FYP Final Report

Decentralized Credit Score System

Final Year Project Report

By

Arslan Ali

Muhammad Muzammil Raza

In Partial Fulfillment

Of the Requirements for the degree

Bachelors of Science (CS/SE)

Sukkur IBA University,

Sukkur, Sindh Pakistan

(2021)

Decentralized Credit Score System

By

Arslan Ali

Muhammad Muzammil Raza

SUBMITTED TO THE DEPARTMENT OF
COMPUTER SCIENCE IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

**BACHELOR OF SCIENCE IN COMPUTERR
SCIENCE / SOFTWARE ENGINEERING**

at the

SUKKUR IBA University

Month, 2021

Signature of Author(s)

Arslan Ali

Muhammad Muzammil Raza

Certified by: Internal Examiner

<Name>,<Designation>, <Department>

External Examiner

<Name>,<Designation>, <Department & Organization>

Accepted by:

Dr. Ahmad Waqas,

Head Department of Computer Science

DECLARATION

We hereby declare that this project report entitled “Decentralized Credit Score System” submitted to the “Department of Computer Science”, is a record of an original work done by us under the guidance of Supervisor “Dr. Asif Raza Shah” and that no part has been plagiarized without citations. In addition, this project work is submitted in the partial fulfillment of the requirements for the degree of Bachelor of Computer Science.

Team Members	**Signature**
Arslan Ali	______________
Muhammad Muzammil Raza	______________

Supervisor:	**Signature**
Dr. Asif Raza Shah	______________

Date: ______________

Place: Sukkur IBA University

DEDICATION

We are greatly thankful to our parents, teachers, fellow students and beloved friends for helping us through this entire journey. Without teachers' love, prayers, support, proper guideline, direction, and without the support of fellow students and friends, this could have been never ever possible.

We dedicate my all work to my supervisors **Dr. Asif Raza Shah** who always appreciate and properly guide us to achieve the task and complete the project. We also thankful and dedicate our work to Mr. Touqeer Shah who help us and give support to complete the project.

In last we dedicate this work and give special thanks to the all the negativity for always being there with us throughout the entire bachelor program, which always encourage us to fight back and come out from our comfort zones. You made us the fighter in your every form either in a form of situations, thoughts or a human.

ACKNOWLEDGEMENTS

Our heartfelt thanks go to ALLAH for having given all the necessary knowledge to plan and execute this project and the curriculum. There was never a scarcity or a need. During all this research, he took control of all that might have hindered us and helped us even in our hardest moments.

we respect Professor Nisar Ahmed Siddiqui, the Vice Chancellor of the great institution, because he has pursued God 's guidance and creation and nurtured the vision of Sukkur IBA University, a great institution where we were educated to take up our destiny as a royalist and dominator. We can't ignore Computer Science Department Head Dr. Ahmed Waqas, a real gentlemen and a young leaders' mentor and the person who listens and act according to betterment for students. Never the less our all teachers who been there to motivate and guide us to achieve our goal.

Contents

LIST OF FIGURES

ABSTRACT

In Pakistan only 49 million people are using banks' services, and total 150 million people are unbanked meaning 75% of total population is fully deprived of banking services in Pakistan. Lending service can play an important role to lift the country's economy and only small portion of the banked population is labeled as eligible for lending. Unfortunately, credit penetration in Pakistan is very low and only $4400 million private loans has been reported till July 2019 which is not a good figure to involve individual citizens in the economic cycle of country. We felt this chronic problem is only because of trust deficit and lack of transparent information between financial institutions and borrowers. The traditional way to evaluate the creditworthiness of an individual borrower is to make a prediction over one's financial picture which involves income, expenditure, and credit history. In Pakistan, each bank is managing credit history of individual borrowers independently and they are not sharing the credit information with other banks because of competitive advantages. They are sharing the credit information of borrowers only with Govt. licensed credit bureau institutions to calculate the credit score. Credit scoring institutions in Pakistan (such as eCIB, DataCheck, AISL, etc.) are mostly centralized third-party systems that rely on obsolete techniques and technologies

We are going to propose a model called DCScore (decentralized credit score) system that will be composed of a decentralized identity and credit scoring system and it will be implemented by using a decentralized credit platform as a proof of concept. Our system will provide the solution of all above-mentioned issues. We are going to develop a fully decentralize credit score calculating system by using permissioned Blockchain technology.

Our system will eliminate the dependency of centralized institutions (credit bureau) for credit score calculation, and it provide the secure and transparent solution to every financial institution with the feature of privacy. All lenders will be able to check the credit scores of individuals, and the complete information of individuals will be secret expect credit score. The purpose of credit score is to check the financial condition of individual that he or she either is eligible for lending or not.

Chapter 1

INTRODUCTION

1.1 Introduction

“Credit Score” is consider very important factor for borrower while requesting for loan from any financial organization. Credit Score is numeric score, which describes the credit-worthiness of individual. It also represents that borrower is risky or beneficial for that financial Institute. Our system “Decentralized credit score system” is web-based application. The main motive of our system is to calculate the credit score to evaluate trustworthiness of an individual in term of returning the credit. The whole system is based on DLT (Distributed Ledger Technology) using Blockchain. There will be only two organizations, which are allow to participate in our system. Both organization can add their client record, calculate the credit score, and also search the credit score on any client. Another other most important purpose of our system is to share the credit score of borrower to all organizations which are participated.

1.2 Project Objectives

In Pakistan 75% of population are not using bank services. If we talk about the lending service, the ratio of using lending services is very low. As the research said that lending services play the big role to lift up the country’s economy. The question arises that why people are not taking benefit from the lending services? There are certain factors like trust issues between customer and Banks. Banks also do not believe the information, which they are taking from credit bureau institutes for checking the credit worthiness of his/her clients. The biggest factor of not sharing the client information is that one Bank does not share the information of his/her client to other Bank because of competitive reason. They only share information to government license credit bureau. Currently they are relying on centralized system, so the monopoly can occur while calculating the credit score.

So by analysis on this problem we extract out the following objective

- Create trust between customer and Banks
- Increase the lending services
- Banks become independent
- Eliminate third party
- Build transparency of data in the system and it will show correct credit score of customer based on his/her payment history or valid information.
- Feature in the system where one Bank can see other Bank customer's credit score with its permission.
- Technology where we can store the data in distributed ledger

1.3 Project Scope

- In this prototype, we will use two banks.
- Each Bank can store their customer information and calculate credit score in decentralized system.
- Bank A cannot view the credit history of Bank B's client and similarly Bank B cannot view the credit history of Bank A client respectively.
- The only one thing they can share with each other that is general information (name, CNIC,) and "Credit Score" of customer.

1.4 Not in scope/Limitation

- No more than two banks can participate right now.
- Bank cannot directly communicate with other banks.
- Both bank are not setup at individual system

Chapter 2

LITERATURE REVIEW

In this chapter, we would start with background of project, problem statement and scope of the overall project namely “Decentralized Credit Score System”. In past 50 years we are making application by using centralized system. In centralized system we store all the data on same hardware at single system.

For example, we have centralized database which has thousands of records and multiple users can access the database. So there will be have some issues like we do not know who made changes in database? Who removed or retrieve the records from the database or when?

Some data are very critical like banks customer’s data if someone hacked this type of data so it will gives big lose to the banks. So we also need the databased security plus databased status and it will create complexity and given us low performance. So we need technology who provide the solution all above problems. Blockchain is the famous technology which provide transparency, immutability and also show the status of data.

2.1 Equifax Data Breach

Equifax, one of the three largest consumer credit reporting agencies in the United States, announced in September 2017 that its systems had been breached and the sensitive personal data of 148 million Americans had been compromised [4].

EQUIFAX

The data breached included names, home addresses, phone numbers, dates of birth, social security numbers, and driver’s license numbers. The credit card numbers of approximately 209,000 consumers were also breached. The Equifax breach is unprecedented in scope and severity. There have been larger security breaches by other companies in the past, but the sensitivity of the personal information held by Equifax and the scale of the problem makes this breach unprecedented [4].

Now credit agencies are trying to convince people to used lending services and motivate the people who are still not using bank services or unbaked. Like Bloom, MicroMoney and Pave are using blockcahin and applying different strategies to motivate people for using lending services.

2.2 Bloom

Blockchain startup Bloom wants to change that by giving consumers power over their identity and data. The Bloom mobile application, which allows consumers create and build a blockchain-based identity profile, is part of how the startup wants to improve consumer data security.

The company has now announced the launch of a second major version of the Bloom mobile app. With the new version, users now have access to what the company calls a "robust credit monitoring" feature including free credit score checks, which is possible thanks to a new partnership with American credit reporting agency TransUnion[5].

TransUnion is one of the three largest credit bureaus in the United States, along with Experian and Equifax.

Bloom provide end-to-end protocol, which focus on risk assessment and credit scoring. Its aim is that to provide complete ecosystem to unbanked people so they can access credit services [5].

The BloomID is central to this system, allowing users to establish a "global federated identity with independent third parties who publicly vouch for their identity information and legal status." The Bloom network relies on established nodes to verify user identity information. All pre-ordained nodes are open and equally verified. Therefore, while a user could apply for credit using a set of fake information, it is extremely unlikely that it would receive confirmation [5].

2.3 MicroMoney

MicroMoney offers a solution to helping people build their first credit history on the blockchain. It uses new technologies and Big Data to determine a person's creditworthiness via an innovative neural scoring credit [6].

MicroMoney differs from Bloom and its main goal is to connect new customers to existing financial services. New customers can connect to existing businesses, or start their own using access to new lines of credit [6].

2.4 Pave

Pave is an existing alternative credit agency that focus on underwriting and lending to those with limited credit history. Pave has provided funds to over 1,600 individuals with limited credit history, with a focus on younger borrowers and immigrants.

Using blockchain and related technologies, Pave’s GCP (Global Credit Profile) will decentralize the storage and ownership of an individual’s financial data by placing the user in control. The GCP thereby removes the reliance on a singular record keeper making security breaches infinitely less likely [7].

With the GCP each individual's financial profile will become a personal asset, portable on a global basis, verifiable and secure. GCP will empower and incentivize people to enrich their profiles with a range of significant and trustworthy financial data that is currently ignored by most traditional credit bureaus [7].

While the centralized systems of companies such as Experian, Equifax and TransUnion continue to perform a valuable service by acting as a reliable source of information for third parties, they are plagued with systemic problems including a lack of transparency and control over personal data, vulnerability to fraud and data theft and unnecessary administrative costs. Using blockchain and related technologies [7].

Secure Lending (Blockchain and prospect theory based decentralized credit score) we suggest how blockchain can provide the answer to the assessment of decentralized credit scoring and reduce the amount of paperwork dependency. To any lender, lending money is not always factual, but subjective. Depending on their viewpoint, the lending decision entails varying degrees of risk and uncertainty. In order to model the best investing strategy for various risk vs. return conditions, this paper uses the prospect principle [8].

Credit Scoring (using the hybrid neural discriminant technique) Most experiments have focused on developing an accurate model of credit scoring to determine whether or not to give credit to new applicants, and it seems that attempts to build a more accurate model of credit scoring are not significant [9].

We also proposed a system name as "Decentralized Credit Score System". In this system we are using blockchain to developed this system. The main functionality of this system is that it is decentralized system where each bank can store their customer's private information in a distributed way and there is no chance for breach or hacked the data. It's also calculate the credit score automatically by taking different parameter from customer and calculate the credit score. One bank can only see the credit score of other Bank's customer. So it will help to detect the defaulter who is applying for loan in other bank.

The scope of the project is that we create two banks and each bank will have two users like user1, user2 from Bank A and Bank B. Each Bank can store their customer information and calculate credit score in decentralized system. Where Bank A cannot view the information of Bank B and respectively Bank B cannot view the information of Bank A. The only one thing they can share with each other, which is "Credit Score" of customer.

Chapter 3

PROBLEM DEFINITION

Suppose a person is requesting for loan/credit card to financial institute. Then how a bank validates the person's will in term of returning the amount. It might be, already the person is availing the loan facility in another bank and failed to return the credit. In Pakistan, banks/financial institutions are not willing to share the sensitive data with another banks because of security and nature of data. In this case how an individual bank will measure the creditworthiness of person who is requesting. By resolving this problem, the third party is managing the individual's credit history from its financial institute members which is known as credit bureau institutions. Currently in Pakistan credit bureau institutions are (such as eCIB, DataCheck, AISL, etc.) calculate the "credit score" on behalf of individual's credit history, income, expenditure. Financial institute are sharing the credit information of borrowers only with Govt. licensed credit bureau institutions to calculate the credit score.it means the credit bureau institution is only one way to calculate the individual's credit score.

Credit bureau institution stores the sensitive data in centralized system which leads to security issue and every financial institute is dependent on it. Centralized Systems that rely on obsolete techniques. What if, the credit bureau provides the inaccurate information. How we can validate, because the data is not transparent. Having in very few in numbers, they are playing monopoly role. Therefore, the existing system does not meet with the level of user's satisfaction. We are going to develop a decentralize credit score calculating system, that will solve the above-mentioned problems.

Chapter 4

METHODOLOGY

4.1. Overview

In this chapter, we will discuss the design, workflow and methodology relevant to the 'Decentralized Credit Score System'. It investigates the approaches, procedures, concepts, theories and techniques relevant to this project. Some of the questions briefly explained in this chapter are:

1. What methods are used while development of DCScore System?
2. What inspection techniques are applied when system has been developed?
3. What tools and techniques are used, in designing and development of the system?

