or funding. To mitigate these risks, the project team will need to create a detailed resource plan and ensure that adequate resources are allocated to each phase of the project.
4. Schedule risks: These are risks associated with the project timeline, such as delays in the development process or unexpected changes in requirements. To mitigate these risks, the project team will need to create a detailed project plan, regularly monitor progress, and be prepared to adjust the schedule as needed.
5. Legal and regulatory risks: These are risks associated with compliance with laws and regulations, such as data privacy laws or export regulations. To mitigate these risks, the project team will need to conduct a thorough analysis of the legal and regulatory environment and ensure that the project is designed to comply with all applicable laws and regulations.

Chapter 5: High-Level and Low-Level Design

This chapter outlines the comprehensive design of the Credit Risk Liaison, including both high-level and low-level aspects. It covers various considerations such as system overview, dependencies, assumptions, general constraints, and the overall architecture of the system. The chapter also includes detailed class and sequence diagrams that depict the project's workflow, along with the policy tactics used to ensure its smooth functioning.

5.1 System Overview

Provide The Credit Risk Liaison system is a web-based platform that provides a secure and efficient way for wholesalers and retailers to connect and conduct business transactions. The system integrates three main technologies: MERN stack for the web-based application, AI modelling for credit score generation and predictive analysis, and Blockchain for the back-end database.

The system aims to reduce credit risk for wholesalers and retailers by providing accurate and real-time credit scores generated by AI modelling. It also streamlines the transaction process, making it easier for both parties to keep track of their business deals and payments.

The system has two main components: the web-based application and the back-end database. The web-based application provides a user-friendly interface that allows wholesalers and retailers to register, connect, and conduct transactions securely. The back-end database is a blockchain-based ledger that stores all the transaction and payment history.

5.1.1 Web-Based Application

The web-based application is built on the MERN stack, which includes MongoDB as the database, Express as the web application framework, React as the front-end library, and Node.js as the server-side runtime. The application is designed to provide a seamless user experience with minimal technical barriers.

The application consists of different modules, including registration, connection, transactions, and analytics. The registration module allows new users to register as either a wholesaler or a retailer. The connection module enables wholesalers and retailers to connect and transact with each other. The transaction module facilitates the exchange of goods and payments between the connected parties. The analytics module provides real-time analytics and insights into the transaction and credit score data.

5.1.2 AI Modeling

The credit score generation and predictive analysis are powered by an AI model that has been trained on a large dataset picked from the market. The model uses various data points such as credit history, transaction history, and financial statements to generate credit scores. The scores are then used to assess the creditworthiness of the retailers and provide risk analysis for wholesalers.

The AI model uses a combination of supervised and unsupervised learning algorithms, including decision trees, random forests, and neural networks. The model is continually learning and improving its accuracy over time by analysing new transaction data.

5.1.3 Blockchain-based Database

The back-end database is built on a blockchain-based ledger that provides a secure and tamper-proof record of all the transactions and payment history. The database is distributed and replicated across multiple nodes, ensuring data redundancy and high availability. It provides several benefits, including increased security, transparency, and immutability. It also enables the system to support smart contracts, which are self-executing contracts that automatically trigger transactions when specific conditions are met.

Overall, the Credit Risk Liaison system is designed to provide a comprehensive solution for credit risk management and efficient transaction processing for wholesalers and retailers. The system leverages the latest technologies in AI modelling, blockchain, and web application development to provide a seamless and secure user experience.

5.2 Design Considerations

Here are some design considerations that need to be addressed for our Credit Risk Liaison project:

1. User Interface Design: Our web-based application should have an intuitive and user-friendly interface to ensure that users can navigate the platform easily. We need to ensure that the platform is accessible on multiple devices, including desktop, tablet, and mobile devices.
2. Security: We need to ensure that the data of our users is secure and that the platform is not susceptible to hacking attempts or data breaches. We will need to use encryption, secure authentication protocols, and other security measures to safeguard user data.
3. Scalability: The platform needs to be designed to handle a large number of users and transactions and be scalable as the business grows.
4. Integration: The platform needs to be integrated with different technologies like MERN, AI-Modelling and Blockchain, ensuring smooth communication among them.
5. Performance: The platform should be designed to deliver high performance, with minimal downtime, and fast response times.
6. Data Management: As the AI-Modelling will be the heart of the system, we need to ensure that the data is well-managed, and the database is optimized for efficient data storage and retrieval.
7. Regulatory Compliance: We will have to make sure that our platform complies with any regulatory requirements applicable to the financial industry, especially related to data privacy and security.
8. Maintenance: The platform needs to be designed to be easy to maintain, and any updates or bug fixes should be easy to deploy without disrupting the platform's functionality.
9. Cost-effectiveness: Our solution should be cost-effective and efficient in terms of development, deployment, and maintenance. We will need to use open-source technologies where possible and explore cloud-based solutions for hosting our application and database.

These are some of the key design considerations that need to be addressed to ensure that our Credit Risk Liaison platform is efficient, scalable, and user-friendly.

5.2.1 Assumptions and Dependencies

5.2.1.1 Related Software or Hardware

The software system will require a web browser and an internet connection. It will also require a compatible device, such as a computer, tablet, or smartphone.

5.2.1.2 Operating System

The platform is web-based and can be accessed on any device with a web browser, regardless of the operating system. However, the platform has been tested and optimized for use on desktop and laptop devices running Windows, macOS, and Linux operating systems.

5.2.1.3 End-user Characteristics

The platform is designed to be user-friendly and intuitive, but users will need to have basic knowledge of business operations and financial management to use it effectively. It is also assumed that users will comply with relevant laws and regulations governing their business activities.

5.2.1.4 Possible Changes in Functionality

As with any software project, there may be changes in functionality over time as user needs evolve and new features are developed. We plan to use an agile development methodology to respond to changes in user requirements and ensure that the platform remains relevant and useful to our target audience.

5.2.2 General Constraints

Following are the general constraints:

5.2.2.1 Hardware or Software Environment

The software may need to be designed to work with specific hardware or software environments, such as a particular operating system, web browser, or database management system.

5.2.2.2 End-user Environment

The software may need to be designed to work in specific end-user environments, such as a particular country or region, or with users who have limited access to technology.

5.2.2.3 Availability of Resources

The software may need to be designed to work with limited resources, such as limited bandwidth, processing power, or memory. Additionally, the availability of these resources may be volatile, which can impact the reliability of the system.

5.2.2.4 Standard Compliance

The software may need to comply with specific industry standards, such as security or privacy standards, which can impact the design of the system's security features.

5.2.2.5 Interoperability Requirements

The software may need to be designed to work with other systems or software, which can impact the design of the system's interfaces and protocols.

5.2.2.6 Interface Requirements

The software may need to be designed with specific interface or protocol requirements, such as RESTful API or SOAP.

