MAKERERE UNIVERSITY

COLLEGE OF COMPUTING & INFORMATION SCIENCES (COCIS)

COMMUNITY FUND ADVISOR (CFA)

A Software Design Document Submitted to the School of Computing and Informatics Technology in Partial Fulfillment of the

Requirements for the Award of the Degree of Bachelor of

Science in Software Engineering of Makerere University.

By

BSE 22-2

Supervisor

Dr.Mwotil Alex

Department of Networks

School of Computing and Informatics Technology, Makerere University.

Date: 19/06/2022

Declaration

We, group BSE 22-2, hereby declare that the work presented is original and has never been submitted for an award to any university or institution of higher learning.

#	NAME	STUDENT NO.	REGISTRATION NO.
1	NAJJUMA DIANA	1800723408	18/U/23408/EVE
2	KIKONYOGO KENNETH TREY	1800742094	18/U/42094/PS
3	KALUNGI EDWARD	1800723419	18/U/23419/EVE
4	MIREMBE MARIAM HAVILLAH	1800723410	18/U/23410/EVE

Table of Content

List of Figures

List of Tables

1. INTRODUCTION

1.1 Purpose

The purpose of this software design document is to show how the CFA software system will be structured to satisfy the requirements. The document specifies the architecture and system design of the CFA. The CFA system will be used for automatic fund allocation, portfolio rebalancing, loan optimization and information management of Village Savings and Loan Associations(VSLAs). The system shall be cloud based, and shall be used by the Trade Representation Officer(TRO) and the VSLAs. The intended audience for the document includes; system developers, system testers, project managers, system maintainers and Kampala Capital City Authority (KCCA) staff.

1.2 Scope

The system shall provide a secure and responsive online version of the CFA. The project will cover features including robo-advisory, intelligent portfolios, loan optimization and access control through user roles.

The objectives of the project are as follows:

1. To design a fund allocation algorithm that uses respective user data to make automatic decisions.

2. To design an algorithm that builds a portfolio for each of the members based on their provided personal information, balances, savings and credits.

3. To implement a loan module that stores information about loans and issues financial cards to creditors.

4. To implement user role-based access to the system data in order to enforce the principle of least privilege.

5. To design a two-factor authentication mechanism to make it harder for identity theft.

The system shall offer the following benefits to the TRO:

1. TRO will use the system to track transactions performed on their data in order to establish the source of errors as a result of data input and modification.

2. The data shall be protected from malicious people since it will be encrypted and any authorized person who gains access to it will not be able to make sense out of it.

The system will allow only authorized access to data and so this will protect data from being accessed by untrusted users.

4. The system shall allow only authorized users to access data, which will protect data against any untrusted modifications.

5. The system will log and keep track of user activities to ensure data integrity. In case of any changes to the data, the log will keep the Who? What? When? And Why?

1.3 Overview

The document has six sections. The first section is an introduction, which contains the purpose of the document, scope, reference material, definitions and acronyms. The second section gives the system overview, which describes the system being proposed. Section three is the system architecture which has the architectural design and decomposition description of the system. Section four is the data design, which has data description, dictionary and component design of the proposed system. Section five gives the human interface design which contains an overview of the human interfaces, screen images, screen objects and actions of the system intended to be developed. Section six is the requirements matrix which is used to trace system requirements.

1.4 Reference Material

Forbes "Best Robo-advisors Of June 2022," 2022 [Online] Available: https://www.forbes.com/advisor/investing/best-robo-advisors/#:~:text=Most%20robo%2Dadvisors%20have%20low,help%20clients%20with%20investment%20planning [Accessed: 19-June-2022].

1.5 Definitions and Acronyms

SDD: Software Design Document

KCCA: Kampala Capital City Authority

FinTech: Financial Technology

CFA: Community Fund Advisor

VSLA: Village Savings and Loan Association

2. SYSTEM OVERVIEW

CFA is a Uganda based FinTech that will allow community members save, access credit in their respective VSLAs and also allow tracking, assessment and monitoring from KCCA division authorities. CFA will have four primary users namely TRO, Chairman, Ordinary member, administrator and super administrator with distinguished roles.

The CFA will be designed so as to replace the existing manual system(s) that basically do book-keeping and storage of hard copy files at insecure places and moved whenever required. The TRO had no clear information to base on when recommending a group for the fund. With all these needs, the main purpose is to improve the efficiency of the existing manual process.

