**THE FAMILY SACCO SYSTEM**

**SUPERVISOR**

**NSABAGWA MARY**

**GROUP 9 DEVELOPMENT TEAM MEMBERS**

SSEBIRIMBI NICHOLAS HENRY (GROUP LEADER)

KUNYA NATHERN

MUGARURA ALLAN

BUSUULWA MARTIN

**TABLE OF CONTENTS**

1. INTRODUCTION 4

1.1 Purpose 4

1.2 Scope 4

1.2.1 Goals 4

1.2.2 Benefits 4

1.3 Overview 4

1.3.1 Introduction 5

1.3.2 System overview 5

1.3.3 System architecture 5

1.3.4 Data design 5

1.3.5 Component design 5

1.3.6 Human interface design 5

1. System Overview 6
2. System architecture 7

3.1 Architectural design 7

3.2 Decomposition description. 10

3.3 DESIGN RATIONALE 12

1. Data design. 13

4.1 Data description 13

4.1.1 Conceptual design. 14

4.1.2 Physical design. 15

Table 1 : Loan 15

Table 2 : Member 15

Table 3 : Loan\_repayment 15

Table 4 : Contribution 16

Table 5 : Investment 16

Table 6 : Business 16

Table 7 : Benefit 16

Table 8 : Administrator 17

Table 9 : Business\_follow up 17

4.1.3 Logical design 17

4.2 Data dictionary. 19

1. Component design 21
2. HUMAN INTERFACE DESIGN 23

6.1 Overview of the user interface 23

6.2 Screen images 24

6.3 Screen Objects and Actions 28

**SOFTWARE DESIGN DOCUMENT**

# INTRODUCTION

## 1.1 Purpose

The purpose of the software design document is to provide the description and track the necessary information required to effectively define architecture and system design in order to give the development team guidance on the architecture of the family SACCO system to be developed.

Its intended audience are the development team and supervisors. The family SACCO members will be able to view the user interface design that is contained in the SDD.

## 1.2 Scope

The family SACCO system is composed of two primary components;

A client side application which is a command line socket program that will allow users to submit their contributions to a temporary file on a **C** server and

A server side application that consists of a C command line server and a PHP web based interface that will read information written to a file and cuts it out and saves it in the database which will only be accessed and managed by the system administrator

### 1.2.1 Goals

**To manage and store the family SACCO data such as details of all the family SACCO members, their contributions and benefits**

**To automate some of the family SACCO services such as submitting contributions, checking loan status and requesting for the loans**

### 1.2.2 Benefits

**The system shall allow users to make contributions at any time anywhere which makes services more readily available and easily accessed**

**The system shall have a secure central repository of the family SACCO information which is a database**

**The system shall emphasize security measures such as authorization and authentication where users are assigned passwords and usernames to login into the system to ensure that the family SACCO data is not vulnerable to attacks**

**The system shall be able to manage the shares of all SACCO members**

**The system shall generate and run reports, and allow shareholders to view information.**

## 1.3 Overview

**The SDD is composed of eight sections with other subsections so as to give a clear understanding of the project. These sections include the following;**

### 1.3.1 Introduction

**This gives the general understanding of the purpose of the SDD and a brief description of the design along with the scope, intended goals, objectives and benefits of the software to the SACCO members. The overview, definitions and acronyms of the words and phrases used in the SDD, and the reference materials used to get additional information of the software design.**

### 1.3.2 System overview

**This section contains the general functions, operations and capabilities of the system to be developed and an understanding of how the users interact with the system to achieve their goals**

### 1.3.3 System architecture

**This shows the type of architecture design considered in the designing of the system and the modular program structure with a description of how the modules relate to achieve a complete functionality of the system.** **The main purpose is to gain a general understanding of how and why the system will be decomposed, and how the individual parts will work together. This section of the SDD also has diagrams showing the major subsystems and data repositories and their interconnections.  
This section also shows the decomposition description of the subsystems such as the data flow diagrams and the function decomposition diagrams.  
The design rational that expresses reasons why the architectural design used was considered and why other types weren’t considered.**