5.2.2.7 Data Repository and Distribution Requirements

The software may need to be designed to store and distribute data in specific ways, such as using a distributed database or cloud storage system.

5.2.2.8 Security Requirements

The software may need to be designed with specific security requirements, such as encryption or multi-factor authentication, to protect sensitive data and prevent unauthorized access.

5.2.2.9 Memory Limitations

The software may need to be designed to work within specific memory or capacity limitations, which can impact the system's overall performance.

5.2.2.10 Performance Requirements

The software may need to be designed to meet specific performance requirements, such as response times or throughput, to ensure that the system can handle the expected workload.

5.2.2.11 Network Communication

The software may need to be designed to work with specific network communication protocols or technologies, such as TCP/IP or HTTP.

5.2.3 Goals and Guidelines

The design of the software system is guided by several key goals, guidelines, and principles that help ensure its functionality, usability, and maintainability. These goals include:

5.2.3.1 Simplicity

The software system is designed to be simple and easy to use, following the KISS principle ("Keep it simple stupid!"). This goal is desirable because it makes the software more intuitive for end-users, reduces the risk of errors, and simplifies maintenance and support.

5.2.3.2 Speed and Performance

The software system is optimized for speed and performance, with a focus on minimizing response times and maximizing throughput. This goal is desirable because it enhances user experience and satisfaction, reduces the risk of performance issues, and enables the system to handle larger volumes of data and transactions.

5.2.3.3 User-Centred Design

The software system is designed with a user-centred approach, prioritizing the needs, preferences, and limitations of end-users. This goal is desirable because it ensures that the system is intuitive and easy to use, enhances user engagement and adoption, and reduces the risk of user errors and dissatisfaction.

5.2.3.4 Consistency

The software system adheres to consistent design patterns, user interfaces, and behaviour across all modules and functions. This goal is desirable because it enhances user familiarity and comfort, reduces the risk of confusion and errors, and simplifies maintenance and support.

5.2.3.5 Modularity and Flexibility

The software system is designed with a modular architecture, enabling flexibility and scalability for future enhancements and modifications. This goal is desirable because it enables the system to adapt to changing user needs, business requirements, and technology trends, and reduces the risk of obsolescence and maintenance issues.

By adhering to these goals and guidelines, the software system is designed to meet the needs and expectations of its end-users, while ensuring optimal performance, usability, and maintainability.

5.2.4 Development Methods

The development method used for this software design is Agile methodology. Agile methodology emphasizes flexibility and adaptability, with an iterative and incremental approach to software development. It allows for frequent collaboration between developers and stakeholders, as well as continuous feedback and improvement throughout the development process.

Specifically, the Scrum framework within Agile methodology was adopted for this project. Scrum provides a structured approach to managing and organizing the development team's work through a series of sprints, each lasting for a set amount of time. The Scrum framework includes specific roles (e.g., Scrum Master, Product Owner, Development Team), meetings (e.g., Sprint Planning, Daily Stand-up, Sprint Review, Sprint Retrospective), and artifacts (e.g., Product Backlog, Sprint Backlog, Increment) to facilitate effective communication and collaboration among team members.

While other development methods were considered, such as Waterfall and Spiral, Agile methodology was ultimately chosen due to its flexibility and ability to accommodate changes and evolving requirements throughout the development process. Additionally, the team had previous experience with using Agile methodology and felt that it would be the most suitable approach for this project.

5.3 System Architecture

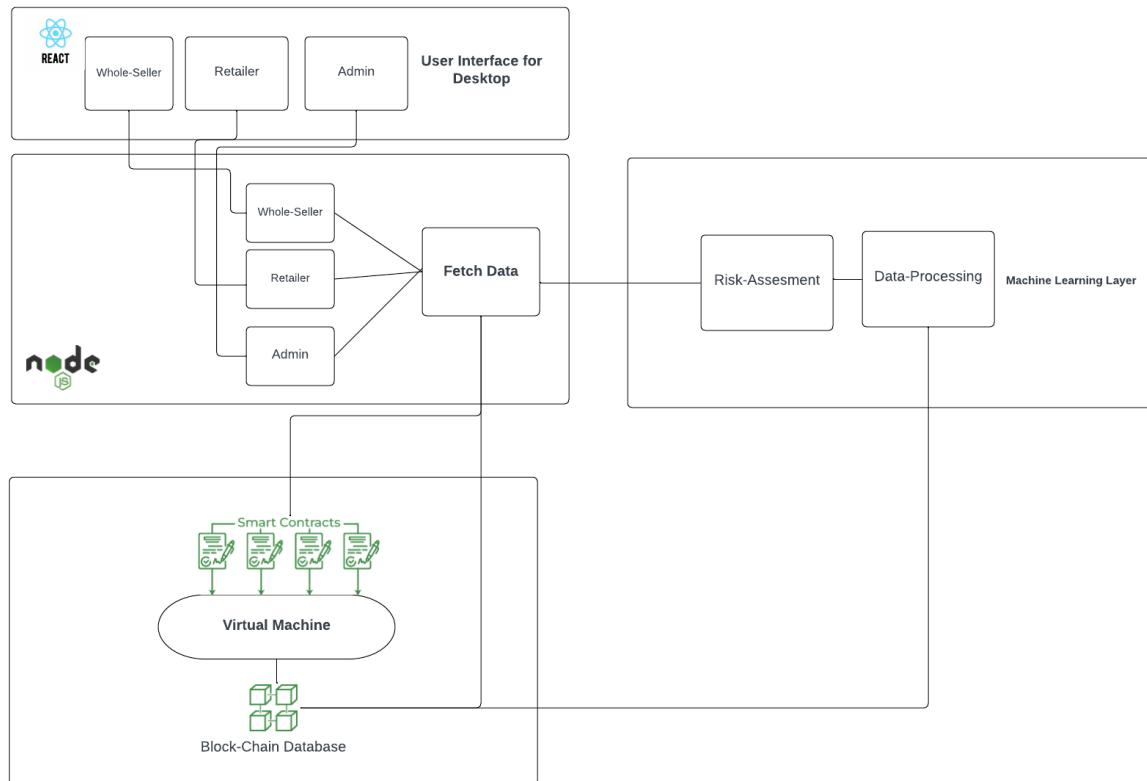


Figure 10: System Architecture
This figure shows the system architecture of our application.

As shown in the system architecture, we'll implement a client-server structure for our system, where users can send requests to the server to perform their respective functions. Three types of users can use this system, an admin, a retailer, and a whole seller. Our backend server will consist of 3 modules (an admin, a retailer, and a whole seller). The backend server will be able to access the database to store or retrieve any valuable data for executing functions. After performing requested functions/operations at the server, it will send a response to the user showing the results or status of their sent request.