The proposed system will comprise of two main modules namely the VSLA management information system, and the TRO's Administrative system. The purpose of the VSLA Management Information system is to store member portfolios, group reports, group activity details, and finances(Loans, savings withdraws). On the other hand, the purpose of the TRO's Administrative system is to carry out reporting and monitoring in order to provide valuable and timely decision support to the TRO for effective actions.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design

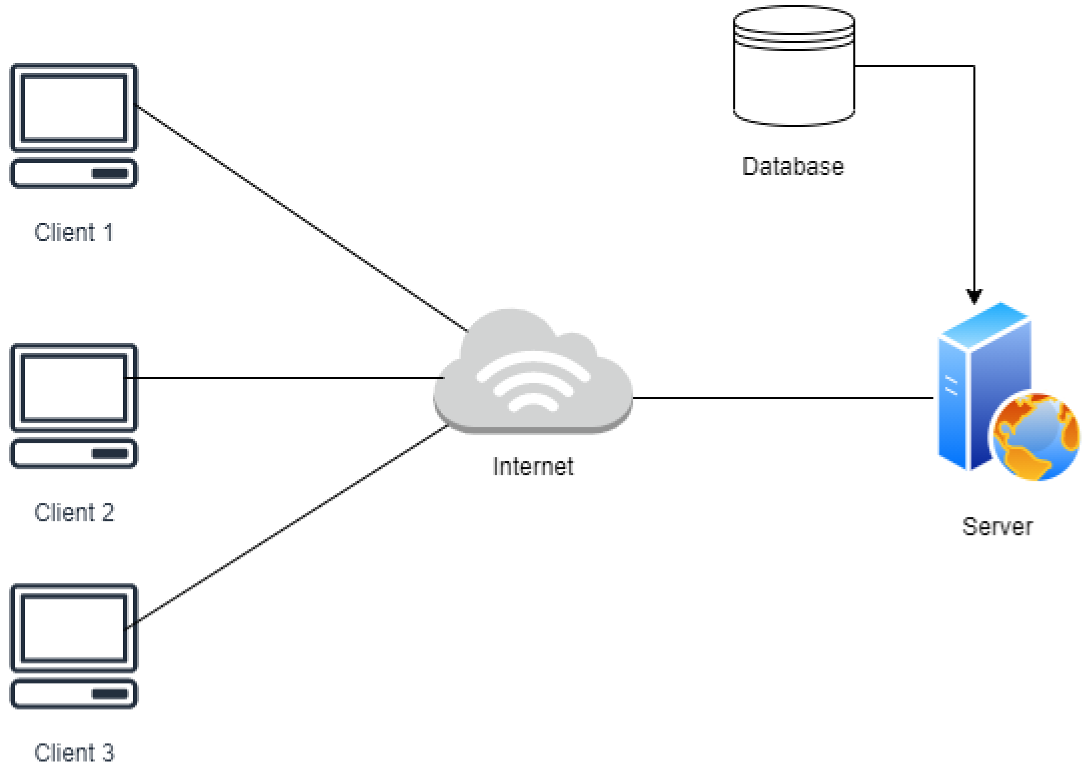


Figure 1: Client Server Architecture for the CFAsystem.

Each user, through a client CFA application sends a request to its instance of the local database server. The server handles those requests and sends a response back to the client. When a client is connected to the internet, the data on the local database server is directly synced with the data on the remote main database server. Otherwise, the client has to wait until it gains Internet connection to sync with the remote database.

3.2 Decomposition Description

The system shall be composed of the following modules.

TRO

Automatic Fund Allocator: -

The system handles re-balancing of the VSLA profiles all by itself when it deems fit without any operational hassles and paperwork from the TRO's end.

VSLAs Management System: -

The system will enable collection and storage of VSLA information from the time of registration to editing of that information.

Regulatory Reporting module: - The system will be configured to have all the

quantitative and qualitative Reports that can be accessed anytime from the system.

VSLAs

Intelligent Portfolio: -

Monitors your portfolio daily and automatically rebalances it as needed. It does this based on the personal information provided by the member.

Loans Module: -

The system will support loan processing in areas such as Loan Issues, Disbursal, Recovery and Follow up.

Savings module: The systems will enable the members to keep track of their savings history.

User access control module: The system will authenticate User so that only authorized users access the system.

Data protection module: The system will validate, encrypt and decrypt the loan data.