### 1.3.4 Data design

This describes how the information domain of the SACCO system is transformed into data structures and how the major data or system entities are store, processed and organized. The data dictionary is also in this section describing the system entities or data along with their types and descriptions, functions and function parameters alphabetically

### 1.3.5 Component design

**This section provides a summary of the algorithm for each function or component in form of a pseudocode**

### 1.3.6 Human interface design

**This section describes the system functionality from the user’s perspective. It explains how the user interacts with the system to complete all the expected features and how the system will give feedback to the user.   
Screen images to show the user interface, screen objects and actions are also in this section**

# System Overview

The SACCO members will be able to submit their contributions, loan repayments, loan requests and business ideas inform of commands using a command line interface through a C socket client program that will connect the command line interface to the C server.

The client commands will be received by a C server stored in a temporary file. The system administrator will be able to interact with the PHP web interface which will read and cut out information written to the temporary file and display on the interface for the system administrator to verify the submission records by denying and accepting them.

The system will automatically store the records that are accepted or denied by the system administrator into the database

For the case of loan requests, the system shall be able to check whether the member status is regular and the requested amount is less than half the member’s contribution. If the member status is not regular and the loan amount is more than half the member contribution, the system shall reject the loan request automatically and if otherwise the system administrator will accept the loan requests and then stored into the database.

Assumption

In case the SACCO member has not paid the loan amount with in a period of one year, His or Her interest will be increased by 1% of the amount in loans

The system administrator will be able to enter all the member details and investment details into the system thereby assigning all the SACCO members with their usernames and passwords.

The system administrator will enter the business returns into the system and the system will automatically distribute all the business benefits to the SACCO members based on their percentage shares

The system shall automatically generate reports by getting the information from the database and logically arranging it for all users to view.

The SACCO members and the system administrator shall first login on the website using their passwords and usernames in order to view reports which are displayed on the website. Reports such as the names of all members, the amount in loans, member’s benefits, and the loan repayment details.

# System architecture

## 3.1 Architectural design



Figure 3.1.1 architectural design

The C command line program will only help SACCO members to submit contributions, loan requests, and business ideas. The command line interface will use a socket program to send commands the C server

The C server will write to the file that stores the client’s submissions temporarily until they are cut, read and displayed to the PHP web based interface on the browser to be verified by the system administrator and then saved into the MYSQL database on a Wamp/Xampp server

The browser will host the PHP web based interface that gets information stored in the database via a network or internet and display them to the users. Both the SACCO member and the system administrator will be able to see reports using the browser



**Figure3.1.2 : modular design diagram**

## 3.2 Decomposition description.



figure 3.2.1 : Context diagram of the family SACCO system

.



figure 3.2.2 : LEVEL O diagram of the family SACCO system.

## 3.3 DESIGN RATIONALE

The scalability of the family SACCO system is guaranteed when using an internet based system architecture since it can handle the increasing transaction volumes of the family SACCO and can also handle hundreds of users that can be added to the SACCO.

The internet based system architecture provides process management services such as process monitoring to the system administrator like adding new members to the system and accepting or denying members contributions.

An Internet based system architecture gives the SACCO members the ability to access the system at anytime and anywhere so long as he or she is connected to the internet so as to view the reports generated by the system. It also gives all its users a fast experience when interacting with the system since it has the processing logic

An internet based architecture has a central repository of all the SACCO’ data which is a MYSQL database making sure that all records are secure and can be accessed by only the system administrator

The processing logic in an internet based system architecture is between the user interface (client) and the data management (server) components where business logic and rules are executed and can accommodate hundreds of users as compared to only 100 users with the two tier architecture by providing functions such as queuing, application execution, and database staging.