5.3.1 Subsystem Architecture

5.3.1.1 Admin Component

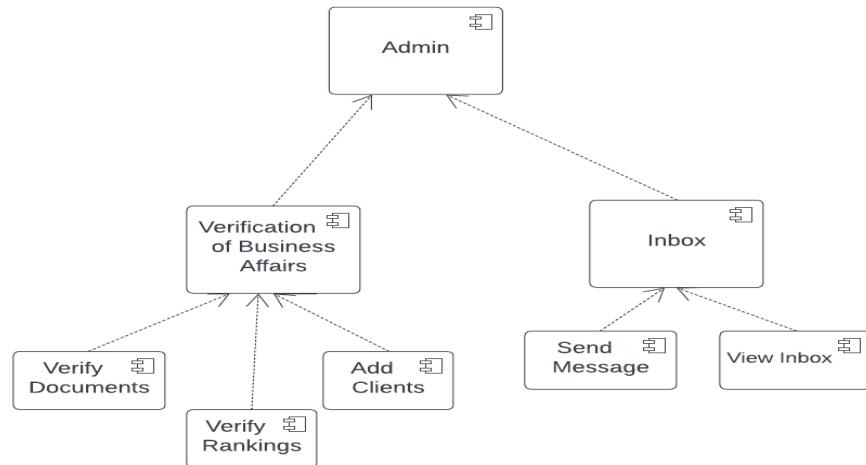


Figure 11:Admin Module
This diagram depicts the component diagram for admin module.

As shown in the system architecture, we'll be implementing client-server structure for our system, where users can send requests to server to perform their respective functions. Three types of users can use this system, an admin, an instructor, and a student. Our backend server will consist of 3 modules (Admin, Teacher, and Student). The backend server will be able to access database to store or retrieve any valuable data for executing functions. After Performing requested functions/operations at the server, it will send a response to the user showing results or status of their sent request.

5.3.1.2 Retailer Component

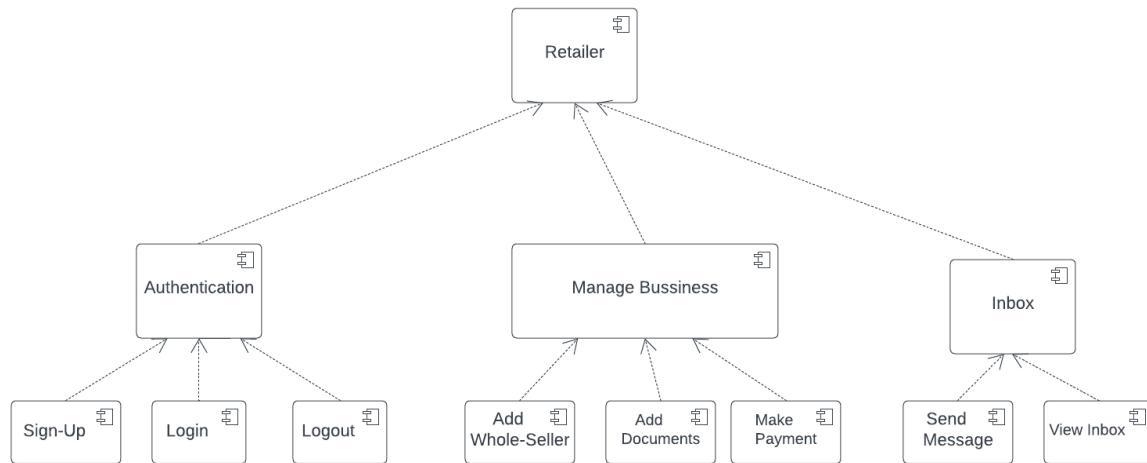


Figure 12: Retailer Component
This diagram depicts the component diagram for retailer component.

The retailer component in our application will be responsible for managing business, sending and viewing messages and authentication of retailers registered.

5.3.1.3 Wholesaler Component

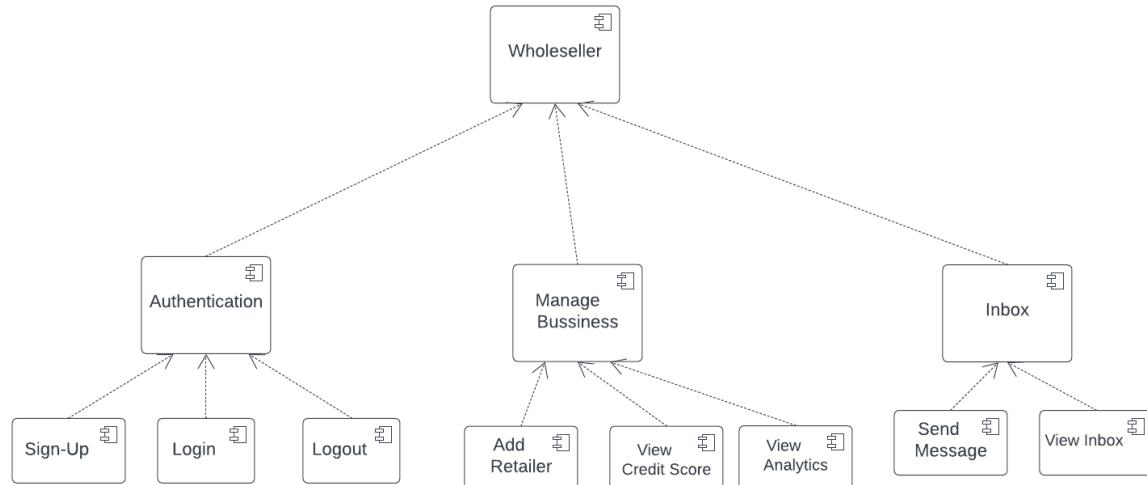


Figure 13: Wholesaler Component
This figure shows the wholesaler's component diagram.

The wholesaler component in our application will be responsible for managing business, sending, and viewing messages and authentication of retailers registered.

5.4 Architectural Strategies

The architecture of the software system was designed with the following strategies in mind:

5.4.1 Front-end Framework

One of the key strategies employed in the design of the system's software is the use of a popular front-end framework for developing the user interface. The chosen framework is React, which

is widely used for building high-performance, reusable UI components. React provides an efficient way of building large-scale applications with a complex data structure that are easy to maintain and scale. The decision to use React was made based on the design goals of creating a modern, responsive and dynamic user interface that can handle complex data and interactions.

5.4.2 Backend Stack

Another key strategy employed in the design of the system's software is the use of a backend stack for handling the business logic and data management. The chosen stack is the MERN stack, which is a popular and widely used stack that includes MongoDB, Express, React, and Node. This stack provides an efficient way of building full-stack applications with a focus on scalability, flexibility, and performance. The decision to use the MERN stack was made based on the design goals of creating a highly scalable, modular, and flexible backend system that can handle a large amount of data and user interactions.

5.4.3 Microservices Architecture

The system's software architecture employs a microservices architecture to handle the business logic and data management. The microservices architecture is a design pattern that decomposes large and complex applications into smaller, more manageable services that can be developed, deployed, and scaled independently. This architecture provides several benefits, including improved scalability, modularity, and flexibility. The decision to use a microservices architecture was made based on the design goals of creating a highly scalable, modular, and flexible system that can handle a large amount of data and user interactions.