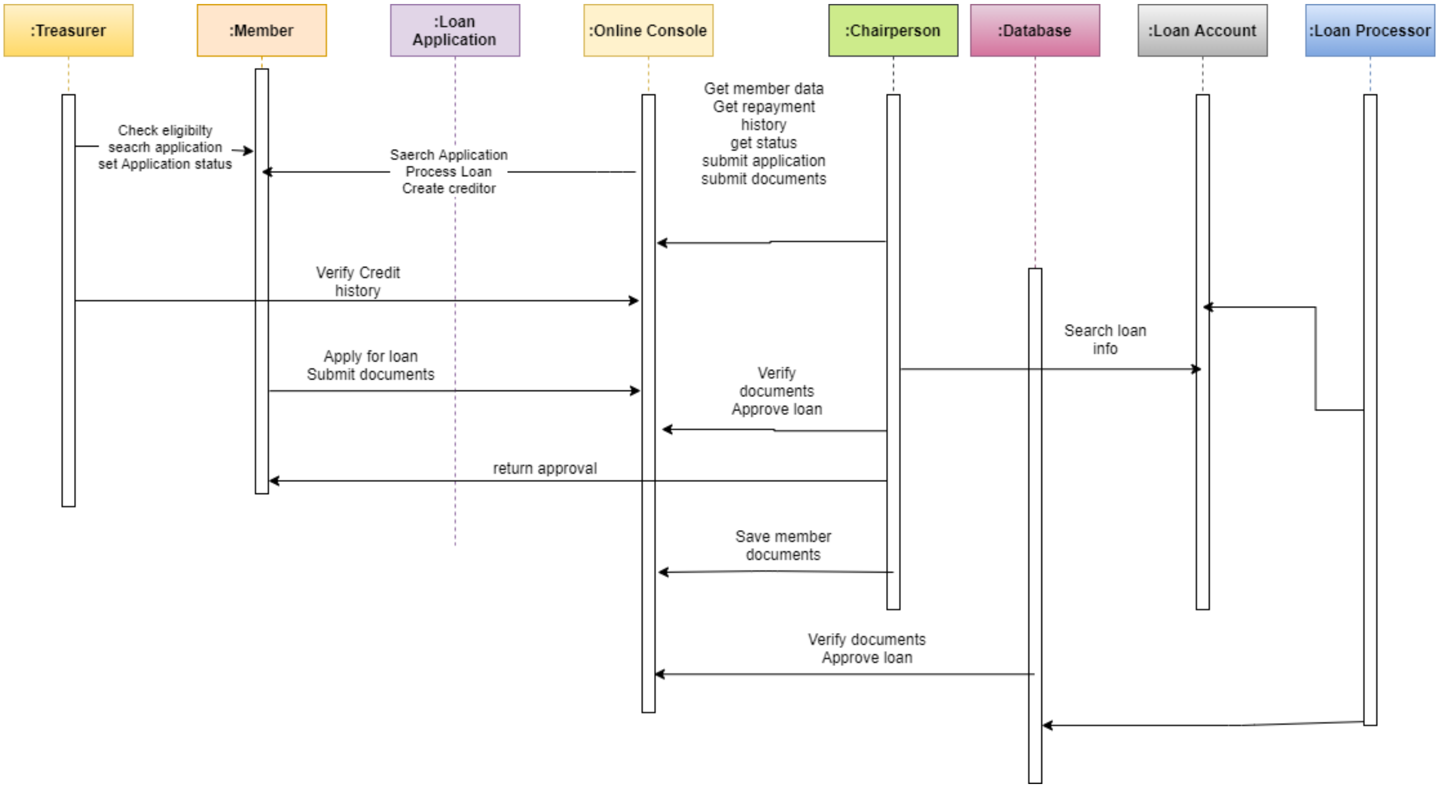


Figure 2 : Sequence diagram for the loan module

3.3 Design Rationale

Our design is solely based on the client-server architecture because its centralized architecture helps make it easier to protect data with access controls that are enforced by security policies.

The different users (chairperson,administrator,TRO) (remote processors) request and receive service from a centralized server (host computer) and the user's interface then displays the results the server returns.

4. DATA DESIGN

4.1 Data Description

Since the CFA is an information-based system, the users of the system are expected to create, manipulate, update and delete data. This therefore gives a description of the data structures of the system.

Table 1: List of entities and attributes.

Entity	Description	Attributes(s)
User	A user is any person involved in any activity related to the CFA.	● id (primary key) ● firstName ● lastName ● email ● phone_number ● pin ● role_name (foreign key)
Role	Users on the system will have different roles for example TRO, Chairman, member, admin and super-admin.	● id (primary key) ● role_name ● permissions
VSLA	A savings and loan group or association to which members belong. It is led by a chairperson.	● id (primary key) ● user_id ● vsla_name ● capacity

		● Location ● status
Loan	Amount of money borrowed by a member from the VSLA, its purpose and time references.	● id (primary key) ● loan ● amount ● purpose ● date_taken ● date_returned ● interest
Creditor	A person or member who takes a loan from the VSLA and has not yet paid back.	● id (primary key) ● amount ● balance
Card	A financial card issued to each creditor on taking loan so as to keep track of their loan history and status.	● id (primary key) ● qr_code ● creditor_id
Receipt	A financial document showing a member's balances on their account.	● id (primary key) ● receipt ● date
ShareOut	Amount of money each member receives at the end of the year when finally withdrawing their savings.	● id (primary key) ● amount ● rate ● user_id

Described here is the database structure composed of the user, role, vsla loan, creditor, card, receipt and share_out tables. Fields indicated PK are primary keys for each table and fields marked FK are foreign keys.