4.2. System Design Phases

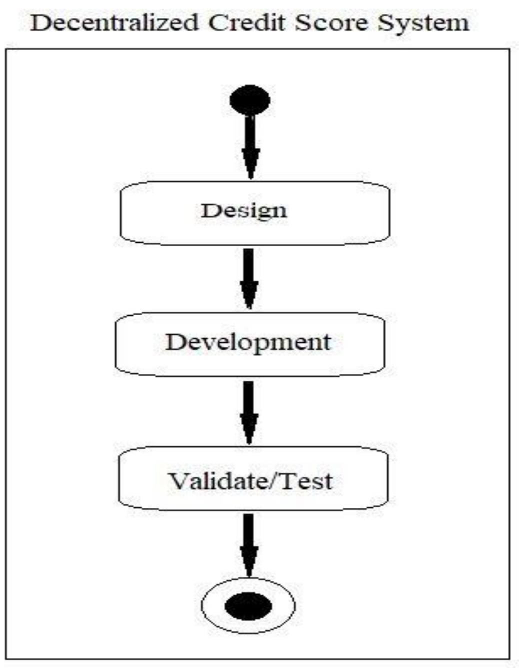


Figure 1 Decentralized Credit Score System

4.3. Agile Methodology

Agile is the ability to create and respond to change. It is an iterative approach and a way of dealing and succeeding in an uncertain environment. Agile methodology for the development of this system, Decentralized Credit Score System, is picked because we needed the rapid development of this project in the limited given time. That is why this methodology deemed suitable, feasible and greatly served the purpose. While followings are some of the main features of this methodology, which best fits as per our requirements to build DCScore System:

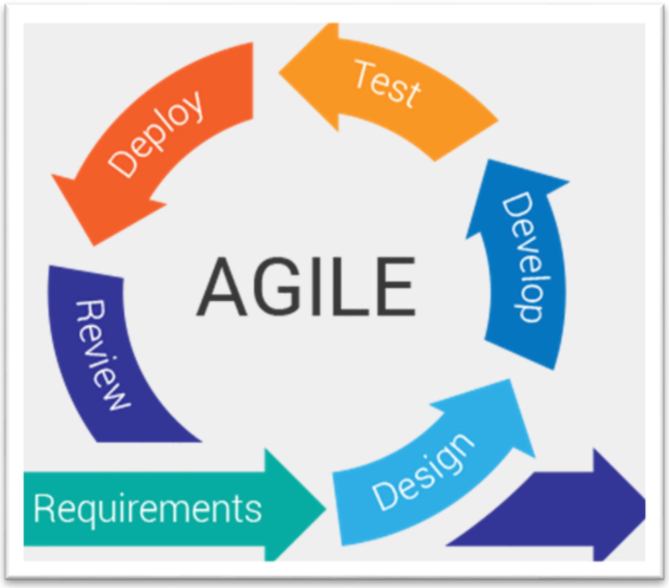


Figure 2   Agile Methodology

- Incremental Delivery (as this software was to be developed with multiple iterations and having green signal on current iteration is utmost necessity to move onto the next iteration)
- Maintain Simplicity
- Development according to user easiness

4.4. Design

We have designed our project using StarUML which is a highly sophisticated software modeler aimed to support agile and concise development which is what we needed. We have made the use of domain model, use cases, and sequence diagram because of their significance in making our path clear in the development phases. These type of diagrams are widely used and help model requirements accurately. However, main activities of this project are described in Chapter 5.

4.4.1. Techniques

- Use Case Model
- Sequence Diagram
- Domain Model

4.4.1.1 Procedure

The procedure starts with the design of the system initially by sketching on the paper. While for physical and abstract design, we used StartUML software which is great tool in this regard. Furthermore, we used Hyperledger Fabric framework along with curl, Docker engine, Docker compose, Go, Node JS SDK, Composer CLI and python for the development of the system, DCScore.

4.5. Development

The development of DCScore System contains the following screens and is designed using the mentioned language/editor.

4.5.1. Tools/Languages and Editor

- Star UML for modeling
- Hyperledger Fabric Framework, curl, Docker engine, Docker composer, Go, Node JS SDK, Composer CLI and python
- However, Atom and VS code editors are used.

4.6. Inspection

Inspection is minutely on every given functional requirement. With testing and inspection, we become able to know if the system is performing as per the requirements and generates desirable output at specified input. Wherefore, some of the important techniques are as following:

4.6.1. Manual inspection

Manual inspection is performed against each functional requirement. We are able to meet every functional requirement at satisfactory level which is dually checked and verified by the respective supervisor, and other core users.

Chapter 5

DETAILED DESIGN AND ARCHITECTURE

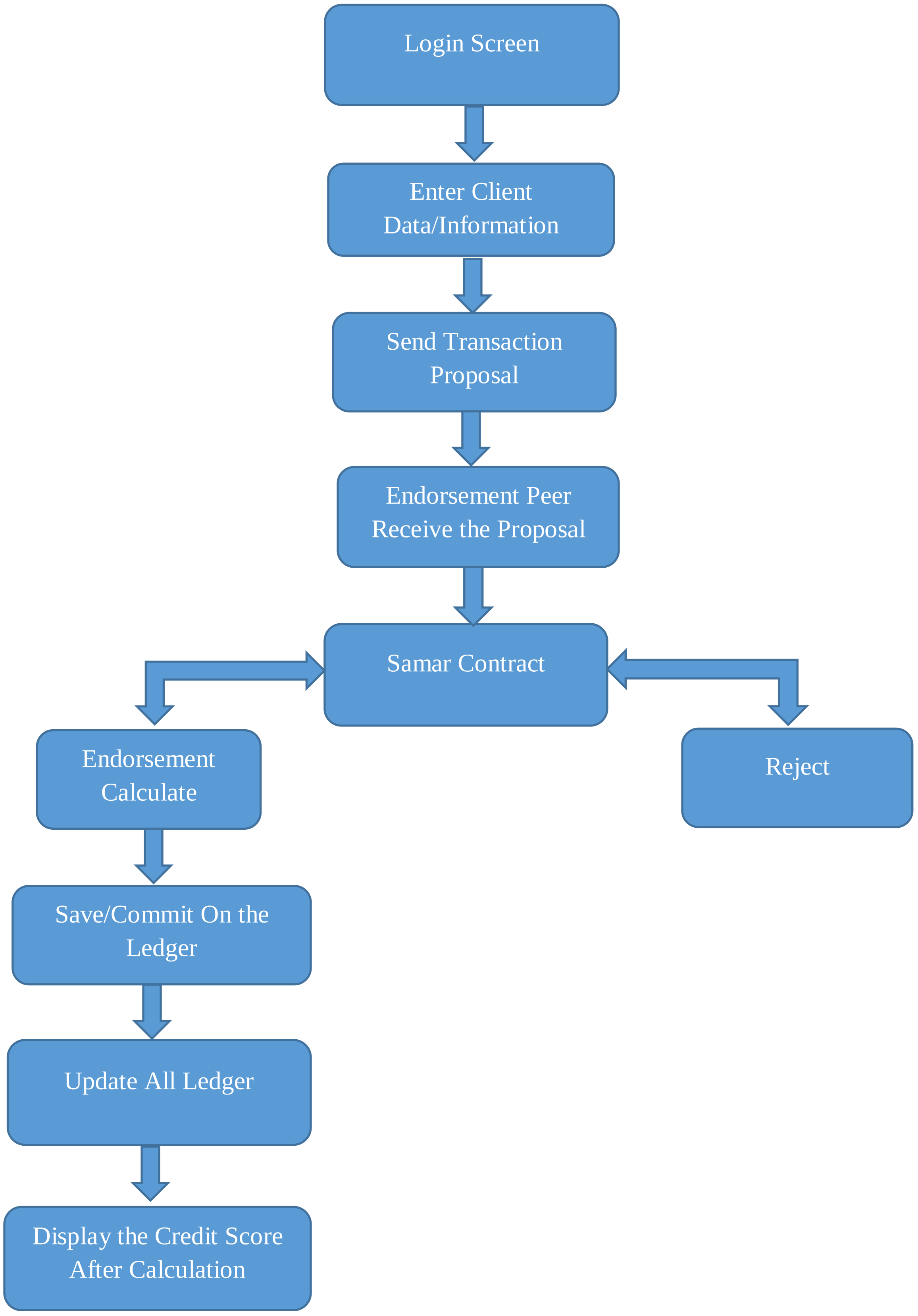
5.1 System architecture

Decentralized credit score system is based own blockchain technology. In this technology, we will store the data in distributed form and used key-value pair structure to get or set the data. The main components of this technology is smart contract, consensus algorithm and nodes. In smart contract we define the rule and the business logic and it is also process input data and provide based on defining rules and business logic in the smart contract. The transparency of data sustains by consensus algorithm. Node are any type of computer (like laptop, servers etc.) which form the infrastructure of blockchain. All nodes are connected to each other and they constantly exchange the latest blockchain data with each other so all nodes stay up to date. It is the brief overview about of blockchain technology.

Now I will tell you why I choose this technology in over decentralized credit score system. Because It is banking side application and it will store the sensitive data of their customers. So maintain the transparency and existence of sensitive data we used blockchain technology.

Now, I will like to tell you about brief introduction of our systems architecture and its component. The first step is user identification and authentication. So we making login screen which check user credentials and allow him to use our decentralized application. Now user enter their client information and this information will be store in ledger in the form of key value pair. After saving the client information the system will create the transaction proposal and send it to the blockchain network. The endorsement peer in the blockchain network receive the transaction proposal and send it to the smart contract. Smart contract will reject or accept the transactional proposal based on defining rules and business logic in the smart contract. After accepting the transaction proposal, the smart contract endorsing the node and calculate the hash. Now system will save and commit the transaction on the ledger. After save and commit, system will update all the ledger and display the credit score after the calculation.

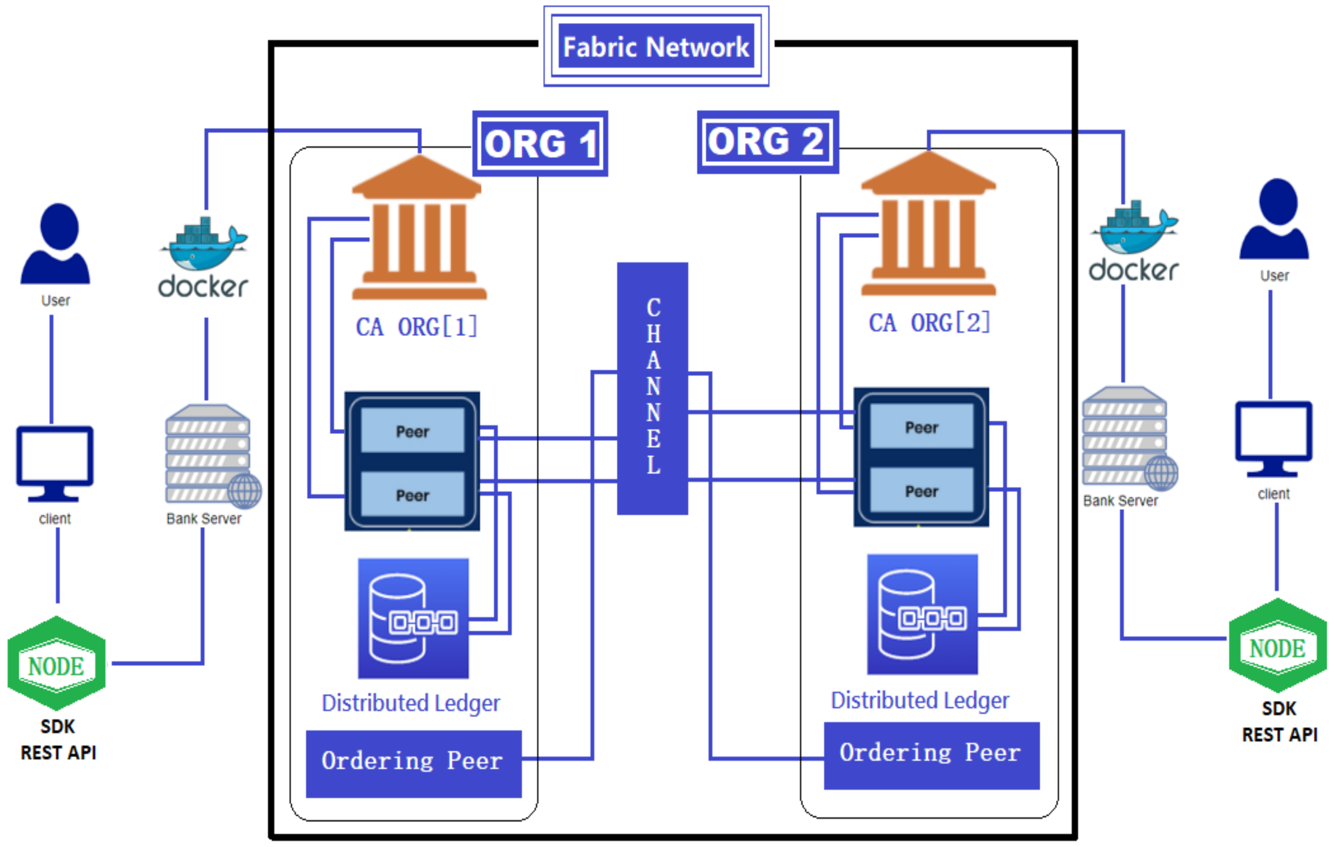
Diagram: System Architecture of Decentralized Credit Score Systems


Login Screen
Enter Client Data/Information
Send Transaction Proposal
Endorsement Peer Receive the Proposal
Samar Contract
Endorsement Calculate
Reject
Save/Commit On the Ledger
Update All Ledger
Display the Credit Score After Calculation

5.1.1 Architecture Design Approach

The design approach, which we use in our system based on client-server architecture. Client-Server Architecture is an architecture in which much of the tools and facilities to be utilized by application are stored, distributed and handled on the cloud. The design of this sort is linked by a network or Internet by one or more client computers to a system.