5.5 Class Diagram

Following is the class diagram:

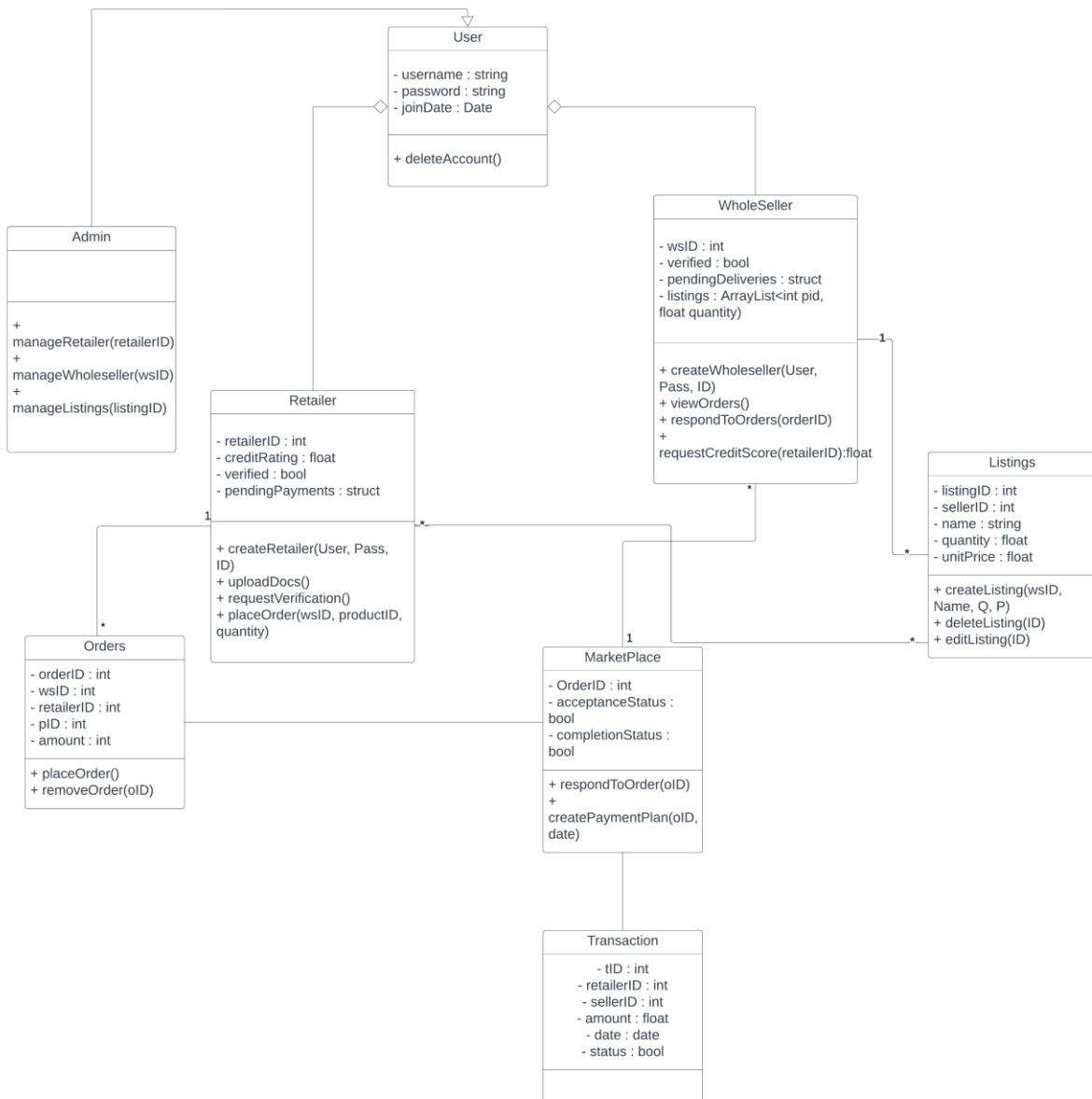


Figure 14: Class Diagram
This diagram depicts the class diagram of our application.

5.6 Sequence Diagram

Following are the sequence diagrams:

5.6.1 Authentication

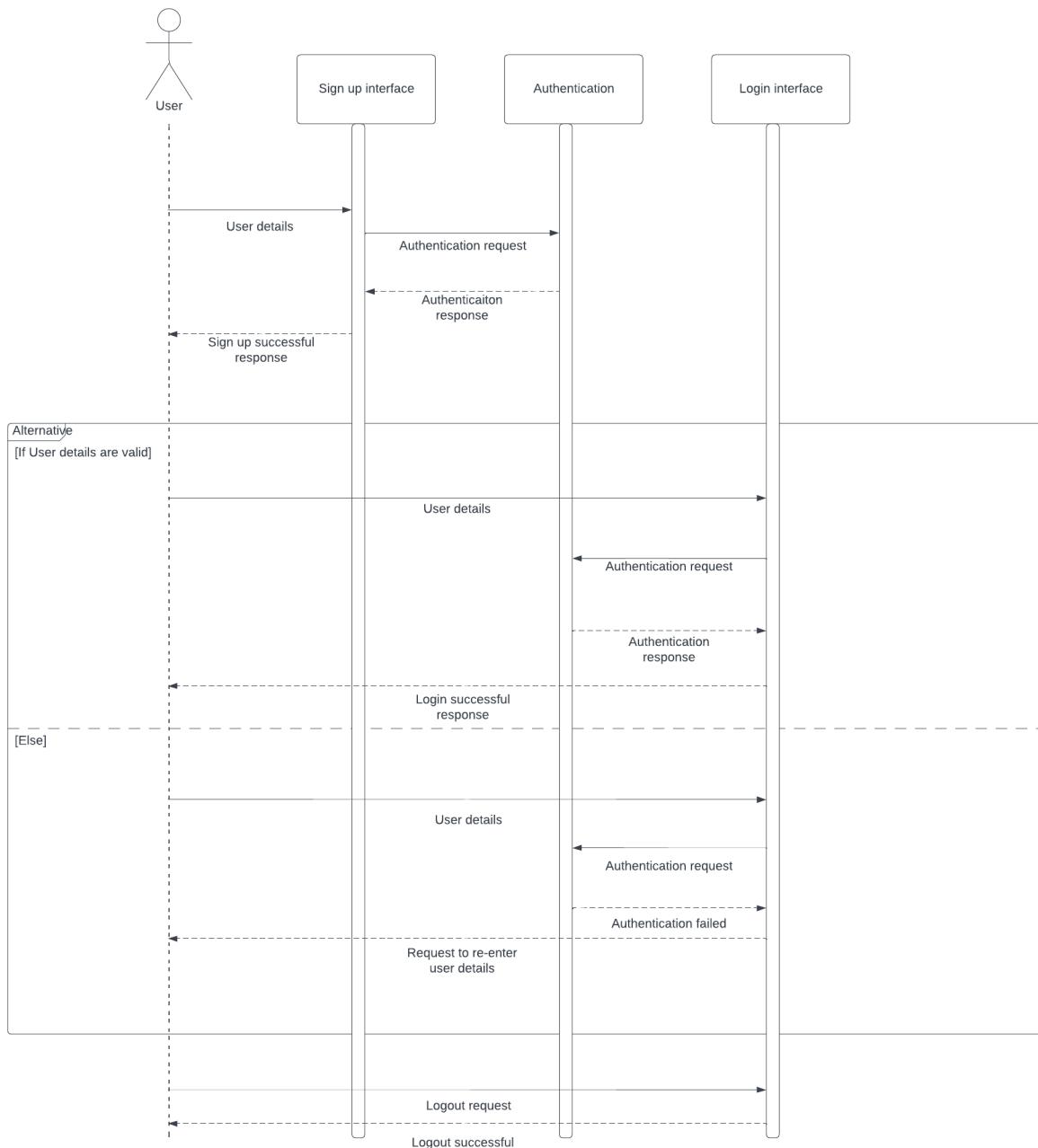


Figure 15: Authentication Sequence Diagrams
This diagram represents the sequence diagram for the authentication of an account.

5.6.2 Searching and Storing Vendor Details

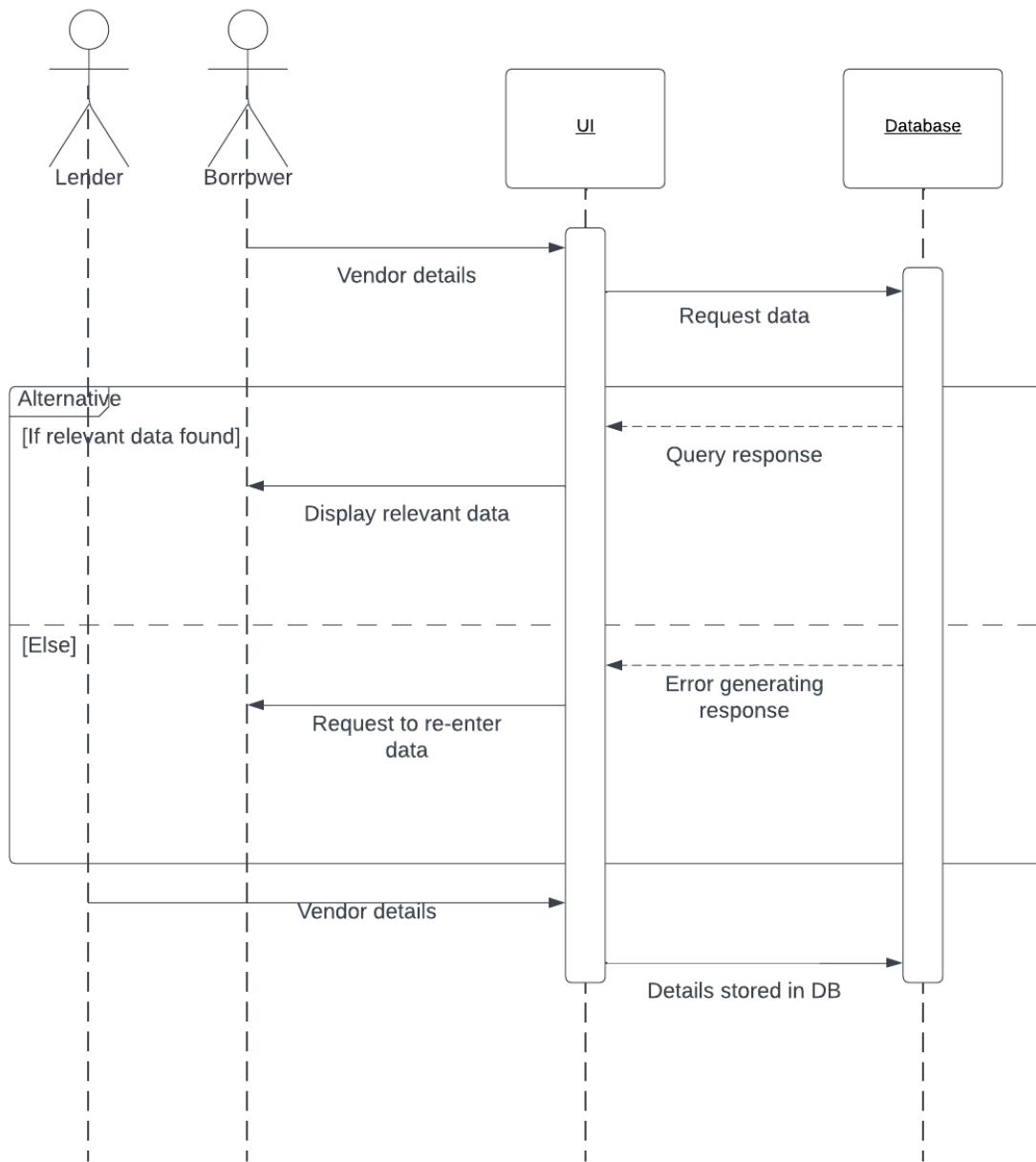


Figure 16: Storing Vendor Details Sequence Diagram
This diagram represents the sequence diagram for storing vendor details of an account.