# ****Data design.****

## 4.1 Data description

The data entered in the system is stored in the database as tables , each table representing an entity having attributes of the entity where the entered data is stored.

The major entities include member, loan repayment, loan , contribution,benefits, administrator, business, investment.

Each member submits a contribution and a regular member is on who made contribution for the last 6 months.

Each member may request for a loan.

Any member can suggest a business

Benefits are shared among many members

A business generates profits

Member have to make loan repayment

A contribution is verified by an administrator, and an administrator may verify many contributions

An administrator adds one or many members and he also add investment ideas.

### 4.1.1 Conceptual design.



figure 4.1 : conceptual design / ERD of the family SACCO system.

### 4.1.2 Physical design.

#### Table 1 : Loan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| loanId | Loan identification | int | 5 | Not null | Primary key |
| loanDate | Loan date | Varchar | 10 |  |  |
| loanStatus | Loan status | Varchar | 12 |  |  |
| paymentPeriod | Payment period | Varchar | 10 |  |  |
| loanAmount | Loan amount | Big int | 15 |  |  |
| Interest | Interest | Big int | 15 |  |  |

#### Table 2 : Member

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| memberId | Member identification | int | 5 | Not null | Primary key |
| memberName | Member name | Varchar | 30 | Not null |  |
| Password | Password | Varchar | 25 | Not null |  |
| additionDate | Addition date | Varchar | 15 | Not null |  |
| Username | Username | Big int | 25 | Not null |  |

#### Table 3 : Loan\_repayment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| repaymentId | Repayment identification | int | 5 | Not null | Primary key |
| amountPaid | Amount paid | Big int | 15 |  |  |
| payDate | Pay date | Varchar | 10 |  |  |
| memberPaid | Member paid | varchar | 25 |  |  |

#### Table 4 : Contribution

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| reeceiptNo | Receipt number | int | 5 | Not null | Primary key |
| memberId | Member identification | int | 5 | Not null |  |
| contributionDate | contribution date | Varchar | 10 | Not null |  |
| contributionAmount | contribution amount | Big int | 10 | Not null |  |

#### Table 5 : Investment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| investimateId | Investment identification | int | 5 | Not null | Primary key |
| memberId | Member identification | int | 5 | Not null |  |
| investmentDate | Investment date | varchar | 15 | Not null |  |
| initialInvestment | Initial investment | Big int | 10 | Not null |  |

#### Table 6 : Business

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| businessId | Business identification | int | 5 | Not null | Primary key |
| businessName | Business name | varchar | 15 | Not null |  |
| memberSuggested | Member suggested | varchar | 30 | Not null |  |
| business\_idea | Business idea | varchar | 30 | Not null |  |

#### Table 7 : Benefit

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| benefitId | Benefit id | int | 5 | Not null | Primary key |
| Benefitamount | Benefit amount | Big int | 15 | Not null |  |

#### Table 8 : Administrator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| adminId | Administrator identificstion | int | 5 | Not null | Primary key |
| adminUsername | Administrator user name | varchar | 15 | Not null |  |
| adminName | Administrator name | Varchar | 20 | Not null |  |
| adminPassword | Administrator password | int | 10 | Not null |  |

#### Table 9 : Business\_followup

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute | Domain name | Data type | Length | Constraint | Keys |
| businessId | Business identification | int | 5 | Not null | Primary key |
| businessOutcome | Business outcome | Big int | 15 | Not null |  |

### 4.1.3 Logical design

Loan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| loanId | loanAmount | loanDate | loanStatus | interest | adminId |

Loan\_repayment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| repaymentId | memberId | amountPaid | payDate | memberPaid |

Member

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| memberId | memberName | Username | password | additionDate | loanId | adminId | businessId |

Investment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| investmentId | initialInvestment | InvestmentDate | adminId | memberId |

Business

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| businessId | businessIdea | businessName | businessReturns | memberSuggested | investimentId |