5.1.2 Architecture Design



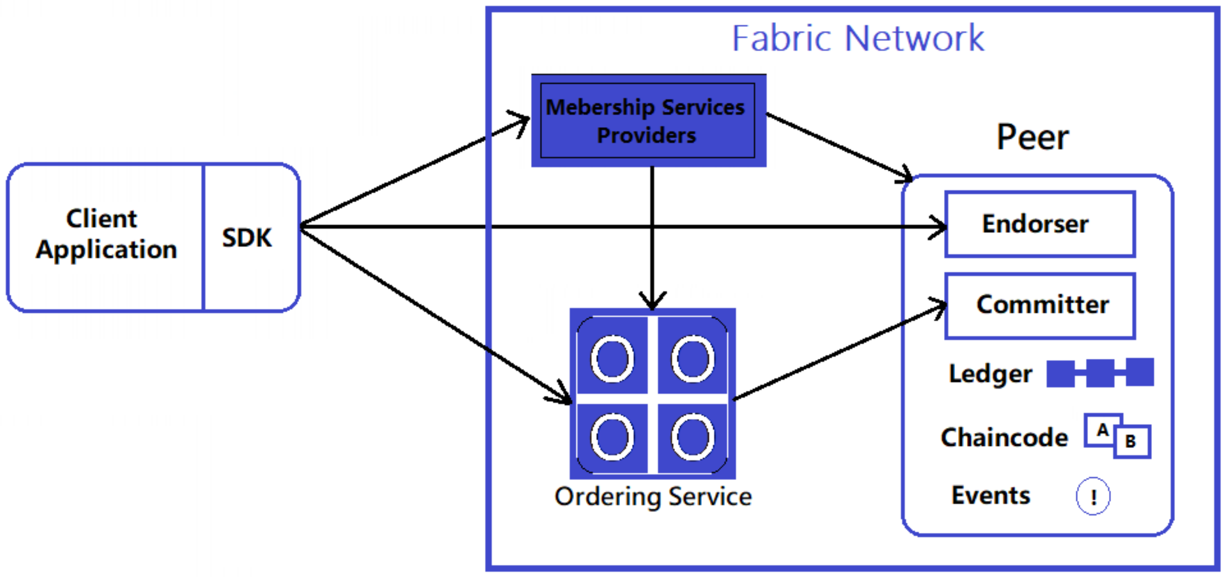
This is the overall architecture design of the project. It consists of main three parts along with multiple subparts which are working together to give desired output.

We can see that, Organization one and organization two are connected to the fabric network. They are communicating to each other by using channel which is the part of fabric network. The design approach we used is not much complex. The authenticate user of an organization use the application and the Node SDK and Rest API connect to the web server of an organization. The web server connects to the docker and the docker also further connected to the fabric network. The fabric network also has some components like certificate authority (CA), peer, ordering peer, chaincode, ledger and the channel. The certificate authority assign certificate to the peer node.

Peers are a fundamental element of the network because they host ledgers and smart contracts. Recall that a ledger immutably records all the transactions generated by smart contracts. Smart contracts and ledgers are used to encapsulate the shared processes and shared information in a network, respectively.

Because a peer is a host for ledgers and chaincodes, applications and administrators must interact with a peer if they want to access these resources. That's why peers are considered the most fundamental building blocks of a Fabric network. Application clients submit transactions containing endorsed transaction proposal responses to an ordering service node. The ordering service creates blocks of transactions which will ultimately be distributed to all peers on the channel for final validation and commit. After the final validation and commit the user can see the credit score.

5.1.3 Subsystem Architecture



It is a simple subsystem architecture, which tell us how the client application interacts with fabric network and how the transaction will process, and update on a ledger.

Committing Peer, which supports the local version of the registry, records transactions and updates data in the registry. In this node, you can optionally install smart contracts.

Endorsing Peer, which is involved in deciding on the execution of a transaction. In this node, smart contracts are required.

Ordering Nodes, which form transaction blocks for adding them to the registry. They provide communication channels between other peers in the network.

5.1.4 Membership Service Provider (MSP)

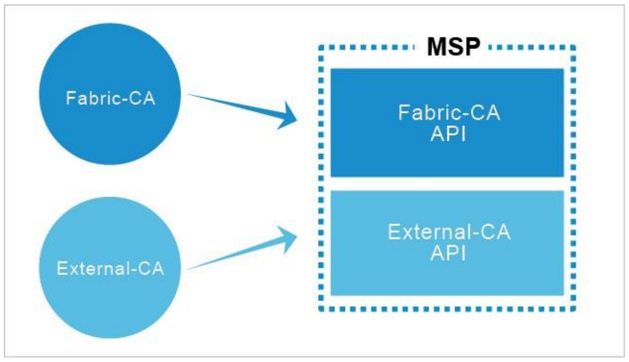


Figure 3 Membership Service Provide

An MSP mange a set of identities within a distributed fabric network.

It provides identity for the following components of fabric network:

- Peers and orderers
- Client application
- Administrator

Identity can be issued by Fabric CA or External CA.

The MSP provides authentication, validation, signing and issuance.

It also supports different crypto standers with a pluggable interface.

The fabric network can include multiple MSPs (typically one MSP for organization). Include with Transport Layer Security (TLS) crypto material for secure communication.

5.1.5 Transaction endorsement

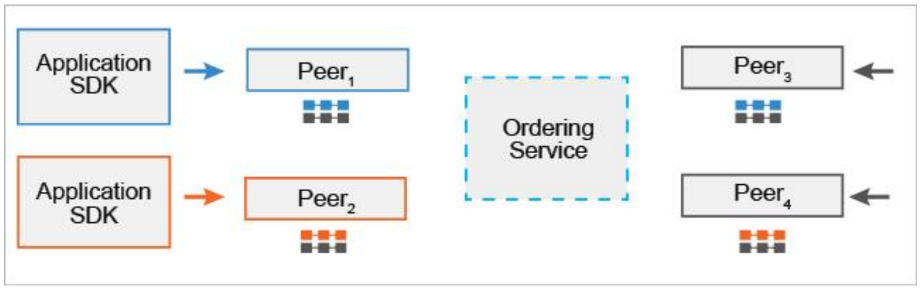


Figure 4 Transaction Emdorsement

A transaction endorsement is a signed response to the results of the simulated transaction. The method of transaction endorsements depends on the endorsement policy, which is specified when the chaincode is deployed. An example of an endorsement policy would be something like, “The majority of the endorsing peers must endorse the transaction.” Since an endorsement policy is specified for a specific chaincode, different channels can have different endorsement policies.

5.1.6 Chaincode (Smart Contarct)

Smart contracts are computer programs that contain logic to execute transactions and modify the state of the assets stored within the ledger

In Hyperledger Fabric, chaincode is the ‘smart contract’ that runs on the peers and creates transactions. More broadly, it enables users to create transactions in the Hyperledger Fabric network’s shared ledger and update the world state of the assets.

Chaincode is programmable code, written in Go, and instantiated on a channel. Developers use chaincode to develop business contracts, asset definitions, and collectively-manage decentralized applications. The chaincode manages the ledger state through transactions invoked by applications.

Assets are created and updated by a specific chaincode, and cannot be accessed by another chaincode.

Chaincode runs in a secured Docker container isolated from the endorsing peer process. Chaincode initializes and manages ledger state through transactions submitted by applications.

Applications interact with the blockchain ledger through the chaincode. Therefore, the chaincode needs to be installed on every peer that will endorse a transaction and instantiated on the channel.

5.1.7 Simple overview of chaincode Program

When creating a chaincode, there are two methods that you will need to implement:

- **Init**

  Called when a chaincode receives an instantiate or upgrade transaction. This is where you will initialize any application state.

- **Invoke**

  Called when the invoke transaction is received to process any transaction proposals.

  As a developer if you write chaincode then you must create both an Init and an Invoke method within your chaincode. The chaincode must be installed using the peer chaincode install command, and instantiated using the peer chaincode instantiate command before the chaincode can be invoked. Then, transactions can be created using the peer chaincode invoke or peer chaincode query commands.

5.2 DETAILED SYSTEM DESING

Client application

5.2.1 Classification

The client application is the module of decentralized credit score system.as a subsystem.

5.2.2 Definition

This module is a web interface module through which user will interact.

5.2.3 Responsibilities

This module will be first responsible for user authentication and secondly, will be responsible for make transaction and fabric network. After processes the transaction, dapp will provide desire output and update data in distributed ledger.

5.2.4 Constraints

There is some constraint with this module

- The user machine should have 8 GB RAM plus Linux operating system
- It also needs internet connectivity to connect with the network
- Authenticate user can login in the client app
- If user want to see the credit score, then user should send the transaction
- When system validate the traction now user can only see the credit score

5.2.5 Composition

There are three sub components of this module such as web server, docker container and the fabric network camera.

5.2.6 Uses/Interactions

This module will be interacted with three other modules such as web server, docker container and fabric network.

5.2.7 Resources

This module requires resources from the Linux machine and all the software should be installed to run the application. They should have a web server where client can use dapp.

5.2.8 Processing

The first task that system is going to perform is authentication of user and after completion of this task user will be able to perform his primary task like make transaction to check credit score then user send transaction to the peer node. Now smart contract will accept or reject the transaction and display credit score to the user if transaction is successful accepted and system will commit or update all distributed ledger.

5.2.9 Interface/Exports

The data will be store in the form of key-value pair. The system provide exception if user not provide right authenticate credential to the system then system will generate the message to provide right credential. If transaction is rejected by system, then system will not be stop and display the message about transaction rejection.

The precise definition or declaration of each such element which are using in the system both abstractly or visually.

- **Anchor Peer**

  This gets used for communications between organizations. It makes peers in different organizations aware of each other.

- **Blocks**

  Consist of a header, block data (transactions) and block metadata (information about nodes involved with creating the block).

- **Certificate Authorities**

  Everyone who wants to interact with the networks needs an identity. The CA provides the means for each actor to have a verifiable digital identity. Hyperledger Fabric has a built in CA component for use in the blockchain network.

- **Chaincode**

The Hyperledger Fabric term for a smart contract. Note that chaincode does not have to be installed on every peer in a channel.

- **Channel**

A channel allows a group of participants to create a separate ledger of transactions. The transactions are only visible to the members of the channel.

- **Channel Configuration**

Rules that govern the channel and the channel is governed by the channel members. The channel configuration is separate from the network configuration.

- **Consortium**

A group of organizations that share a need to transact.

- **Committing Peer**

Every peer in the channel.

- **Endorsing Peer**

Every peer that has the smart contract installed can be an endorsing peer.

- **Endorsement Policy**

The rules for which organizations much approve a transaction before the other organizations will accept a copy. This is specific to the chaincode.

- **Leader Peer**

An organization can have multiple peers in a channel. Only one peer from the organization needs to receive the transactions. The leader distributes transactions from the orderers.

- **Membership Service Provider**

Is a trusted authority.

The MSP identifies which Root Certificate Authorities (CA) and Intermediate CA's are trusted by the network. The MSP identifies what roles different actors in an organization can play in the network. Nodes join the network through a Membership Service Provider.

- **Ledger**

This is an append only file while can be used to recreate the world state.

- **Ordering Nodes**

Is like a network administration point. The ordering nodes support the application channels for ordering transactions into blocks.

- **Peer Nodes**

Each peer maintains a copy of the ledger for each channel it is a member of.

- **Policies**

These determine who has control over the network configuration.

- **Private Data Collection**

This is used for keeping the data in a transaction confidential. The data is stored in a private database that is separated from the channel ledger.

- **Public Key Infrastructure**

This provides secure communication in a network. CAs issue digital certificates that get used to authenticate messages in the network. The PKI provides a list of identities and the MSP says which of them part of an organization are.

- **System Chaincode**

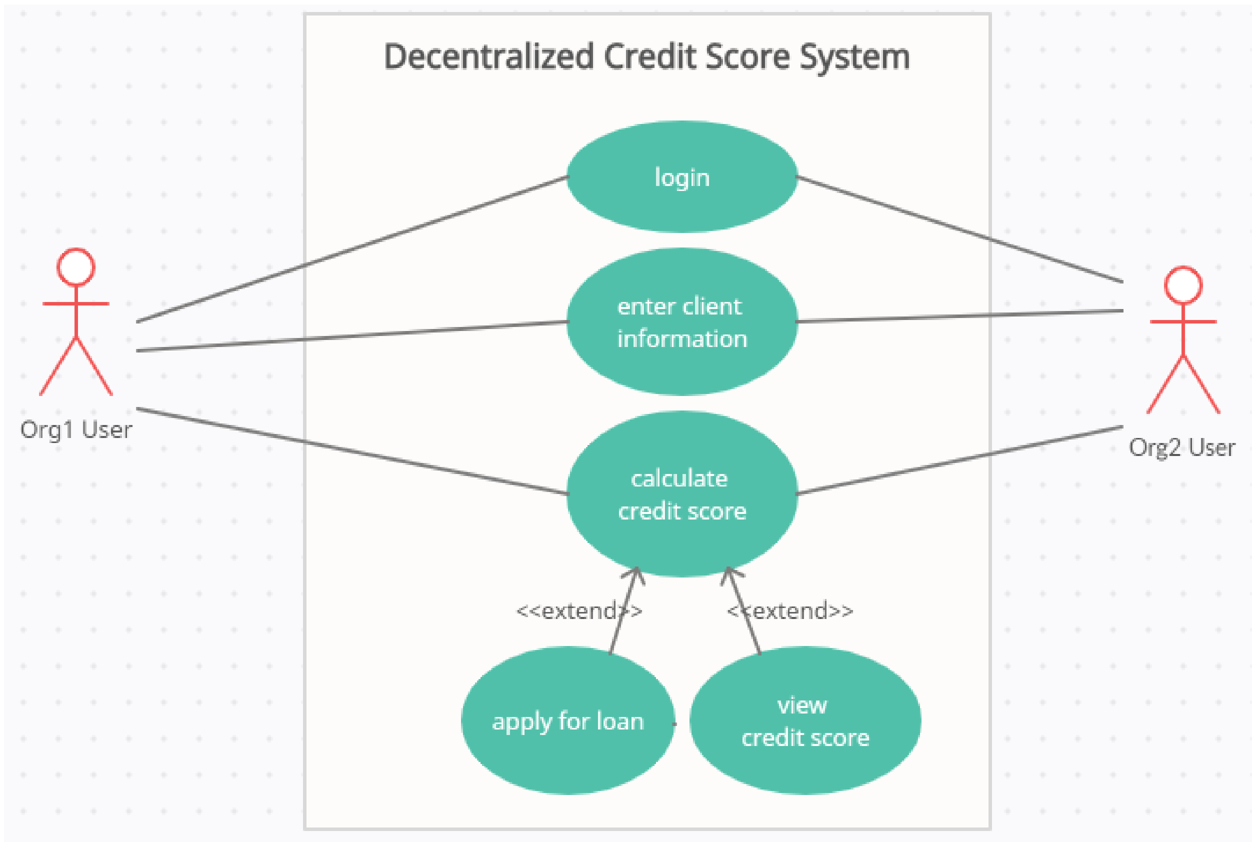
Code that defines operating parameters for the entire channel.