5.6.3 Storing Lender Document in Database

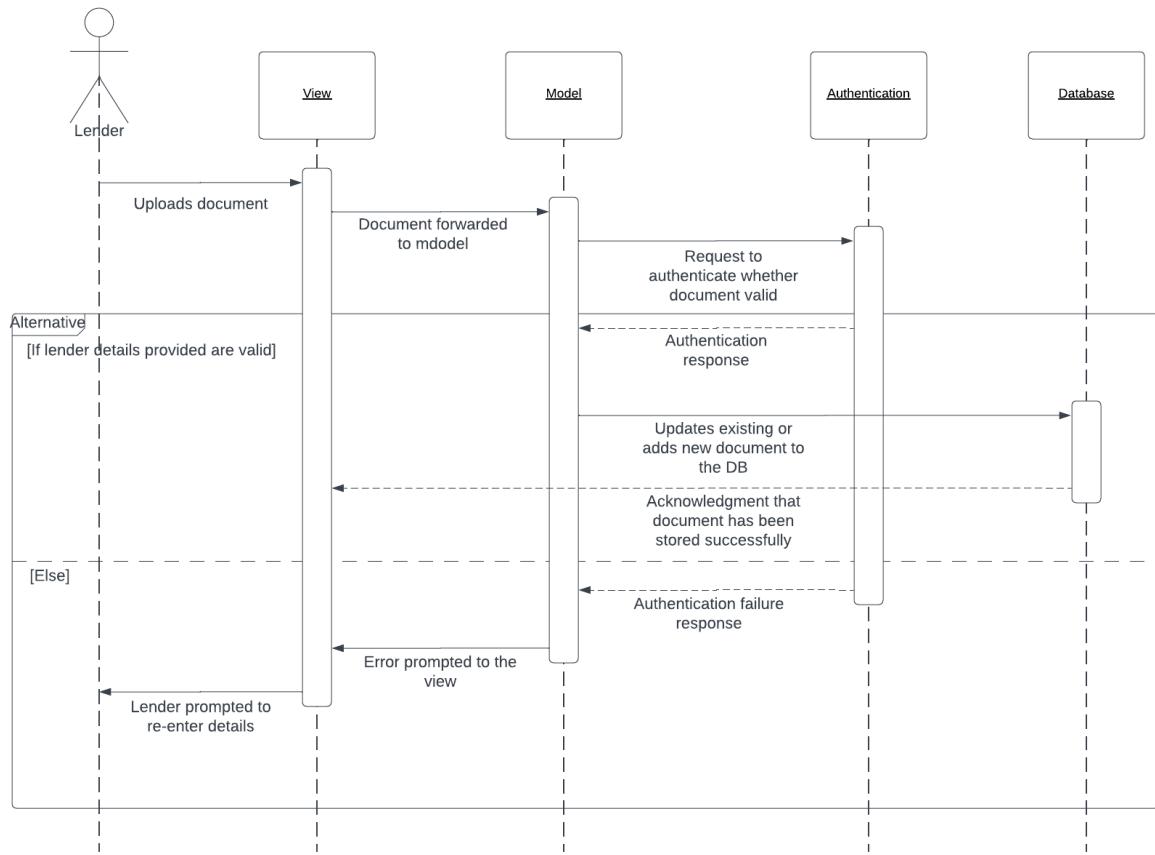


Figure 17: Storing Lender Documents Sequence Diagram
This diagram represents the sequence diagram for storing lender documents in the database.

5.6.4 Retrieving Lender Document from Database

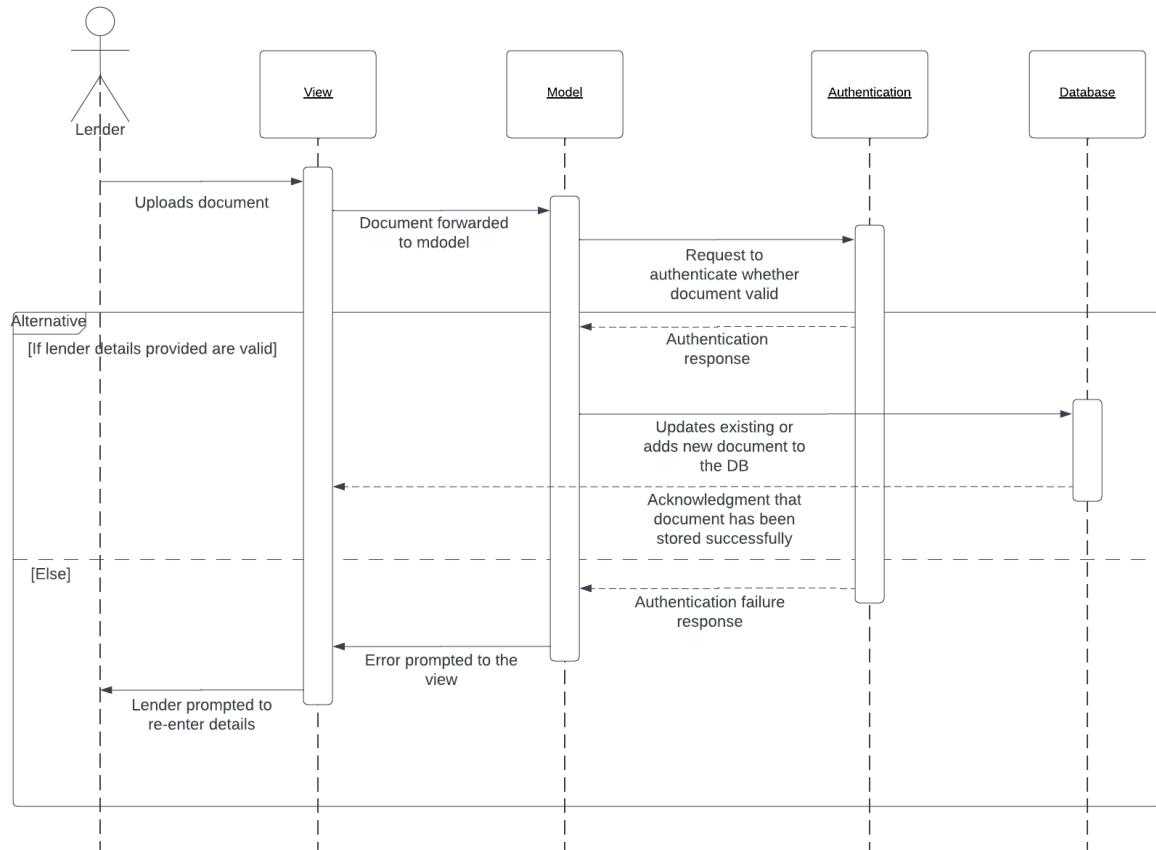


Figure 18: Retrieving Lender Document Sequence Diagram
This diagram represents the sequence diagram for retrieving lender documents.