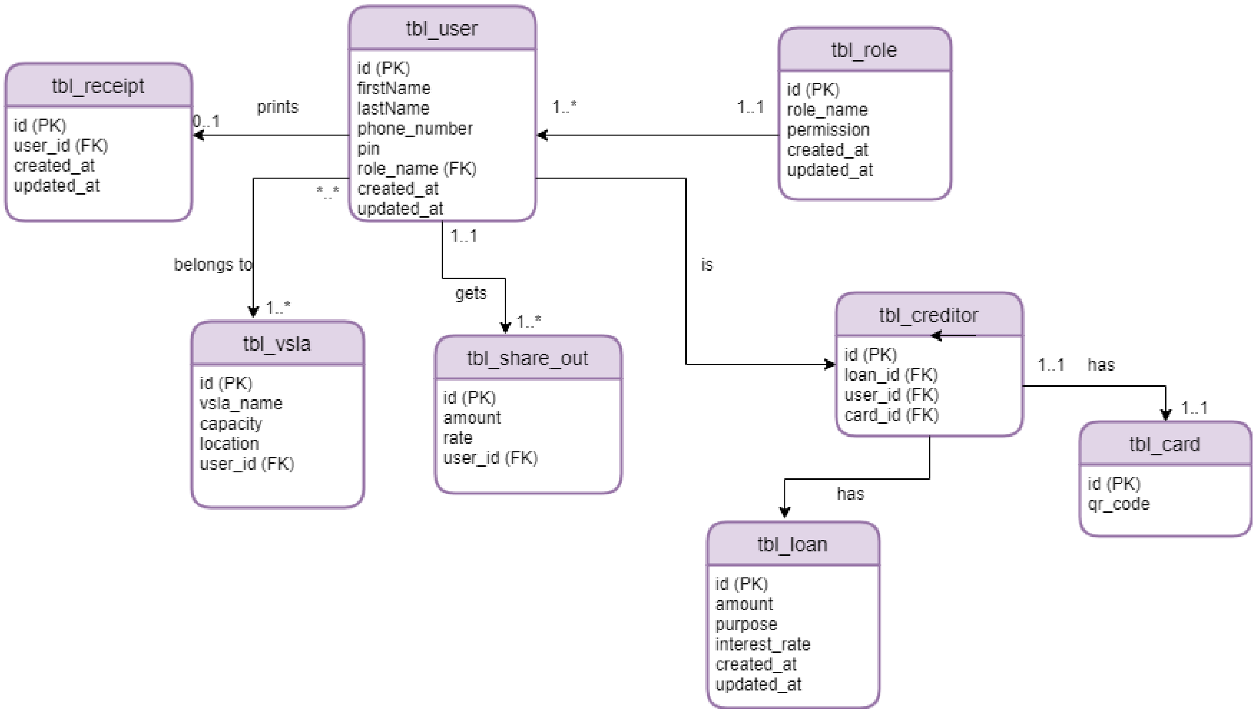


Figure 3: Entity Relation Diagram for the CFA system database

4.2 Data Dictionary

Table 2: User entity for inserting new user

Field Name	Data type	Field size for display	Description
id	integer	11	Unique Identifier for each user
firstName	varchar	25	User's first name
lastName	varchar	25	User's last name
phone_number	integer	15	User's phone number / contact
pin	integer	4	Pin for the user

role_name	varchar	15	Unique identifier for each user role
created_at	datetime		Date the user is created
updated_at	datetime		Date the user is updated

Table 3: Role entity fo the system roles

Field Name	Data type	Field size for display	Description
id	integer	11	Unique identifier for role
role_name	varchar	25	User’s role
permissions			
created_at	datetime		Date the role is created
updated_at	datetime		Date the role is created

Table 4: VSLA entity for inserting new VSLA

Field Name	Data type	Field size for display	Description
id	integer	11	Unique identifier for each VSLA
user_id	integer	11	Unique identifier for each user
vsla_name	varchar	30	Vsla name
capacity	integer	11	Number of members in vsla
location	varchar	20	Where vsla members

			live and meetings happen.
status	varchar	10	Status of the VSLA(Active or Inactive)
created_at	datetime		Date vsla is created
updated_at	datetime		Date vsla is updated

Table 5: Loan entity

Field Name	Data type	Field size for display	Description
id	integer	11	Unique identifier for each loan
amount	integer	11	Amount of money borrowed
purpose	varchar	255	Use of the loan
creditor_id	integer	11	Unique identifier for creditor
status	Enum (‘00’,’11’,’10’,’01’)		Status of loan
created_at	datetime		Date loan is created
updated_at	datetime		Date loan is updated

Table 6: Creditor entity

Field Name	Data type	Field size for display	Description
id	integer	11	Unique identifier for the creditor
user_id	integer	11	Unique identifier for user

loan_id	integer	11	Unique identifier for loan

Table 7: Card entity

Field Name	Data type	Field size for display	Description
id	integer	11	Unique identifier for each card
qr_code	BLOB	16M	QR code for a card

Table 8: Receipt entity

Field Name	Data type	Field size for display	Description
id	integer	11	Unique identifier for each receipt
created_at	datetime		Date receipt is created
updated_at	datetime		Date receipt is updated

Table 9: Share out entity

Field Name	Data type	Field size for display	Description
id	integer	11	Unique identifier for each share out
amount	integer	11	Amount of share out
rate	integer	11	Interest rate of share out
user_id	integer	11	Unique Identifier for user

5. COMPONENT DESIGN

This section provides the description of the components that constitute the architecture of the Community Fund Advisory (CFA) system. Sequence diagrams are used to describe the flow of control and data for each component.