Lifecycle and configuration system chaincode defines the rules for the channel. Endorsement and validation system chaincode defines the requirements for endorsing and validating transactions.

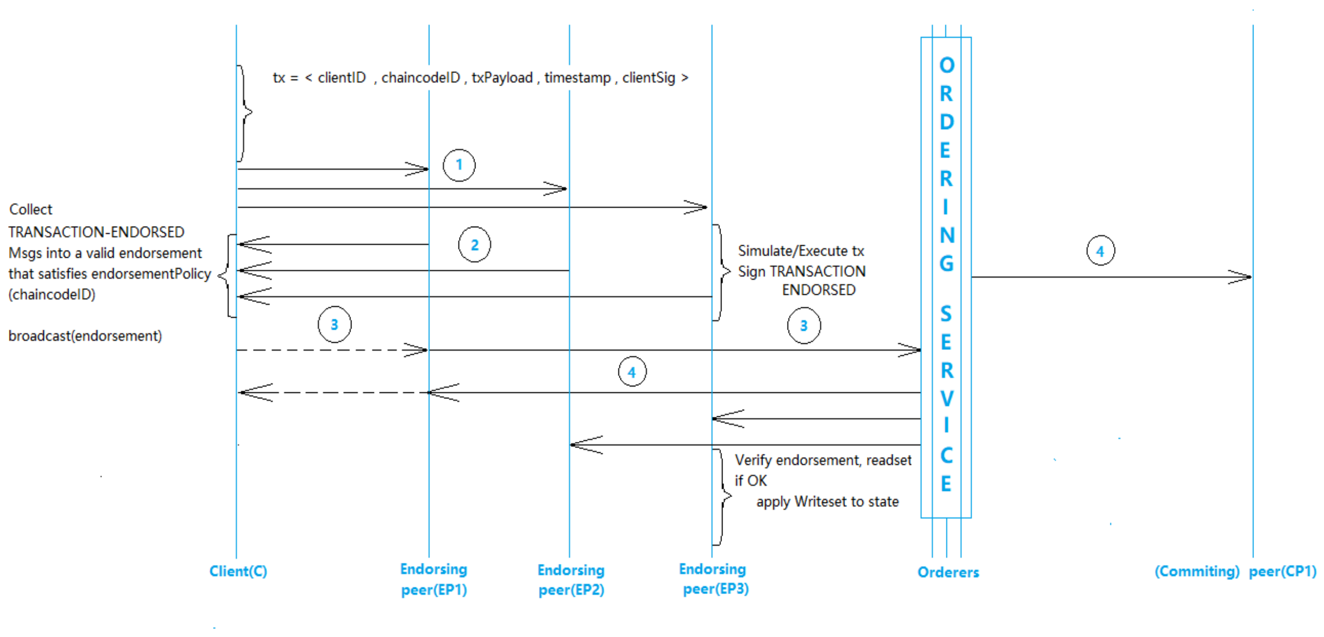
- **World State**

A database that holds current values of a set of ledger states.

5.3 Use Case Diagram

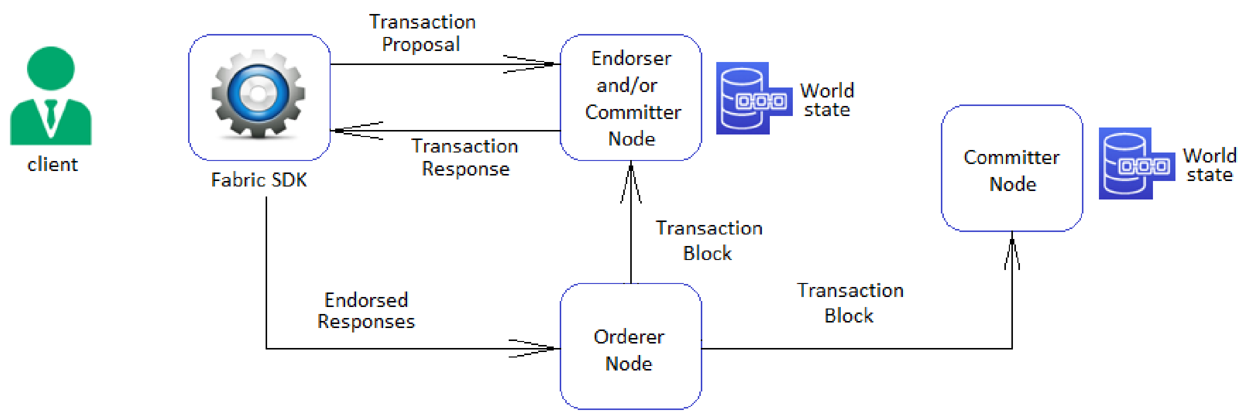


5.4 Sequence Diagram


tx = < clientID , chaincodeID , txPayload , timestamp , clientSig >
1
Collect TRANSACTION-ENDORSED Msgs into a valid endorsement that satisfies endorsementPolicy (chaincodeID)
2
Simulate/Execute tx Sign TRANSACTION ENDORSED
broadcast(endorsement)
3
3
4
4
Verify endorsement, readset if OK apply Writeset to state
ORDERING SERVICE
Client(C)
Endorsing peer(EP1)
Endorsing peer(EP2)
Endorsing peer(EP3)
Orderers
(Commiting) peer(CP1)

In the first setp the Client(C) send the information in key value form as transcation to the Endorsing Peer EP1, EP2 and EP3. The Endorsing Peer (EP3) will simaulate or exectite the trasaction. After the exeucuation it will sign or endrose the transaction and sent it to the Client(C). Now Client(C) will collect the endorosed transaction messages into a valid endorsement that satifies endorsement policy of chaincodeID. Now Client broadcast endorsement to the ordering service.Orderering service verify endorsement , readset . If it is ok the write set to state otherwise reject it.Now ordering servic send this transaction the committing peer or EP1,EP2 and EP3. EP1 will further send it to the Client(C) as a response.

5.5 Transaction Flow Diagram



Hyperledger Fabric Transaction Flow

When clients submit the transaction proposal through the Fabric SDK, this proposal is sent to all Endorsing Peers. These endorsing peers check the transaction verifies and executes and generate the Read and Write set as output. Now, this response is again sent to the client. The client collects all responses from all endorsing peers, and send them to Orderer. Now, Orderer sees all transactions and orders them in ascending order and form a block. Now, this block is sent to all committers which checks the transaction and add a new block in their own copy of the ledger.

Chapter 6

IMPLEMENTATION AND TESTING

Our project “Decentralized credit score system” is web-based application. Our system is consists of two organizations. The whole project is mode up multiple components but at the high-level components are (1) Hyperledger fabric framework, (2) server side, (3) client and further more component are mentioned below

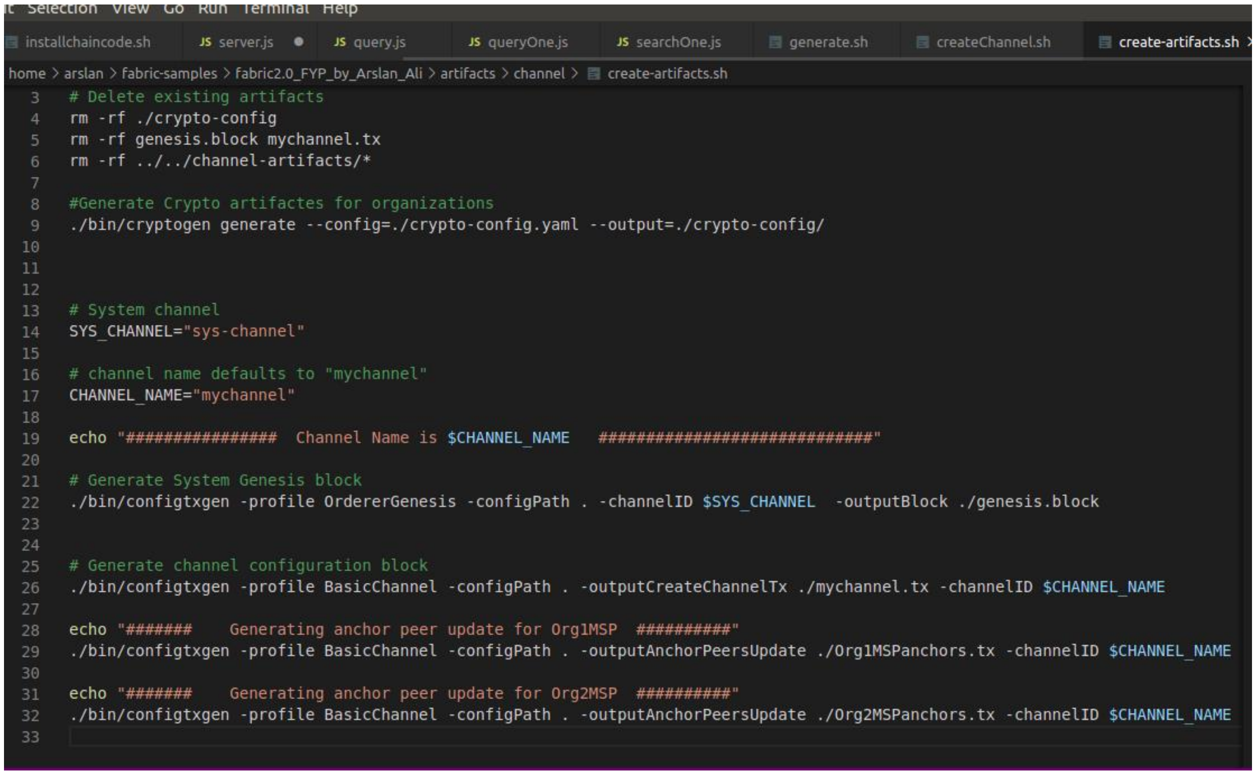
6.1 HYPERLEDGER FABRIC NETWORK

Hyperledger fabric framework comes under the umbrella of Hyperledger, hosted by Linux foundation .Hyperledger fabric is distributed ledger technology that develop to solve the enterprises problems. Fabric architecture delivers scalability, flexibility, resiliency, and high degrees of confidentiality. It’s designed to support pluggable implementations of different components to solve the different level of problem.

6.1.1 Fabric binaries:

Hyperledger fabric binaries are tools, which are used to develop the architecture of organizations admin, and users and credentials as well. We use different binaries tools for different sort of operation. Fabric binaries contains.

- **Cryptogen**: Cryptogen tool is used generate the crypto material for organizations.
- **Configtxgen**: configtxgen tool is used to create channel configuration artifacts, and other like genesis block, anchor peer.
- **Peer** : peer binary is used to create peers to communicate within the organization and with other organizations
- **Orderer** : order binary tool is used to put the transactions in ordering way
- **Fabric-ca-client**: this binary tool is used to fabric CA and users.

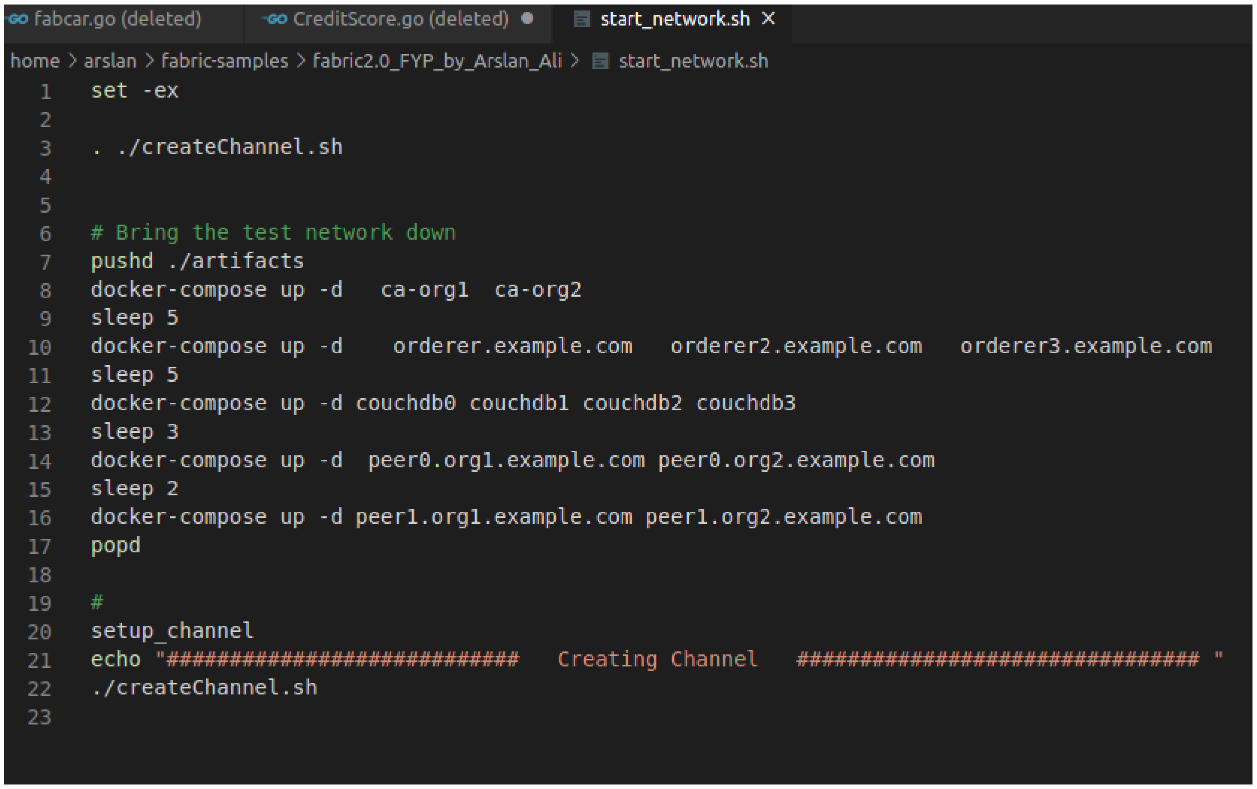

Selection View Go Run Terminal Help
installchaincode.sh
server.js
query.js
queryOne.js
searchOne.js
generate.sh
createChannel.sh
create-artifacts.sh
home > arslan > fabric-samples > fabric2.0_FYP_by_Arslan_Ali > artifacts > channel > create-artifacts.sh

```
# Delete existing artifacts
rm -rf ./crypto-config
rm -rf genesis.block mychannel.tx
rm -rf ../../channel-artifacts/*

#Generate Crypto artifactes for organizations
./bin/cryptogen generate --config=./crypto-config.yaml --output=./crypto-config/



# System channel
SYS_CHANNEL="sys-channel"

# channel name defaults to "mychannel"
CHANNEL_NAME="mychannel"

echo "###############  Channel Name is $CHANNEL_NAME   #############################"

# Generate System Genesis block
./bin/configtxgen -profile OrdererGenesis -configPath . -channelID $SYS_CHANNEL  -outputBlock ./genesis.block


# Generate channel configuration block
./bin/configtxgen -profile BasicChannel -configPath . -outputCreateChannelTx ./mychannel.tx -channelID $CHANNEL_NAME

echo "#######    Generating anchor peer update for Org1MSP  ##########"
./bin/configtxgen -profile BasicChannel -configPath . -outputAnchorPeersUpdate ./Org1MSPanchors.tx -channelID $CHANNEL_NAME

echo "#######    Generating anchor peer update for Org2MSP  ##########"
./bin/configtxgen -profile BasicChannel -configPath . -outputAnchorPeersUpdate ./Org2MSPanchors.tx -channelID $CHANNEL_NAME
```