5.6.5 Generating Credit Score and Establishing Connection

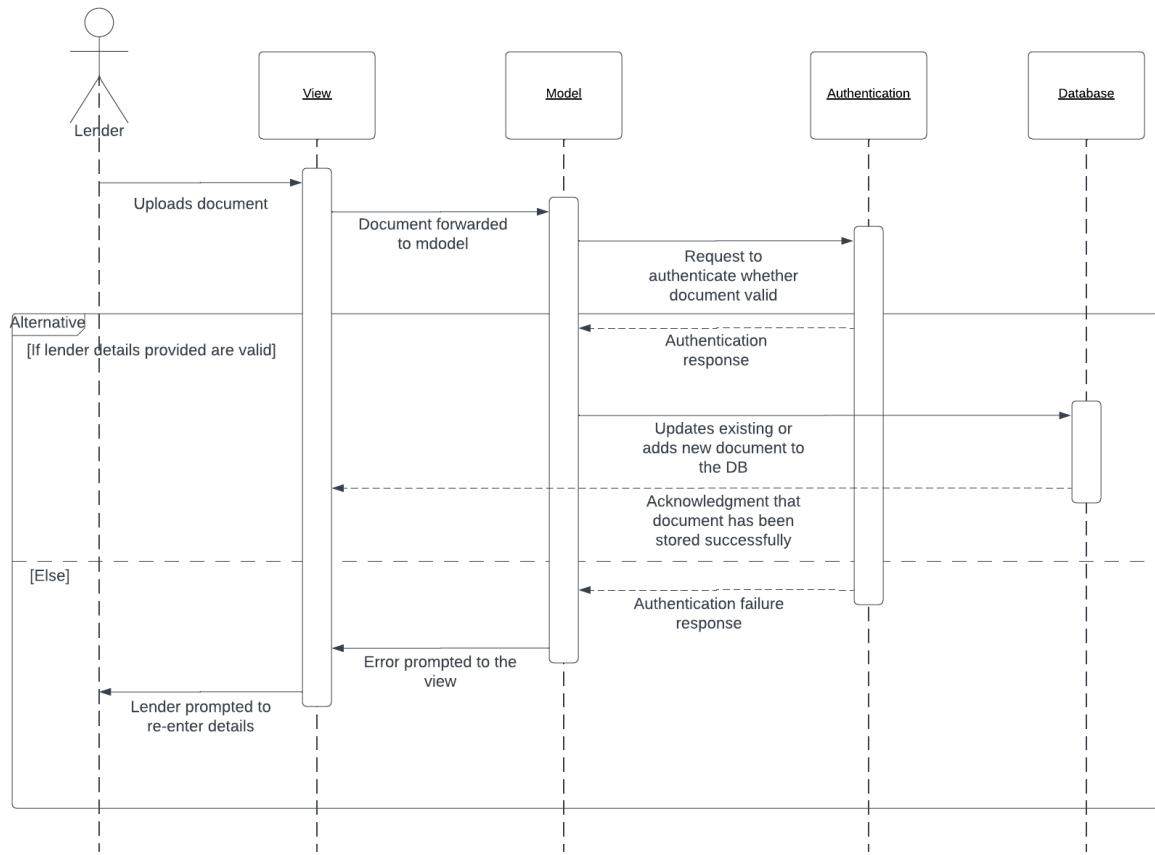


Figure 19: Generating Credit Card Score Sequence Diagram
This figure depicts the generation of credit card score and establishing connections.

5.6.6 Transaction Verification

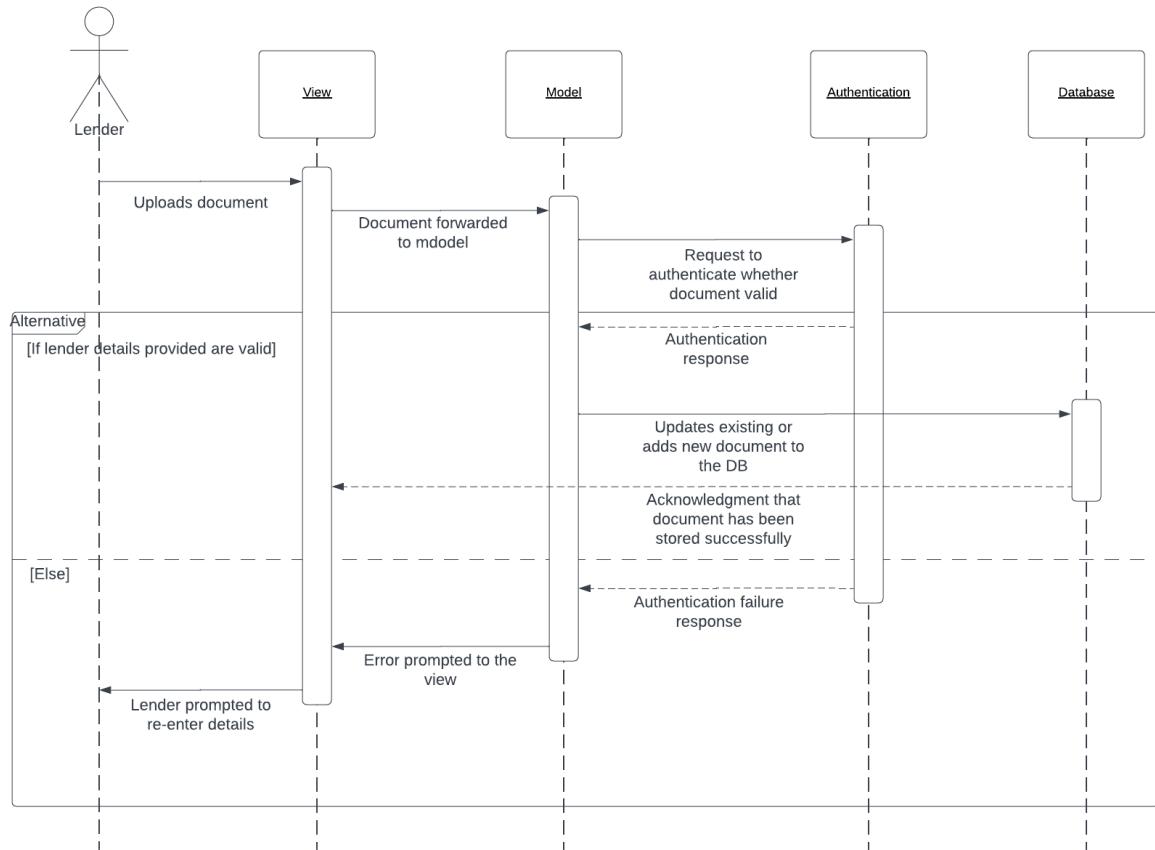


Figure 20: Transaction Verification Sequence Diagram
This figure depicts transaction verification of our application.