Administrator

|  |  |  |  |
| --- | --- | --- | --- |
| adminId | adminName | adminUsername | adminPassword |

Benefits

|  |  |  |
| --- | --- | --- |
| benefitAmount | businessId | benefitId |

Contribution

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| receiptNo | personName | contributionDate | contributionAmount | adminId | memberId |

Member\_benefit

|  |  |
| --- | --- |
| memberId | benefitId |

Business\_followup

|  |  |  |
| --- | --- | --- |
| businessId | businessOutcome | businessId |

## 4.2 Data dictionary.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity | Attribute | Data type | constraint | description |
| Benefit | benefitId | int | Primary key | Benefit id, auto generated |
|  | benefitAmount | Big int | Not null | Benefit amount |
| Business | businessId | int | Primary key | Business id, auto generated |
|  | businessName | Varchar | Not null | Name of the business |
|  | memberSuggested | Varchar | Not null | Member who suggested the member |
|  | businessIdea | Varchar | Not null | Business idea |
| Investment | investmentId | Int | Primary key | Investment id, auto generated |
|  | investmentDate | Varchar | Not null | Date the investment was made |
|  | memberId | int | Not null | Member identification |
|  | initialInvestment | Big int | Not null | Initial investment in of business |
| Administrator | adminId | Int | Primary key | Administrator id, auto generated |
|  | adminUsername | Varchar | Not null | Administrator username |
|  | adminName | Varchar | Not null | Administrator name |
|  | adminPassword | varchar | Not null | Administrator password |
| Contribution | receiptNo | Int | Primary | Receipt number of the member contribution |
|  | memberId | int | Not null | Identification of the person who has submitted the contribution |
|  | contributionDate | Varchar | Not null | Date the contribution was submitted |
|  | contributionAmount | Big int | Not null | Amount of the contribution |
| Loan | loanId | Int | primary key | Loan id, auto generated |
|  | loanDate | Varchar |  | Date the member was given the loan |
|  | loanAmount | Big int |  | Amount of loan given to a member |
|  | loanStatus | varchat |  | Loan status showing if the member loan request was denied or accepted |
|  | PaymentPeriod | varchar |  | Period in which the loan is to be paid. |
|  | Interest | Big int |  | Interest a member must return |
| Member | memberId | Int | Primary key | Member id, auto generated |
|  | memberName | Varchar | Not null | Name of the member |
|  | Password | Varchar | Not null | Member password used to log into the system to view reports |
|  | Username | Varchar | Not null | Member username |
| Loan\_repayment | repaymentId | Int | Primary key | Repayment id, auto generated |
|  | amountPaid | Big int |  | Amount which has been repaid so far by the member who was given the loan |
|  | payDate | Varchar |  | Date the repayment was made |
|  | memberPaid | varchar |  | The name of the member who has made the repayment. |
| Business\_followup | businessId | int | Primary key | Business id, auto generated |
|  | businessOutcome | Big int |  | Amount generated from the business. |

# Component design

Submit contribution

Save contribution in a temporary file

Read and display contribution

Contribution state pending

If record contribution is accepted

Cut from temporary file

Store record in database

Else

Cut and delete record

Submit loan request

Save loan request in the temporary file

Read and display loan request record

If record of loan request is accepted

Cut the record from the temporary file

Store the record in the database

If regular member

If loan amount is less than half the contribution amount of the member

Approve loan

Interest is 3% of loan amount

Else

Deny loan

Else

Deny loan

Else

Cut and delete record

Allocation of benefits

Enter business returns.

Determine the difference between business return and initial investment

If existing member

If the difference between the businesses return and initial investment is less than 0

Forego all benefits in the next business

Store difference as loss

Else

Store difference as profit

Add 30% of profits to SACCO savings

Member share is the percentage of what one has contributed divided by the total

Contribution

Add 5% of profit to the member with highest number of shares.