Figure 5   Using Cryptogen & configtxgen

6.1.2 Docker Compose

Basically our project is developed with containerization concept .Where we use images and containers, in our project every entity is individual container, for that we use docker compose to manage multiple containers/entities



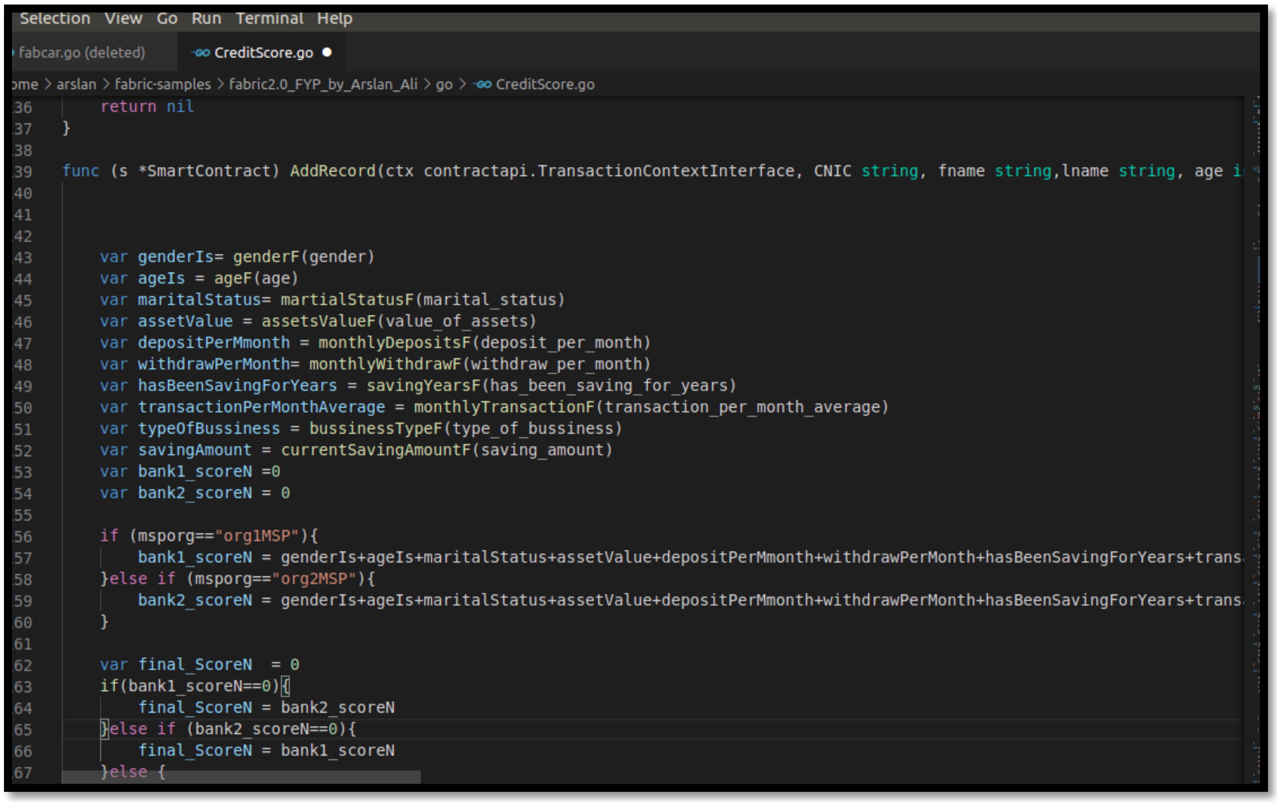
fabcar.go (deleted)
CreditScore.go (deleted)
start_network.sh
home > arslan > fabric-samples > fabric2.0_FYP_by_Arslan_Ali > start_network.sh

```
set -ex

. ./createChannel.sh


# Bring the test network down
pushd ./artifacts
docker-compose up -d   ca-org1  ca-org2
sleep 5
docker-compose up -d    orderer.example.com   orderer2.example.com   orderer3.example.com
sleep 5
docker-compose up -d couchdb0 couchdb1 couchdb2 couchdb3
sleep 3
docker-compose up -d  peer0.org1.example.com peer0.org2.example.com
sleep 2
docker-compose up -d peer1.org1.example.com peer1.org2.example.com
popd

#
setup_channel
echo "############################   Creating Channel   ################################ "
./createChannel.sh

```

Figure 6 Running the containers

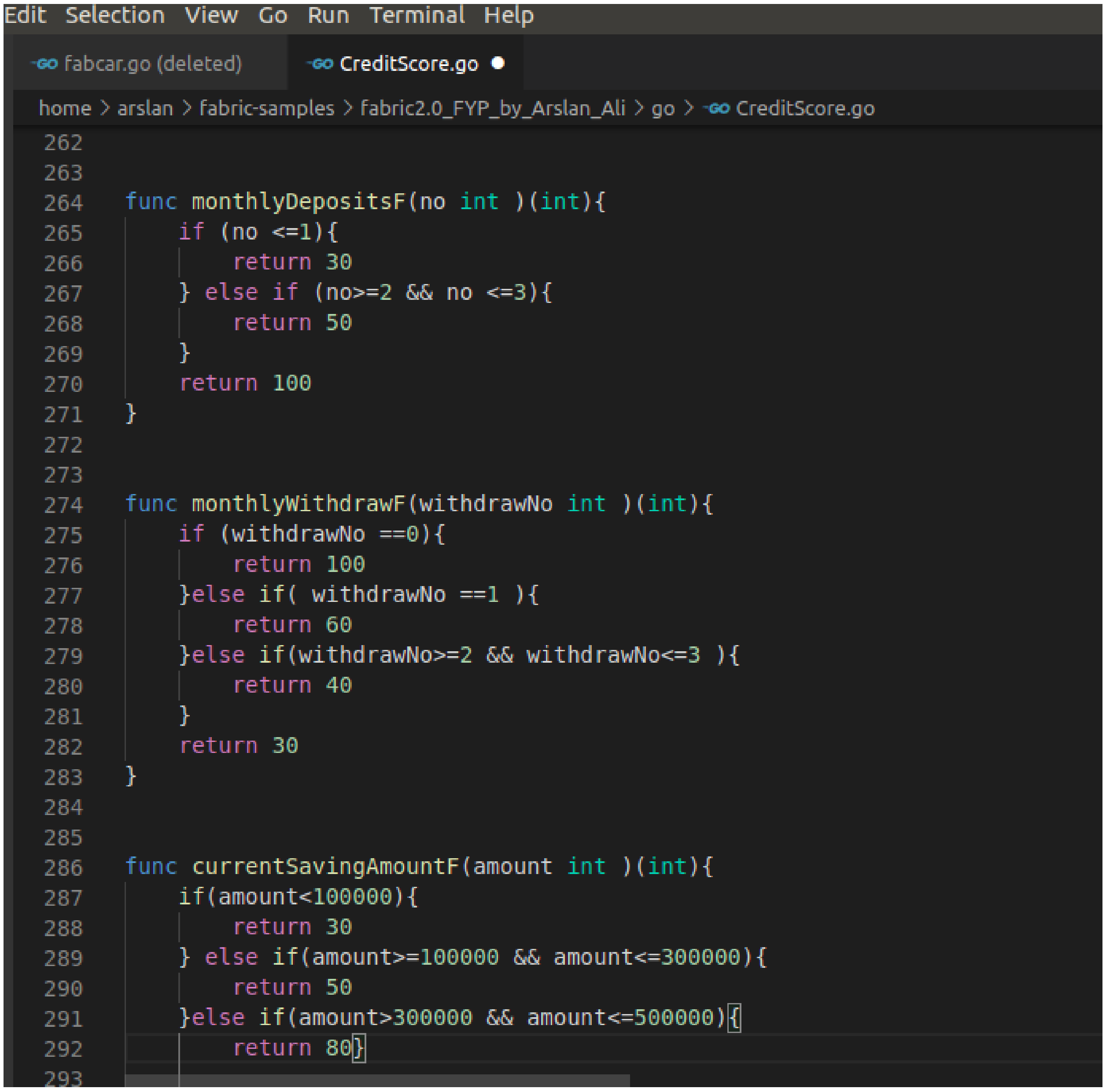
6.1.3 Chaincode (Go language)

Go language is programming language.it is efficient, reliable and easy to use. Hyperledger supports many programing languages in the development of chaincode (smart contract) such as go, java, JavaScript (Node.js) .But we use go language for chaincode development, because go language is base and default chaincode language


Selection View Go Run Terminal Help
fabcar.go (deleted)
CreditScore.go
ome > arslan > fabric-samples > fabric2.0_FYP_by_Arslan_Ali > go > CreditScore.go

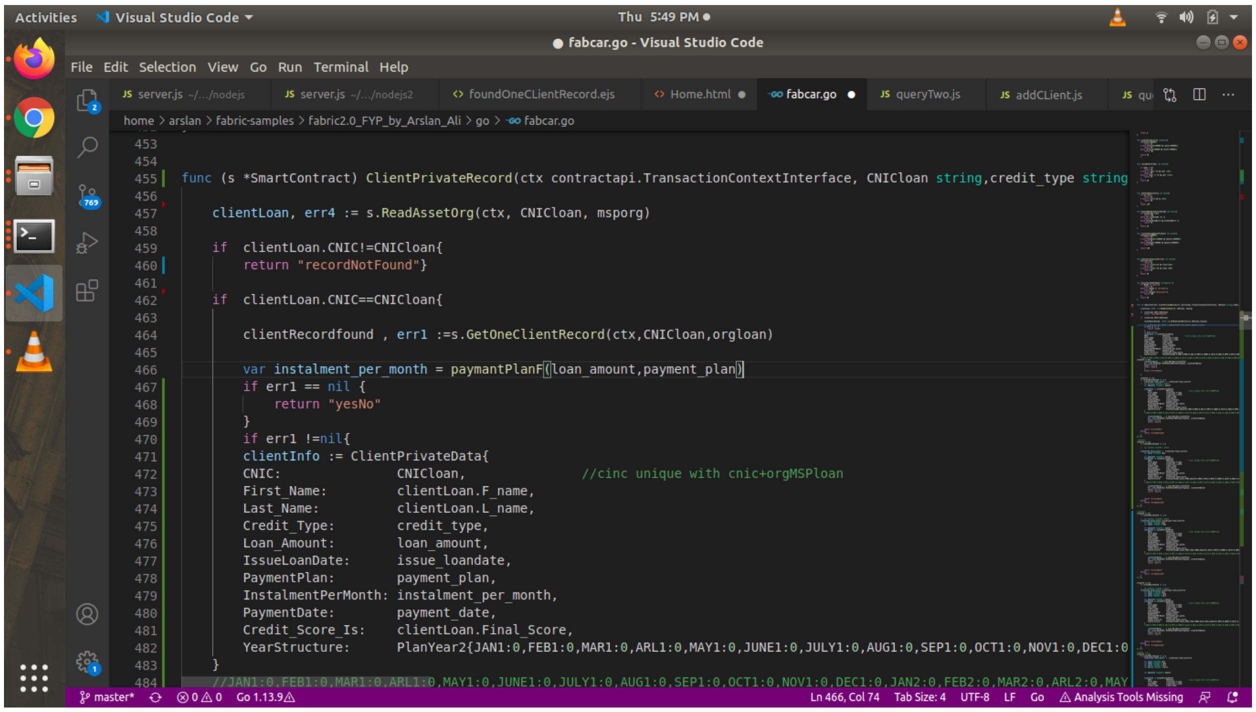
```
    return nil
}

func (s *SmartContract) AddRecord(ctx contractapi.TransactionContextInterface, CNIC string, fname string,lname string, age i



    var genderIs= genderF(gender)
    var ageIs = ageF(age)
    var maritalStatus= martialStatusF(marital_status)
    var assetValue = assetsValueF(value_of_assets)
    var depositPerMmonth = monthlyDepositsF(deposit_per_month)
    var withdrawPerMonth= monthlyWithdrawF(withdraw_per_month)
    var hasBeenSavingForYears = savingYearsF(has_been_saving_for_years)
    var transactionPerMonthAverage = monthlyTransactionF(transaction_per_month_average)
    var typeOfBussiness = bussinessTypeF(type_of_bussiness)
    var savingAmount = currentSavingAmountF(saving_amount)
    var bank1_scoreN =0
    var bank2_scoreN = 0

    if (msporg=="org1MSP"){
        bank1_scoreN = genderIs+ageIs+maritalStatus+assetValue+depositPerMmonth+withdrawPerMonth+hasBeenSavingForYears+trans
    }else if (msporg=="org2MSP"){
        bank2_scoreN = genderIs+ageIs+maritalStatus+assetValue+depositPerMmonth+withdrawPerMonth+hasBeenSavingForYears+trans
    }

    var final_ScoreN  = 0
    if(bank1_scoreN==0){
        final_ScoreN = bank2_scoreN
    }else if (bank2_scoreN==0){
        final_ScoreN = bank1_scoreN
    }else {
```