5.1. Automatic Fund Allocator component.

Pseudo code;

```
Checks how many VSLAs are registered in the system
Checks the savings potential in monetary terms on each VSLA's account
Checks the period spent for each VSLA in service //active or deactive
Compares current statistical data about each VSLA
    Assigns funds to a VSLA with a leading saving and active potential.
```

5.2. Savings Module component

Pseudo code;

```
Check VSLA under which a member is registered.
Check for member account
   Input member ID
```

```
    Input account Number
    Retrieve Member saving details
    Display details
```

5.3. Loan Management component

Pseudo code;

```
Check if a member is registered under a certain VSLA
If a member is registered
      Check balance on their VSLA account.
           Check if the balance on their account meets the minimum amount on which a loan can be disbursed
           Disburse loan to a member
    Save all information in the database
ELSE
     Member should be advised to make some savings to qualify for a loan.
ELSE
    Redirect member to register under a VSLA
```

5.4. VSLA management module.

Pseudo code;

```
Check all registered VSLAs in the system.
If VSLA is not registered in the system
    Register VSLA
    Save the details to a database
ELSE
   Redirect User to register VSLA
IF VSLA is registered in the system
    Edit information about VSLA
   Save information in the database
```

5.5. Data encryption component

Pseudo code;

```
While there is a new action performed on the data.
     Read plain text x
     Apply that advanced encryption algorithm to x to check entry validity to x'.
          Save the x' to the database
ELSE
  Redirect user to login interface.
```

5.6. User level data access component.

Pseudo code;

```
    If a user id logged in
    If user role is x
        Display specific    dashboard(sub-menu) for user role x
ELSE
 Redirect the user to the login interface.
```

5.7. Tracking the typing patterns of the user component.

Pseudo code;

```
   If user is logged in and performs an    action on that data
        Types in the text field
        System monitors the typing speed of the user
        They system retrieves the previous typing patterns
        The system compares the current and retrieved typing speed patterns
ELSE
```

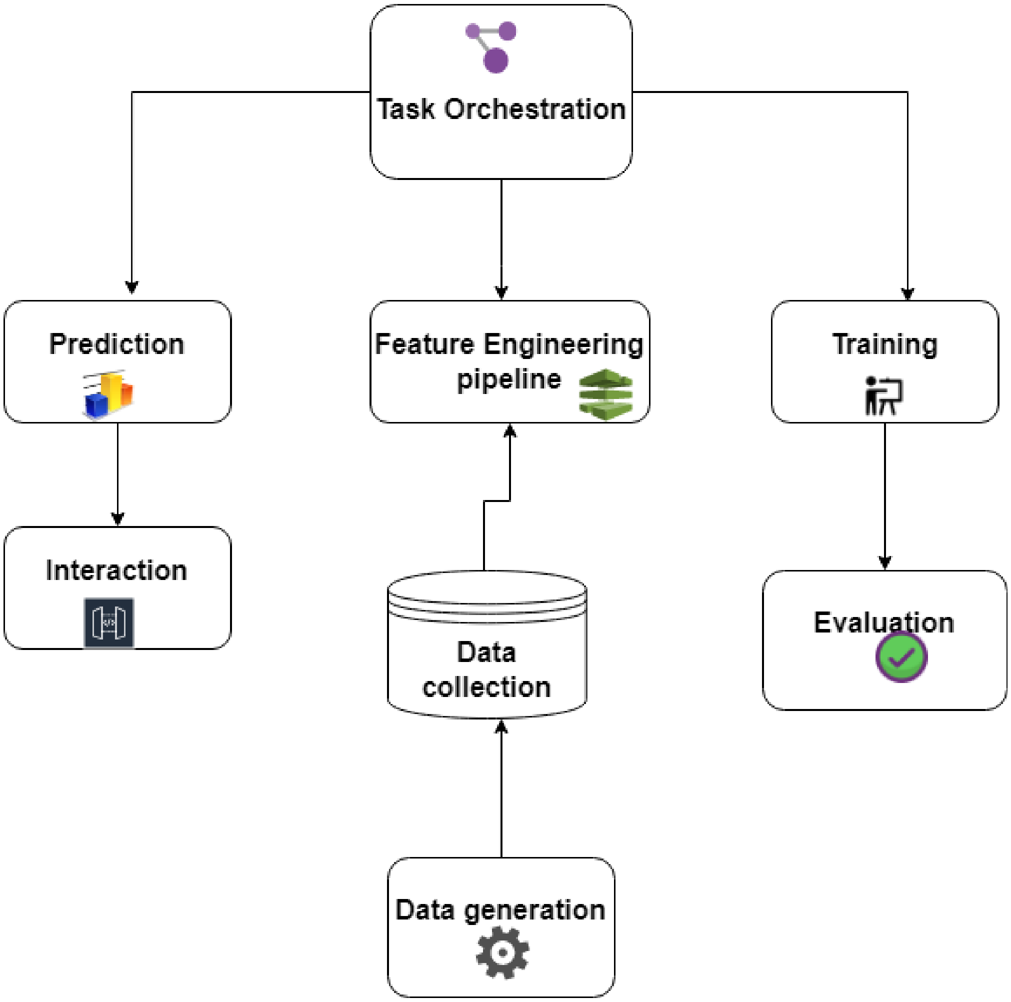
System raises a warning to the manager data and marks the user data as suspected.