Share 65% of profits amongst members according to their shares.

Else

New members cannot receive benefits

Adding new member

If initial deposit is greater or equal to ¾ of the total contribution of the highest contributor

Add new member

Assign login details to the member

Else

New member not added

Adding investment details

Enter investment details

If initial investment amount is greater than ½ of the available money

Investment idea is rejected

Else

Investment idea is accepted

Reports generation

Update all records

Generate reports

If user logs in

View reports

# HUMAN INTERFACE DESIGN

## 6.1 Overview of the user interface

The member will open the command line and enter contributions, business ideas and loan requests.

The member or system administrator will click on the drop down box in **figure 6.1.1** to select whether he/she want to log in as a member or administrator. The user will then enter the username and password, and click on the login button to access the site. If the username or password entered does not match with those in the system’s database, access will be denied and a message “**Your password or username is incorrect**” will be displayed.

If the user is an Administrator, and password or username are correct, a web page in figure 6.1.2 will be displayed.

The administrator will click on the Add member from the menu bar in *figure* *6.1.3* to add new members in the system and a page in *figure 6.1.4* will be displayed.

The administrator will enter member’s name, username and password, and then click on the register button to add a new member in the system.

If the entered username already exists in the database, a message “**Username already exist**” will be displayed and if the password entered in the confirm password field does not match with that in the password field, a message “**Password does not match**” will be displayed.

The administrator will click on the view files from the menu bar in figure 6.1.3 to view submitted records by the clients from the c server. The system administrator will click on accept or deny button in figure 6.1.5 to either accept the records or deny them.

The administrator shall select what to do from the menu bar in figure 6.1.3 for example viewing reports, entering business ideas, investments, business returns. When the administrator is done, he/she will click on the log out button to leave or exit from the system.

If the accepted user in figure 6.1.2 is a member, a page in figure 6.1.6 will be displayed showing different reporting based on his/her account for example viewing his/her contributions, loan status, loan repayment details, benefits by selecting an option from the menu bar. He/she can also view other reporting by clicking on **view report** in figure 6.1.6 and a page in figure 6.1.7 will be displayed. When the member is done with viewing reports, he/she can click on the **log out** button to leave the site.