Figure 7 Inserting Client Record in Chaincode


Edit Selection View Go Run Terminal Help
fabcar.go (deleted)
CreditScore.go
home > arslan > fabric-samples > fabric2.0_FYP_by_Arslan_Ali > go > CreditScore.go

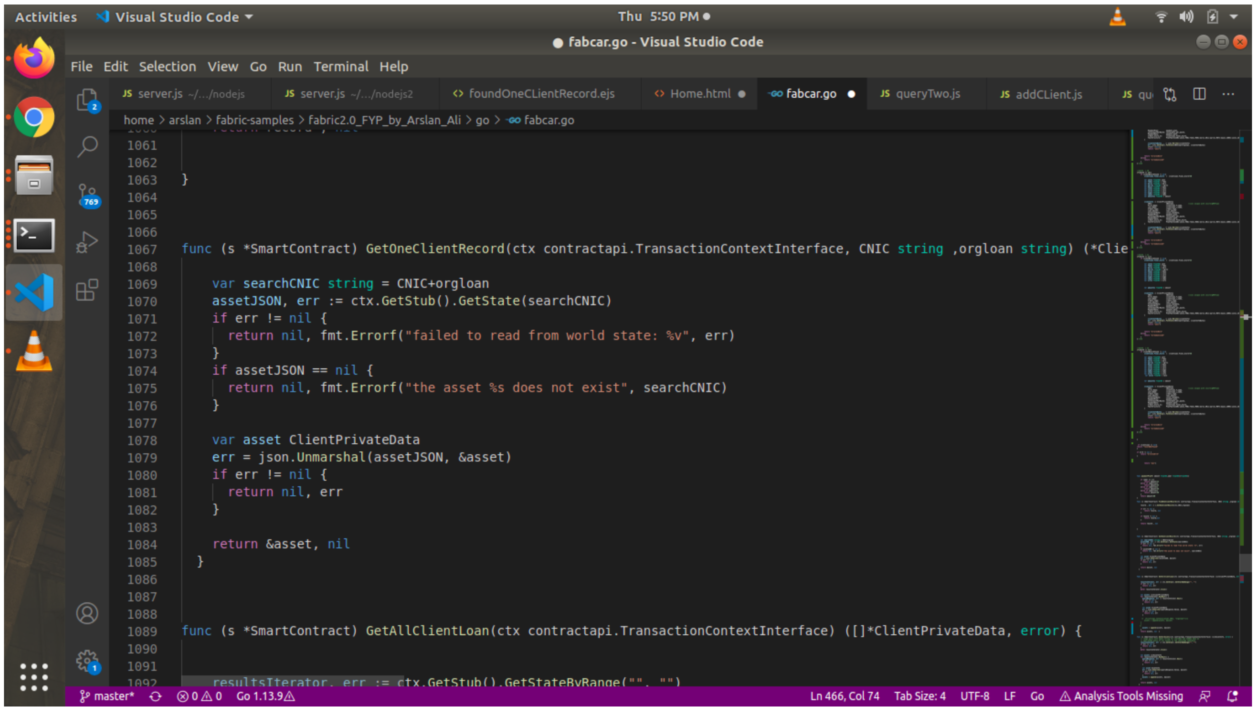
```go


func monthlyDepositsF(no int )(int){
    if (no <=1){
        return 30
    } else if (no>=2 && no <=3){
        return 50
    }
    return 100
}


func monthlyWithdrawF(withdrawNo int )(int){
    if (withdrawNo ==0){
        return 100
    }else if( withdrawNo ==1 ){
        return 60
    }else if(withdrawNo>=2 && withdrawNo<=3 ){
        return 40
    }
    return 30
}


func currentSavingAmountF(amount int )(int){
    if(amount<100000){
        return 30
    } else if(amount>=100000 && amount<=300000){
        return 50
    }else if(amount>300000 && amount<=500000){
        return 80}
```

Figure 8   Calculating the Credit Score



```
func (s *SmartContract) ClientPrivateRecord(ctx contractapi.TransactionContextInterface, CNICloan string,credit_type string

    clientLoan, err4 := s.ReadAssetOrg(ctx, CNICloan, msporg)

    if  clientLoan.CNIC!=CNICloan{
        return "recordNotFound"}

    if  clientLoan.CNIC==CNICloan{

        clientRecordfound , err1 :=s.GetOneClientRecord(ctx,CNICloan,orgloan)

        var instalment_per_month = paymantPlanF(loan_amount,payment_plan)
        if err1 == nil {
            return "yesNo"
        }
        if err1 !=nil{
        clientInfo := ClientPrivateData{
        CNIC:               CNICloan,               //cinc unique with cnic+orgMSPloan
        First_Name:         clientLoan.F_name,
        Last_Name:          clientLoan.L_name,
        Credit_Type:        credit_type,
        Loan_Amount:        loan_amount,
        IssueLoanDate:      issue_loandate,
        PaymentPlan:        payment_plan,
        InstalmentPerMonth: instalment_per_month,
        PaymentDate:        payment_date,
        Credit_Score_Is:    clientLoan.Final_Score,
        YearStructure:      PlanYear2{JAN1:0,FEB1:0,MAR1:0,ARL1:0,MAY1:0,JUNE1:0,JULY1:0,AUG1:0,SEP1:0,OCT1:0,NOV1:0,DEC1:0
    }
    //JAN1:0,FEB1:0,MAR1:0,ARL1:0,MAY1:0,JUNE1:0,JULY1:0,AUG1:0,SEP1:0,OCT1:0,NOV1:0,DEC1:0,JAN2:0,FEB2:0,MAR2:0,ARL2:0,MAY
```

Figure 9 Issuing Loan



```


}



func (s *SmartContract) GetOneClientRecord(ctx contractapi.TransactionContextInterface, CNIC string ,orgloan string) (*Clie

    var searchCNIC string = CNIC+orgloan
    assetJSON, err := ctx.GetStub().GetState(searchCNIC)
    if err != nil {
      return nil, fmt.Errorf("failed to read from world state: %v", err)
    }
    if assetJSON == nil {
      return nil, fmt.Errorf("the asset %s does not exist", searchCNIC)
    }

    var asset ClientPrivateData
    err = json.Unmarshal(assetJSON, &asset)
    if err != nil {
      return nil, err
    }

    return &asset, nil
  }



func (s *SmartContract) GetAllClientLoan(ctx contractapi.TransactionContextInterface) ([]*ClientPrivateData, error) {

    resultsIterator, err := ctx.GetStub().GetStateByRange("", "")
```

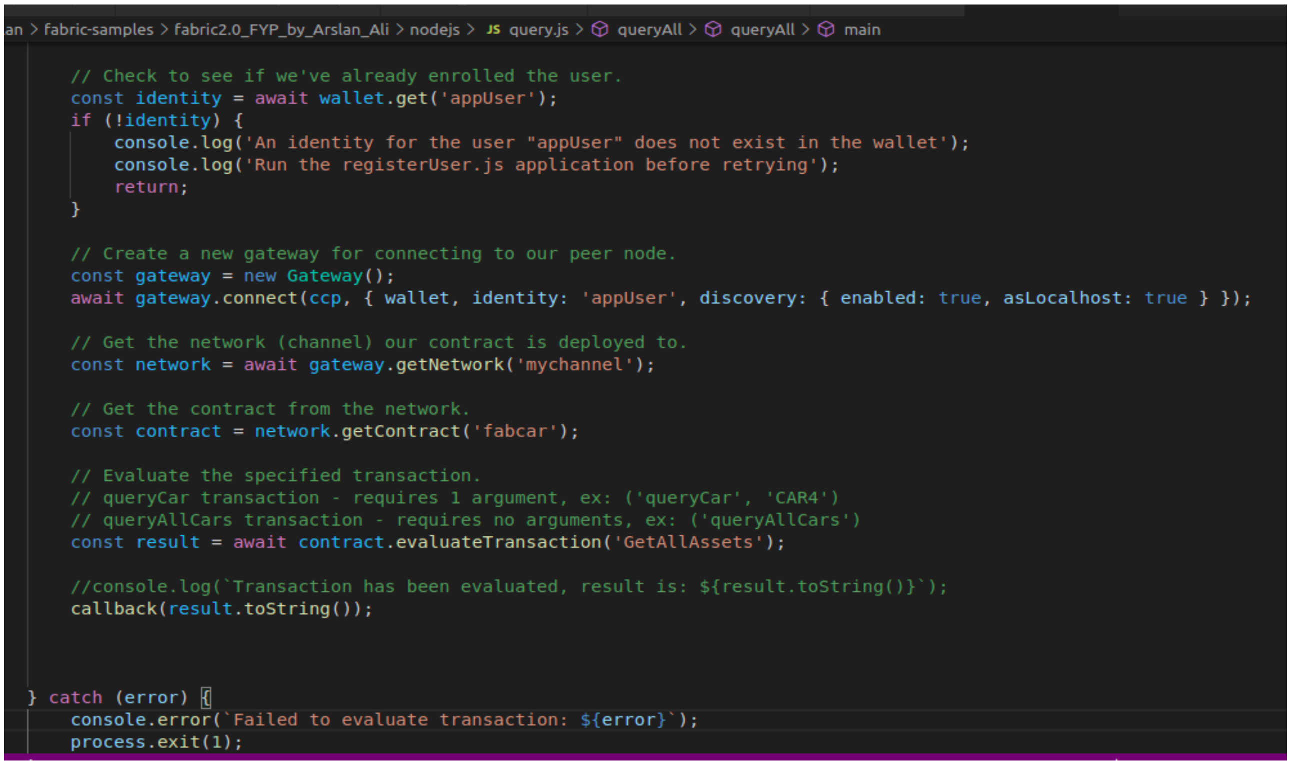
Figure 10 Retrieving Client Data

6.1.4 Bash Script

Linux Bash script is command line interface (CLI) used to direct interact with the operation system (OS).It is very powerful interface it has high priority among other interface. it is also used for system administration .A bash script is simple plain text file which contain no: of commands to be executed .we use bash script in our project to developed and up the fabric network.to run the fabric files bash script has important role .

6.2 NODE SDK

Node.js is JavaScript runtime environment. It is open-source and cross-platform, which execute JavaScript code at client side as well as server side. Developers of node.js use JavaScript code for server-side to develop the web pages with dynamic content. It is very popular paradigm because single programming language supports both client and server side. Node sdk also use to interact with Hyperledger fabric network by using gateway object.Gatway is object, which establish the connection peer and verify the identity with wallet.


an > fabric-samples > fabric2.0_FYP_by_Arslan_Ali > nodejs > JS query.js > queryAll > queryAll > main

```
    // Check to see if we've already enrolled the user.
    const identity = await wallet.get('appUser');
    if (!identity) {
        console.log('An identity for the user "appUser" does not exist in the wallet');
        console.log('Run the registerUser.js application before retrying');
        return;
    }

    // Create a new gateway for connecting to our peer node.
    const gateway = new Gateway();
    await gateway.connect(ccp, { wallet, identity: 'appUser', discovery: { enabled: true, asLocalhost: true } });

    // Get the network (channel) our contract is deployed to.
    const network = await gateway.getNetwork('mychannel');

    // Get the contract from the network.
    const contract = network.getContract('fabcar');

    // Evaluate the specified transaction.
    // queryCar transaction - requires 1 argument, ex: ('queryCar', 'CAR4')
    // queryAllCars transaction - requires no arguments, ex: ('queryAllCars')
    const result = await contract.evaluateTransaction('GetAllAssets');

    //console.log(`Transaction has been evaluated, result is: ${result.toString()}`);
    callback(result.toString());


} catch (error) {
    console.error(`Failed to evaluate transaction: ${error}`);
    process.exit(1);
```

Figure 11 Node SDK

6.2.1 npm

Npm is node module registry.it is open source community where everyone can use the node module and can contribute in npm .multiple organization use npm in there system. The npm is also used node SDK packages to interact the application with fabric network. The node sdk package API used to interact with orderer, peers, fabric network, the transaction is also done with the help of npm packages, that direct call the function in chaincode .there is another npm package for Hyperledger fabric used to interact with fabric-ca and manages the user identity

6.2.2 REST architecture

REST (REpresentational State Transfer) uses the http protocol to send and receive the data .In REST architecture server provides the resource to access where client is access the provided server using http protocol .there are four http method which are commonly used in REST .

- GET – is used to allow read only access to client
- PUT – PUT is used to allow user to update the existing data
- DELETE – to delete the data.
- POST –used to create the data at the server.