ELSE

Redirect the user to the login interface.

5.8 Machine learning module

Diagrammatic representation of the Community Fund Advisor component diagram.



A brief description of each of the main components for the machine learning model.

1. **Data Generation**: Our Community Fund Advisor lives off data. That data comes from the Trade Representation Officer's module of the system.
2. **Feature Engineering Pipeline**: CFA's algorithm can't make sense of raw data. So, as a team, we wrote a script that selects, transforms/cleans, combines, and otherwise prepares the data so the algorithm can find useful patterns.
3. **Training**: This is where the magic happens. We apply the algorithm, and it learns patterns from the data. Then it uses these patterns to perform the intended task; advising on which Village Savings and Loan Association should receive the fund in a given financial year.
4. **Evaluation**: Here, we need to carefully test how well our algorithm performs on data it hasn't seen before (during training). This ensures that the prediction model doesn't work well only on "seen" data, but not in real-world settings.
5. **Task Orchestration**: Feature engineering, training, and prediction all need to be scheduled on our computer infrastructure (Google Cloud Platform) – usually with non-trivial interdependence. So we need to reliably orchestrate our tasks.
6. **Prediction**: This is our moneymaker. We use the model we've trained to perform new tasks and solve the TRO's main problem – and this means making a prediction.
7. **Infrastructure**: Even in the age of the cloud, **CFA** has to live and be served somewhere. We are using GCP for setup and maintenance.
8. **Authentication**: This keeps our model secure and makes sure only those who have permission can use it.
9. **Interaction**: We need some way to interact with our model and give it problems to solve. This is in the form of an API, and a user interface.
10. **Monitoring**: We need to regularly check our model's performance. This usually involves periodically generating a report or showing performance history on the dashboard.

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface

The system will be a role-based system where the overall administrator is the Data manager. Admin will be responsible for adding new system users and their roles as shown in figure.