5.6.7 Chatbot

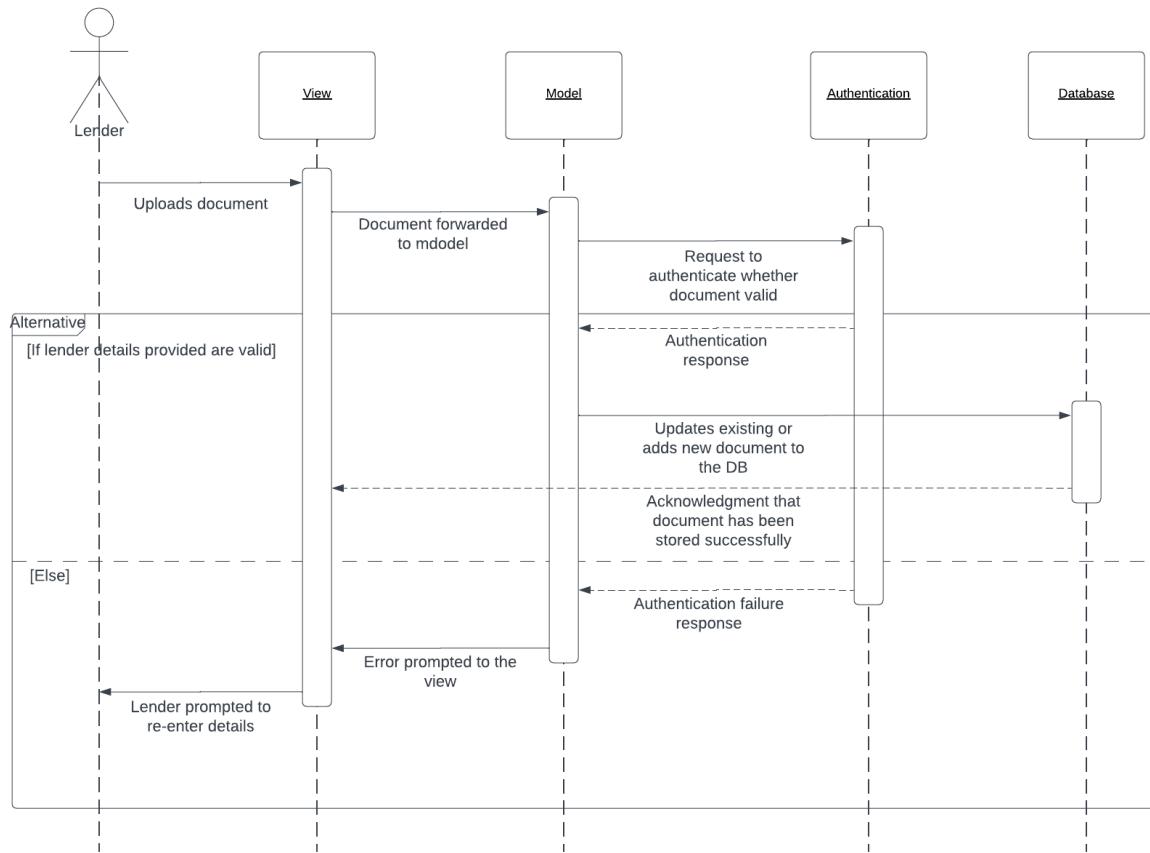


Figure 21: Chatbot Sequence Diagram
This figure depicts the sequence diagram for chatbot of our application.

5.7 Policies and Tactics

The following are the different policies and tactics considered while developing the project:

5.7.1 Coding Guidelines and Conventions

To ensure consistency in the codebase, we will follow a set of coding guidelines and conventions. This includes using namin camelCase for variable and function names, following the indentation and spacing conventions, and using meaningful names for variables and functions.

5.7.2 Testing Framework

To ensure the quality of the code, we will use Jest as our testing framework. Jest is widely used in the JavaScript community and provides a comprehensive testing solution for both frontend and backend code.

5.7.3 Git Branching Strategy

To ensure the codebase is organized and changes are managed effectively, we will follow a Git branching strategy. We will use the Gitflow branching strategy, which involves creating separate branches for feature development, bug fixes, and releases.

5.7.4 Dependency Management

To ensure that dependencies are managed effectively, we will use Yarn as our package manager. Yarn provides a fast, reliable, and secure way to manage project dependencies.

5.7.5 Error Handling

To ensure that errors are handled effectively, we will follow a structured approach to error handling. This includes creating custom error classes for different types of errors, logging errors with relevant information, and handling errors gracefully.

Chapter 6: Implementation And Test Cases

6.1 Implementation

This chapter explains the implementation that we have done so far. We have developed a prototype that will demonstrate what we expect our product to look like from the consumer point of view.

6.1.1 Prototype Description

For the initial prototype, we started implementation of the front end of our project first. Our reasons for making this choice were that not only will the front end help us demonstrate what we want our finished product to look like, it will also help us materialize our expectations and set a clear vision for what needs to happen at the back-end for the program to work cohesively.

We are using the MERN stack to implement our front-end. It is the latest framework for web front-end development that allows us to have an in-depth control of our design features while also providing us the necessary tools to ensure the utility and efficiency of our interface. We also used HTML5 and some CSS libraries to achieve the desired outlook for our front-end.

For demonstration, we have designed three pages, namely “Log-in Page”, “Sign-up Page”, and “Retailer-Whole Seller Interface Page”. These pages will hold most of the information that will be relevant to our users and thus we deemed them to be adequate for prototyping purposes. The background business logic will be abstracted and thus it can be designed and coded later.

6.1.2 Sign Up Page

The sign-up page will be the destination for our first-time users, where they will be able to register an account for our app. The Page has two embedded tabs that allow users to switch between “Retailer” and “Whole-Seller” depending on who they are signing up as. One account can only be of one type. The user is asked to enter the necessary information (which varies depending on whether they’re a retailer or a seller) and after authentication of the information, the account will be registered.

6.1.3 Login Page

The Log In page will allow users that have already registered for an account with the app. For security, Users will also need to fill in a Captcha code. In case of multiple failed logins, the account will be locked and will require being unlocked by going to a link sent via email. For added security, we will implement end to end encryption to protect the credentials of our users. There will also be links users can click on if they have forgotten their password, or if they are having other problems that they need to contact us for.

6.1.4 Retailer – Whole-Seller Interface

This page will hold tabs that can be switched between. For Retailers, they will have a tab where they can add or update the details of their business by entering information and uploading documents. They will also have tabs that show them their outstanding orders, their scheduled/pending transactions, and the history of their dealings through our app. For Whole-Sellers, the page will have individual tabs where they will be able to see and perform CRUD operations on their listings, see and respond to the orders that they have received from retailers, request analytics for any order placed by a retailer, view their received and pending payment timeline, and also view their order completion history.

Chapter 7: Conclusion And Future Work

To make the front-end functional, we will need to implement a powerful back-end program in our FYP-II. We will need to establish a way to standardize the information from retailers that we will be using for generating analytics. This will include selecting which sources of information we will be using, and what sort of a metric we will establish to evaluate their credit-readiness. We will then need to create a containment model for storing this data in the Database in a way that it's easy to process and work with. We will also need to train our AI Model with a sample data of the market and calibrate it properly for credit predictions. Lastly, we need to work on our blockchain integration on the side, to incorporate the in-app payment planning and transaction history features in our app.

References

- [1] B. H. Misheva, J. Osterrieder, A. Hirsa, O. Kulkarni, and S. F. Lin, “Explainable AI in Credit Risk Management,” *arXiv.org*, 01-Mar-2021. [Online]. Available: <https://arxiv.org/abs/2103.00949>. [Accessed: 09-Mar-2023].
- [2] J. G. Mooney, P. Rosati, and M. Cummins, “Disrupting finance: Fintech and strategy in the 21st Century,” *OAPEN Home*, 01-Apr-2020. [Online]. Available: <http://library.oapen.org/handle/20.500.12657/23126>. [Accessed: 09-Mar-2023].
- [3] V. Pacelli and M. Azzolini, “An artificial neural network approach for credit risk management,” *Journal of Intelligent Learning Systems and Applications*, 08-Apr-2011. [Online]. Available: <https://www.scirp.org/html/4587.html>. [Accessed: 10-Mar-2023].