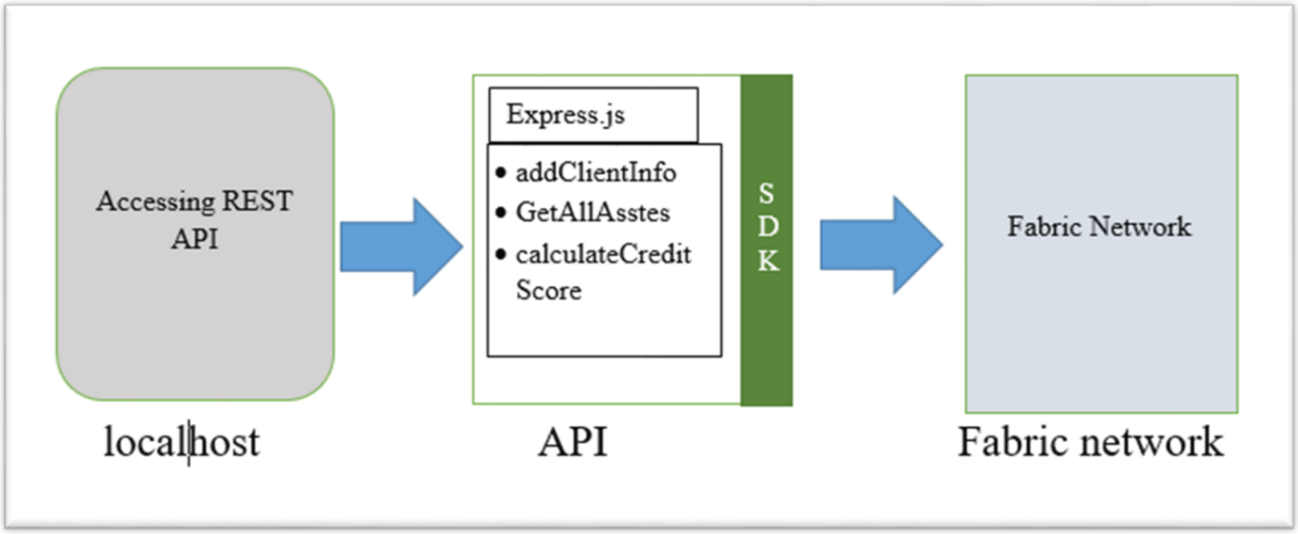


Figure 12 REST Architecture with fabric network

6.3 USER INTERFACE

Client-Side script are those that run for Application user. As we discuss that our system is web based, so for, there should be interface to user through which he can interact with system .In our system we use multiple language html, css, JavaScript, Bootstrap. Each language has its own functionality. Html is use for basic structure .css and bootstrap is use together for graphical interface like button, textbox, label, etc. In the last one JavaScript is used for invalidation .The input validation is called front-end validation, where every user input is validate. If the input user is valid, the desire action will be perform, but vise-versa if the user input is not valid.

Chapter 7

RESULTS AND DISCUSSION

Our system "Decentralized credit score system" developed to calculate the credit score of borrower .Our system is type of permissioned Blockchain network. Initially we have two organizations that are allow participating in our System (Blockchain network). In our system both organization have equal rights to do any operations. They can insert their client record in the distributed ledgers, calculate the credit score and Maintain the credit history and credit score of each individual. However, there is also limitation of each organization in order to maintain the confidentiality. Our system does not allow organizations' to view the client credit history of other organizations client. They can view the credit history of their own client.

7.1 Snap shots of system

7.1.1 Login screen

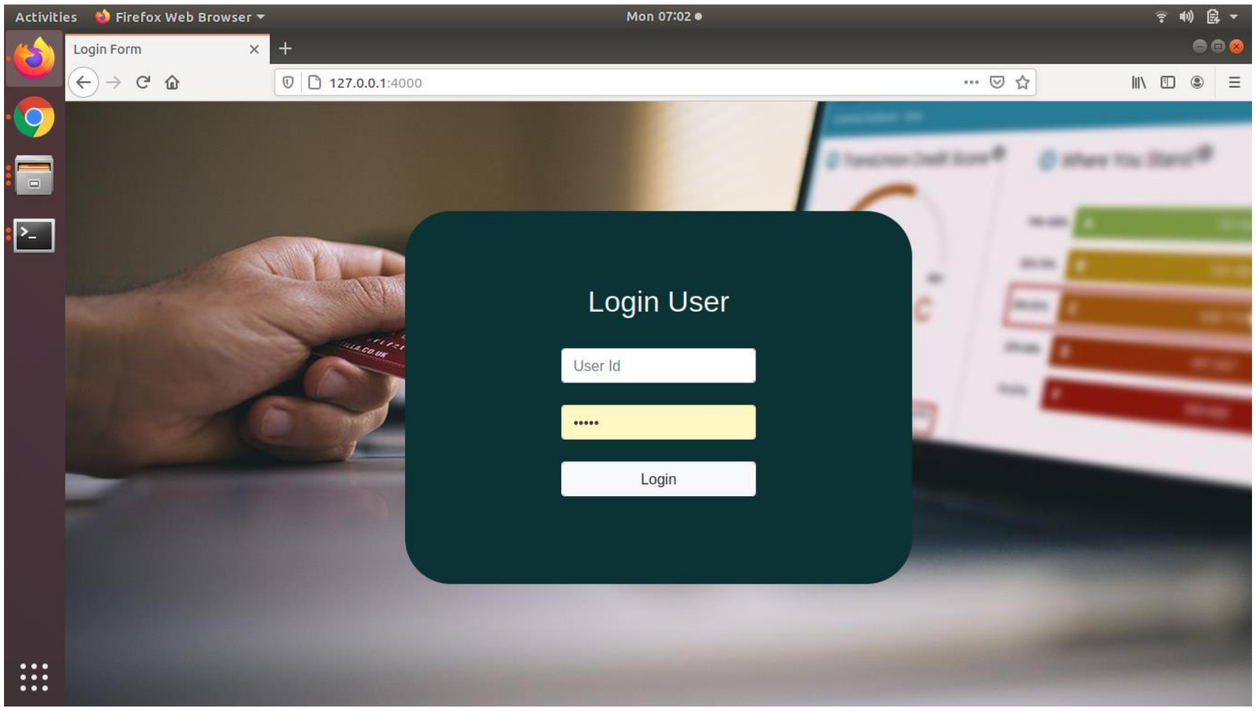


Figure 13 Login Screen

The first screen in our system, which is use to authenticate the user. The authentication system require userid, password to login successfully. User credentials userid and password should be provide by user's organization. If the user successfully login into the system, it means the user is trusted and now he can interact with system.

7.1.2 Home screen

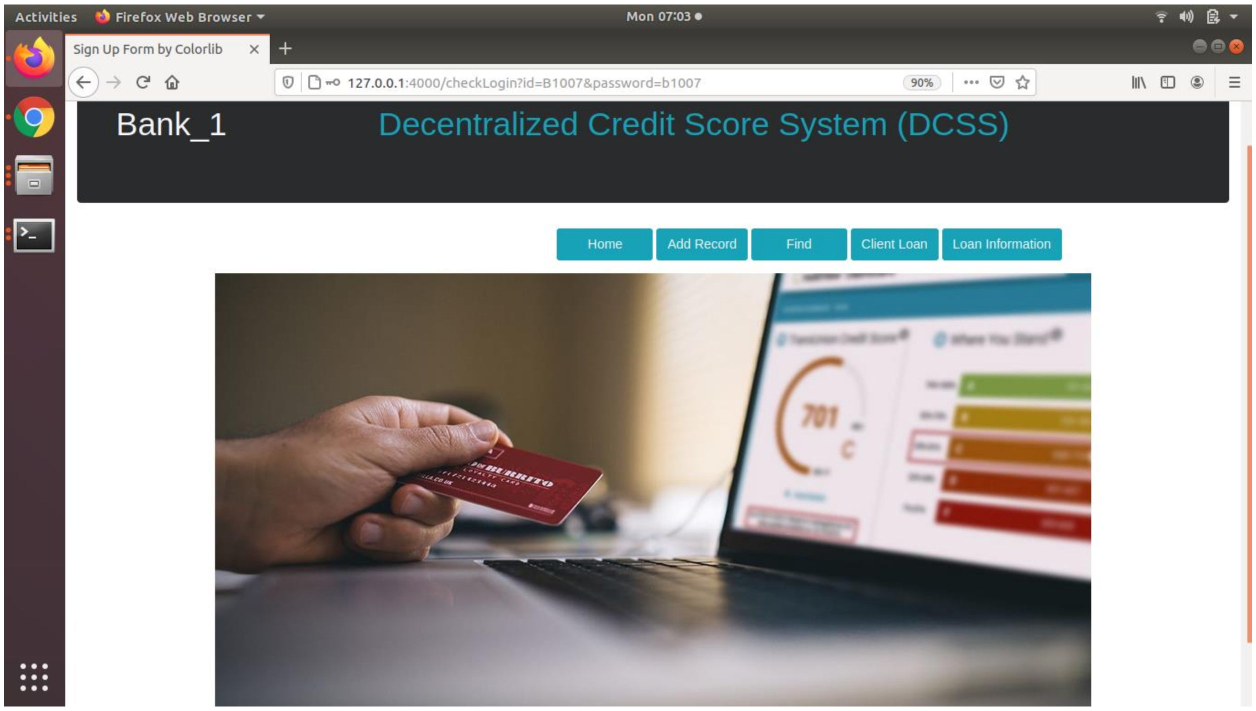


Figure 14 Home Screen Bank 1

This is Home screen of Bank 1. After successful login into the system, now the users can perform actions like add record, view record, grant the loan, and view the credit history of borrower.

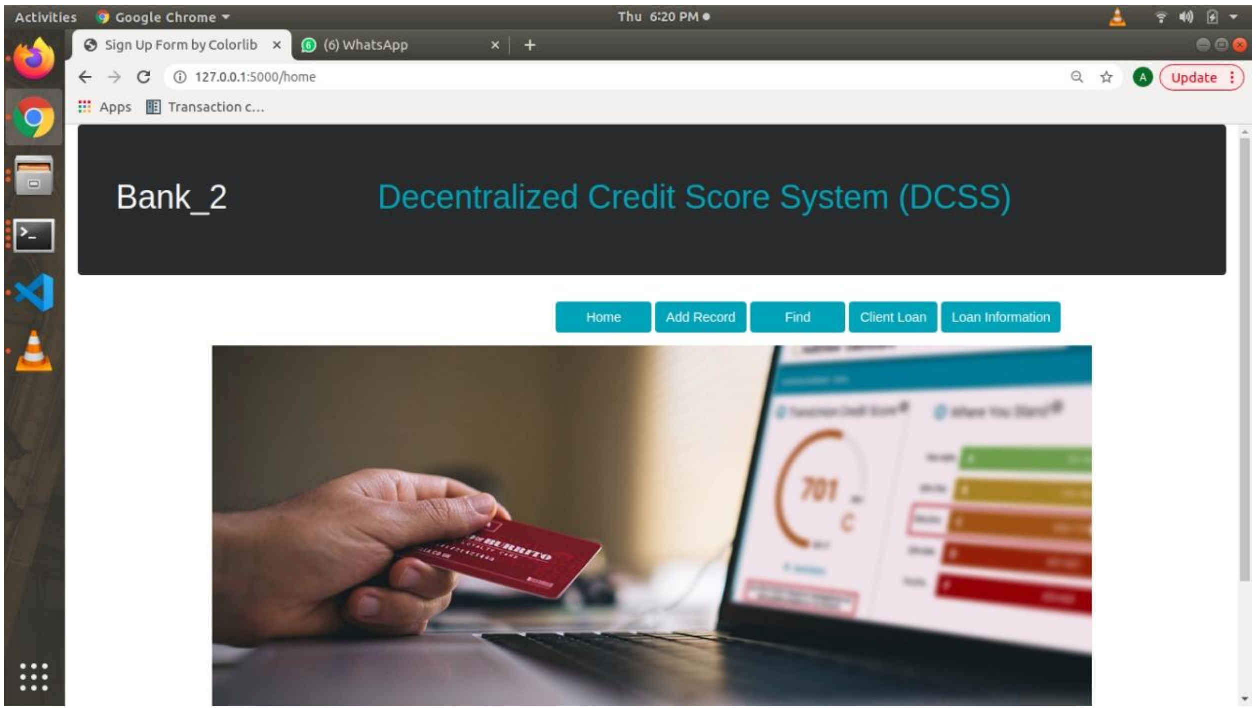
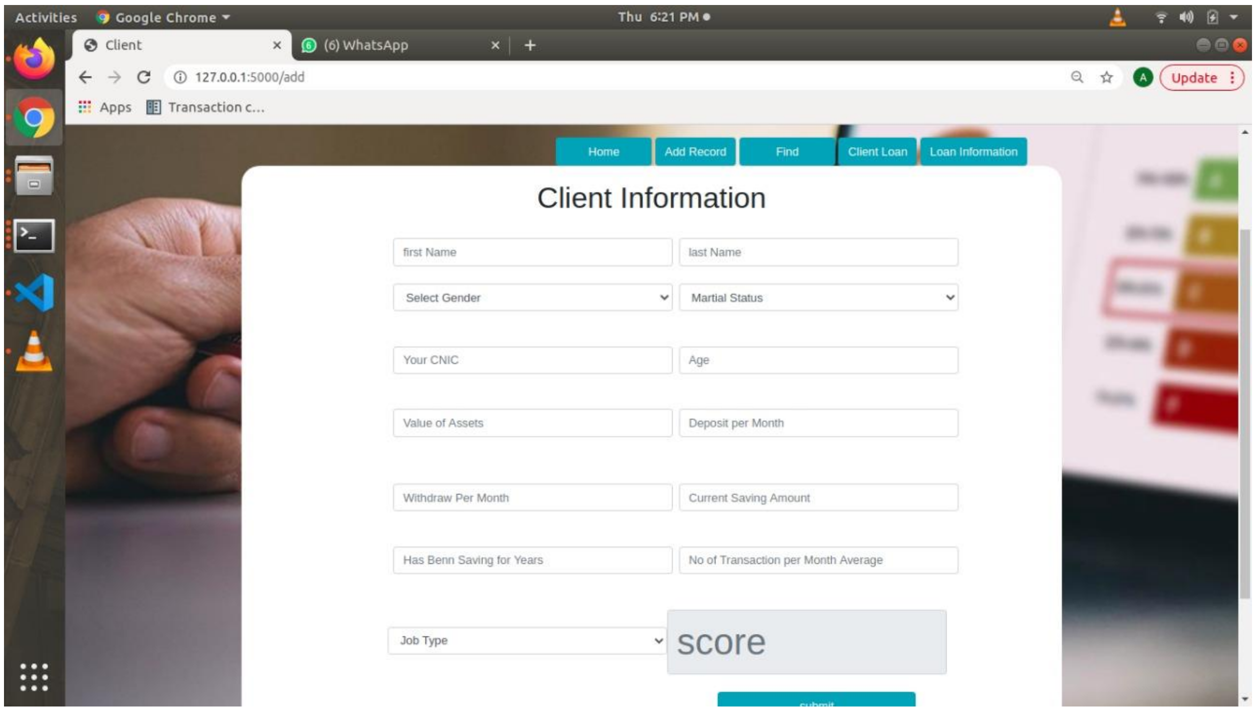