6.2 Screen Images

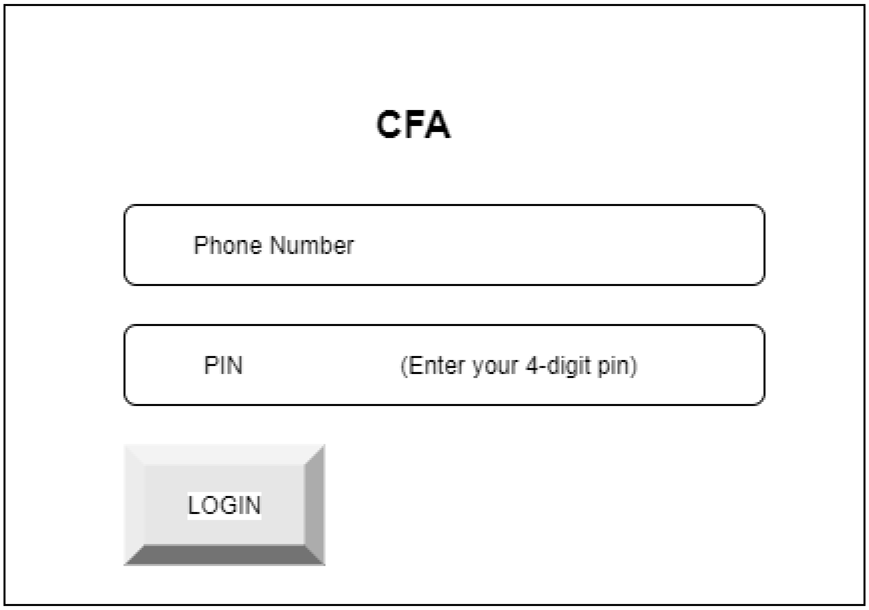


Figure 6.2 1 Log into the system

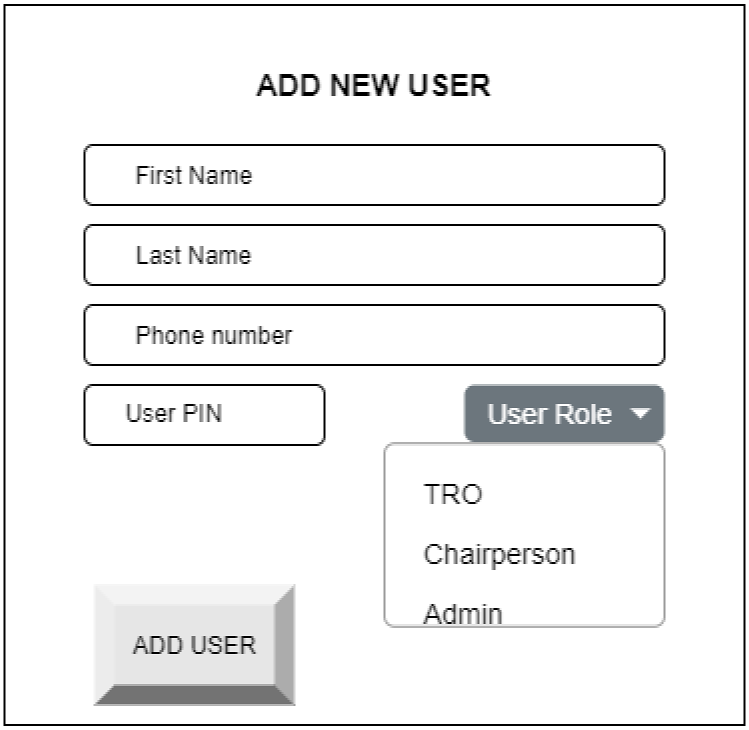
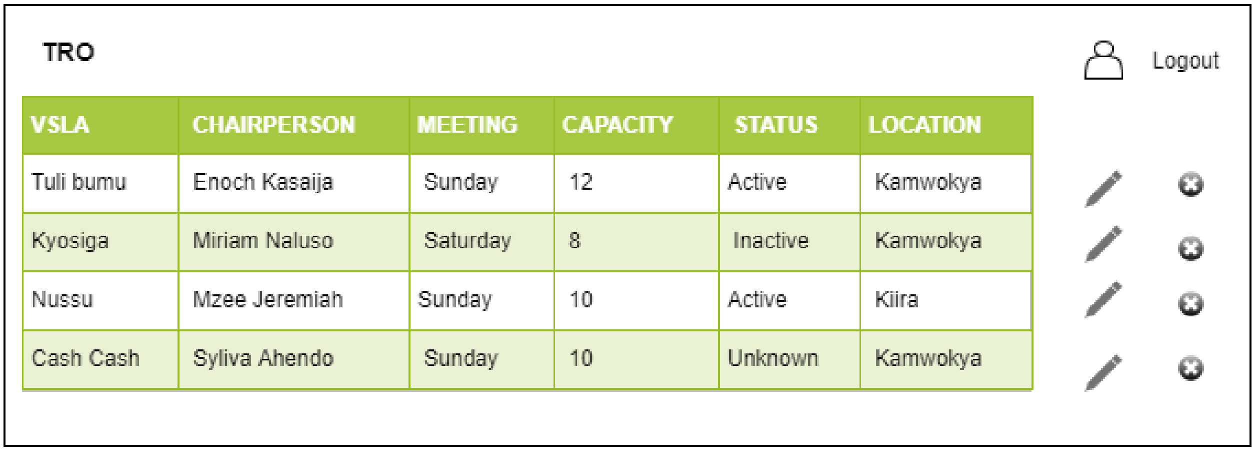


Figure 6.2 2: Create or add new user with role in the system.



VSLA	CHAIRPERSON	MEETING	CAPACITY	STATUS	LOCATION
Tuli bumu	Enoch Kasaija	Sunday	12	Active	Kamwokya
Kyosiga	Miriam Naluso	Saturday	8	Inactive	Kamwokya
Nussu	Mzee Jeremiah	Sunday	10	Active	Kiira
Cash Cash	Syliva Ahendo	Sunday	10	Unknown	Kamwokya

Figure 6.2 3: TRO's dashboard for managing the various VSLAs.