Figure 15 Home Screen Bank 2

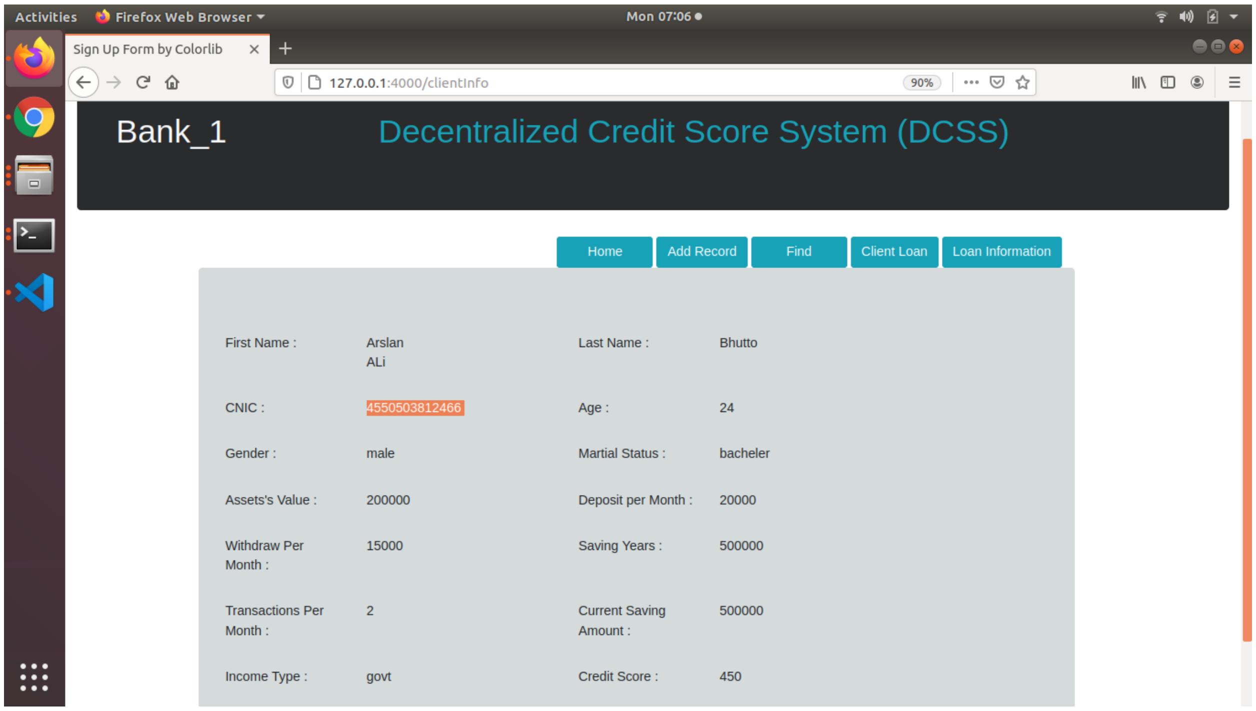
This is Home screen of Bank 2. After successful login into the system, now the users can perform actions like add record, view record, grant the loan, and view the credit history of borrower.



Client Information

first Name	last Name
Select Gender	Martial Status
Your CNIC	Age
Value of Assets	Deposit per Month
Withdraw Per Month	Current Saving Amount
Has Benn Saving for Years	No of Transaction per Month Average
Job Type	score

Figure 16 Client Profile Form

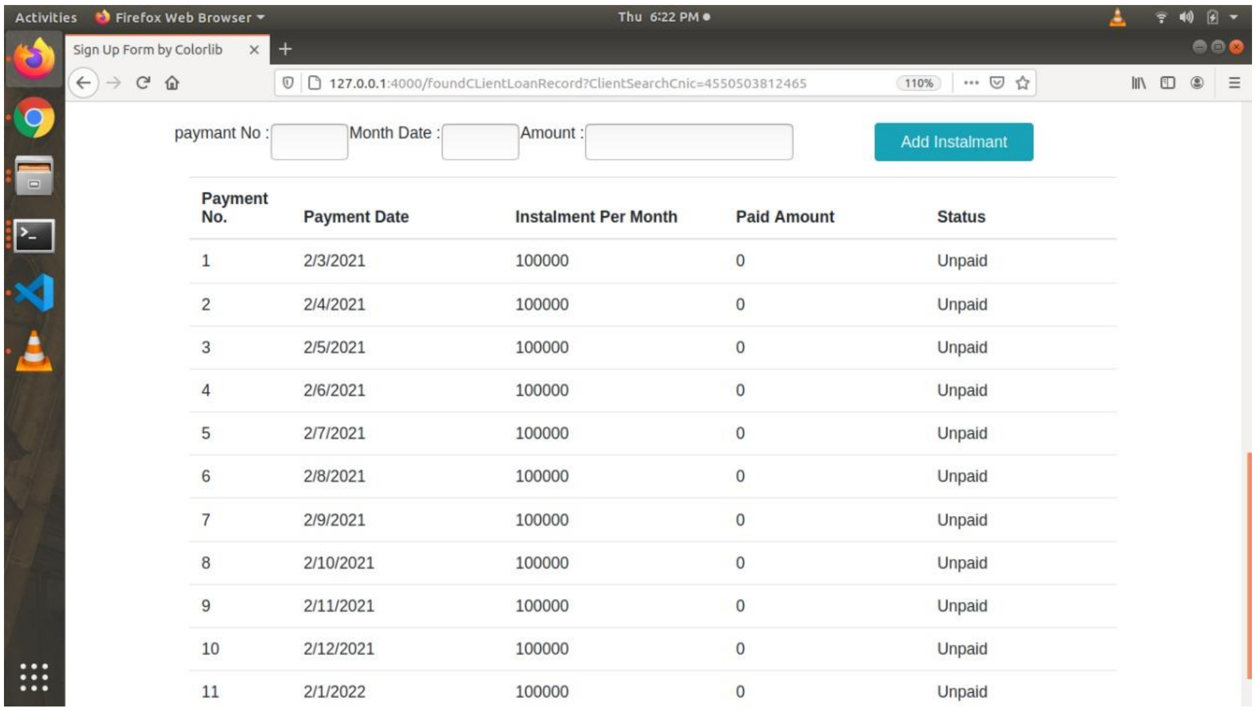


Bank_1

Decentralized Credit Score System (DCSS)

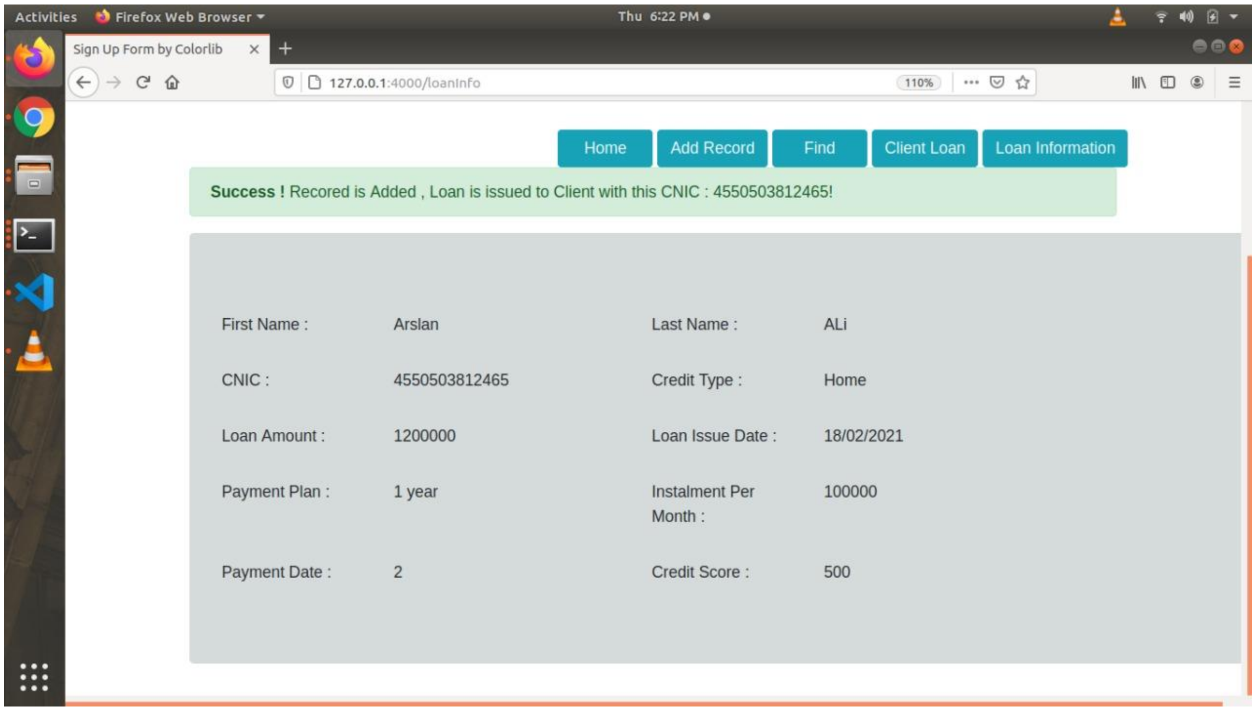
First Name :	Arslan ALi	Last Name :	Bhutto
CNIC :	4550503812466	Age :	24
Gender :	male	Martial Status :	bacheler
Assets's Value :	200000	Deposit per Month :	20000
Withdraw Per Month :	15000	Saving Years :	500000
Transactions Per Month :	2	Current Saving Amount :	500000
Income Type :	govt	Credit Score :	450

Figure 17 Client Information



Payment No.	Payment Date	Instalment Per Month	Paid Amount	Status
1	2/3/2021	100000	0	Unpaid
2	2/4/2021	100000	0	Unpaid
3	2/5/2021	100000	0	Unpaid
4	2/6/2021	100000	0	Unpaid
5	2/7/2021	100000	0	Unpaid
6	2/8/2021	100000	0	Unpaid
7	2/9/2021	100000	0	Unpaid
8	2/10/2021	100000	0	Unpaid
9	2/11/2021	100000	0	Unpaid
10	2/12/2021	100000	0	Unpaid
11	2/1/2022	100000	0	Unpaid

Figure 18 Loan Installment Profile



Success ! Recored is Added , Loan is issued to Client with this CNIC : 4550503812465!

First Name :	Arslan	Last Name :	ALi
CNIC :	4550503812465	Credit Type :	Home
Loan Amount :	1200000	Loan Issue Date :	18/02/2021
Payment Plan :	1 year	Instalment Per Month :	100000
Payment Date :	2	Credit Score :	500

Figure 19 Loan Issuing Profile

Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

There propose solution "Decentralized Credit Score System" solve the following problems that we were facing in centralized or manual credit scoring system. Like one bank cannot share data with other bank for lending services. Now our proposed solution has ability where one bank can share data with other bank. The other major problem that we solved in this system that we have removed third party credit bureau in our systems. Because banks were not satisfied with credit bureau, credit bureau provide false information to the banks and by using that information credit bureau calculate the credit score, which is the big alarm for banks. In our proposed solution, each bank add information of his/her client on distributed ledger. Because distributed ledger technology provide transparency, immutability of data. For achieving this goal, we made our system using Hyperledger fabric, which is the latest version of blockchain technology.

Once we stored information, the system automatically calculate the client credit scoring by looking at its financial history. If client credit score meets the loan approval then he/she can apply for loan. We also add the functionality where we can maintained the history of loan installment and if client paid loan installment on time which makes credit score better else if client will not paid installment on time or unpaid then client credit score will be decreased. If client unable to paid installment that makes client score worst then system automatically treat him as a deflator client. If other defaulter client apply for loan in other bank then systems automatically show client as a defaulter.

8.2 Future work

In future, we can make our system for multiple organization now it is specific for only two organization. Now client can apply for three types of loan (vehicle, business, home, credit card).In future we can add multiple types of loan for our client. Now we are calculating credit score by using static formula based on financial history. In future, we can used artificial intelligence to calculate the credit score of a client. Because we know that, it is era of artificial and we can take benefits from this AI community. We can used these things to make our system more attractive and reliable. Multiple organization can take more benefit from this type system and extends their business in term of lending services where they will provide loan to their client.

Chapter 9

REFERENCES

[1] Nakamoto, Satoshi, and A. Bitcoin. "A peer-to-peer electronic cash system." *Bitcoin.–URL: https://bitcoin. org/bitcoin. pdf* 4 (2008).

[2] Buterin, V. "A next generation smart contract & decentralized application platform (2013) whitepaper." *Ethereum Foundation*.

[3] https://hyperledger.github.io/fabric-sdk-node/

[4] https://epic.org/privacy/data-breach/equifax/

[5] https://www.forbes.com/sites/oluwaseunadeyanju/2020/04/14/bloom-partners-with-transunion-to-bring-consumer-credit-data-to-blockchain/?sh=7a858511105c

[6] https://www.prnewswire.com/news-releases/blockchain-credit-bureau-micromoney-looks-forward-to-raising-30-million-during-the-october-crowdsale-300534988.html

[7] https://www.businesswire.com/news/home/20170913006131/en/New-Pave%E2%80%99s-Decentralized-Global-Credit-Profile-GCP-Unlocks-Access-to-Credit-for-Millions-of-Americans.

[8] Hassija, Vikas, et al. "Secure Lending: Blockchain and Prospect Theory-Based Decentralized Credit Scoring Model." *IEEE Transactions on Network Science and Engineering* 7.4 (2020): 2566-2575.

[9] Lee, Tian-Shyug, et al. "Credit scoring using the hybrid neural discriminant technique." *Expert Systems with applications* 23.3 (2002): 245-254.

[10] https://hyperledger-fabric.readthedocs.io/en/release-2.0/txflow.html

[11] https://medium.com/@touqeershah32/hyperledger-fabric-2-0-bank-simple-e8d4c72acafc