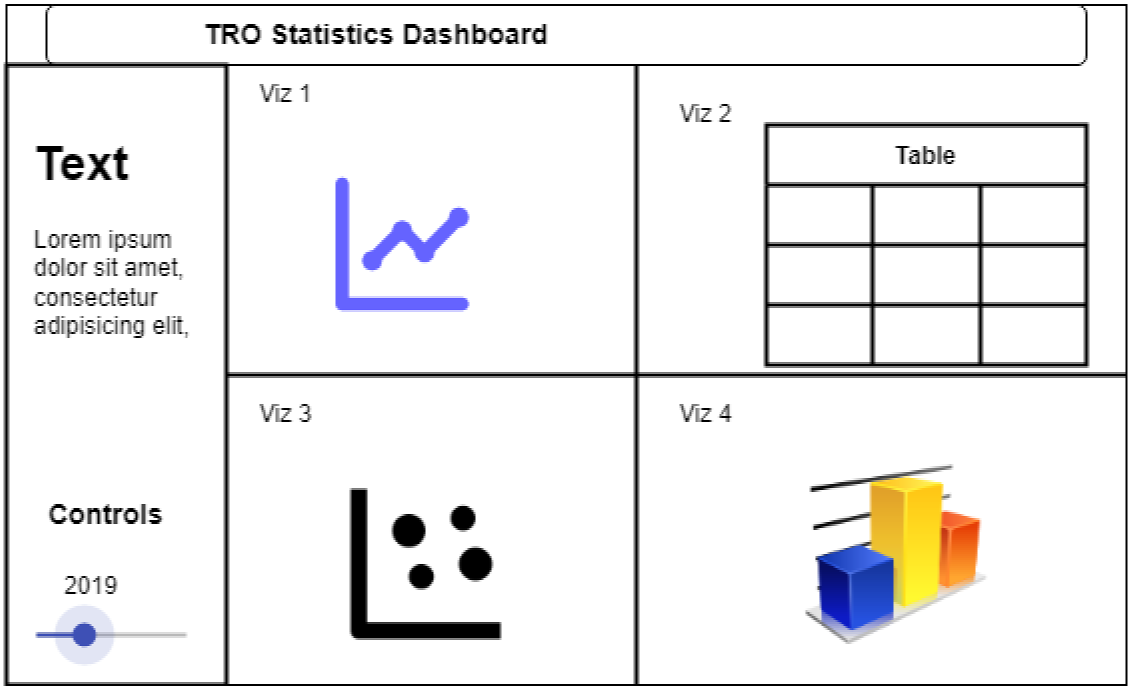


Figure 6.2.4: TRO's dashboard for monitoring the VSLA performance.

6.3 Screen Objects and Actions

Screen Object	Action	What happens on performing the action
Phone number	Enter the phone number for the user being created	User phone number stored
User pin	Enter and create the pin the user will use to access the account	The user pin stored
User role	The role played by the	Display the user role that

	user in the system. For example, Admin	performed the action
Pin	Enter the pin to access user account	If the pin matches the stored -pin of the users account in the database , access is granted and vice versa.
First name	User name of the person performing the action	Display user name of the person who performed the action
Last name	User name of the person performing the action	Display user name of the person who performed the action

7. REQUIREMENTS MATRIX

Table: A requirement Traceability Matrix

Business Use Case	Functional Requirements		Priority	Test Case	
	Requirement ID	Description		ID	Description
Capture user details	REQ-1	Authenticate all users before they access the system	High	TC01	Verify if user is authenticated as they access the system
	REQ-2	Logging of all user activities	High	TC02	Verify if all user activities are logged
	REQ-3	Authenticate all	High	TC03	Verify if the admin

		admins and super admins with one-time pin code			and super admin provide correct codes
Capture savings details	REQ-1	Capture member`s savings through USSD code	High	TC04	Verify that the system is able to capture savings to the database through the USSD code
	REQ-2	Capture member`s savings through the mobile app	High	TC05	Verify that the system is able to capture member savings using the mobile cloud based app
	REQ-3	Capture member`s savings through the web app	High	TC06	Verify that the system retrieves and captures savings on the web to and fro the database
Loan Advisory	REQ-1	Assess member`s loan eligibility	Medium	TC07	Verify that the system can determine which member is eligible
	REQ-2	Grant member specific loan amount	High	TC08	Verify that the system provides the given range or specific amount a member deserves
	REQ-3	Track installment payback of a given member loan	High	TC09	Verify that the system updates member credit history after a given installment is paid

TRO assessment of VSLA	REQ-1	Assess VSLA savings and loan performance	High	TC10	Verify that the system is able to track VSLA savings and loan performance in a given month, quarter or year
	REQ-2	Rate VSLA performance in the specific division	Medium	TC11	Verify that the system provides in order the VSLAs performance in the entire division
	REQ-3	Evaluate financial support or funds need	High	TC12	Verify that the system provides poor performing VSLAs in the division
Encryption of user data	REQ-1	Allow user enter raw data	Medium	TC13	Verify if user provides correct input data
	REQ-2	Encrypt data entered	High	TC14	Verify that the system is able to encrypt data provided by the user
	REQ-3	Save encrypted data to the data	High	TC15	Verify that the encrypted data is saved appropriately