## 6.2 Screen images

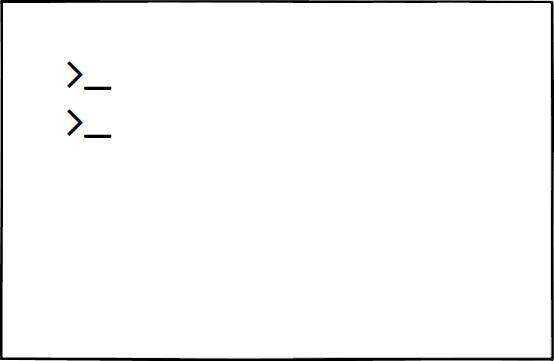


Figure 6.2.1 Command line

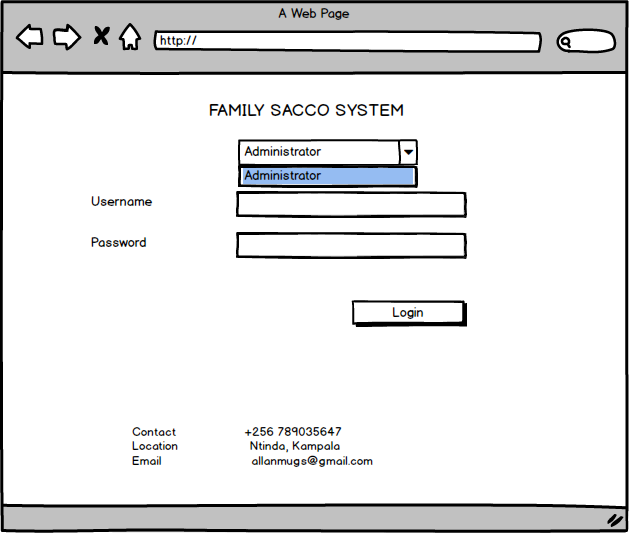


Figure 6.2.2 Log in page

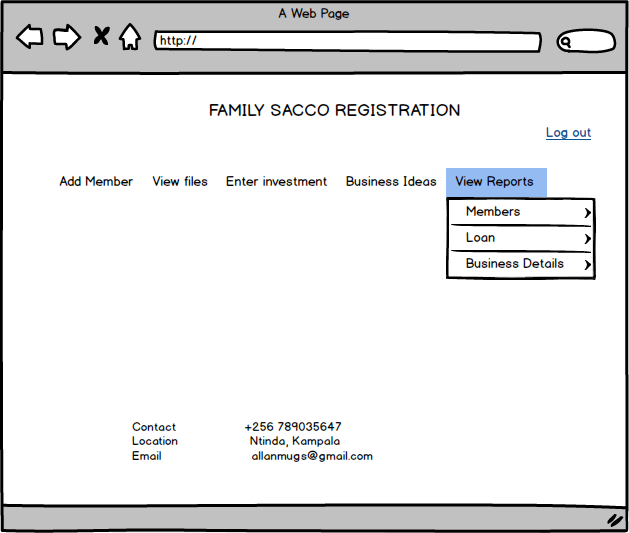


Figure 6.2.3 administrator's page

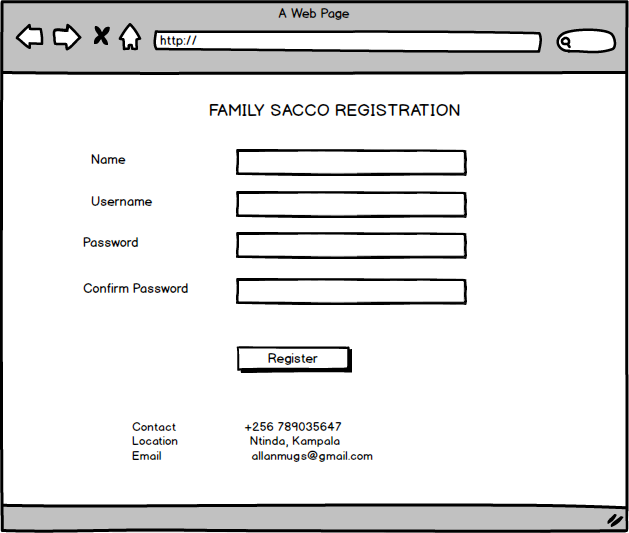


Figure 6.2.4 adding new members

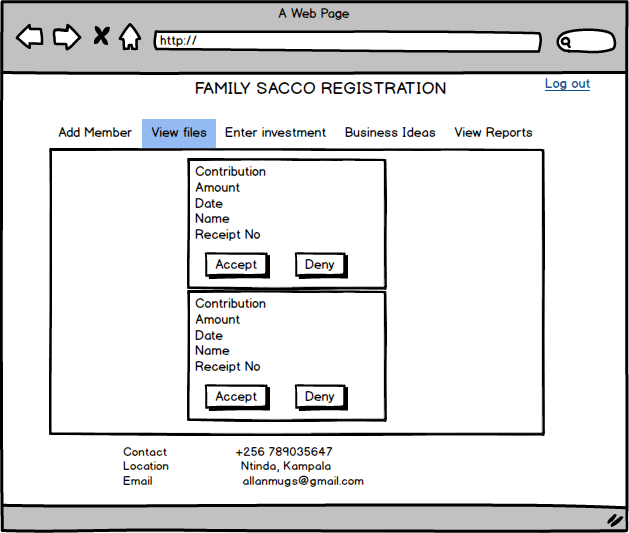


Figure 6.2.5 Reading files from the server

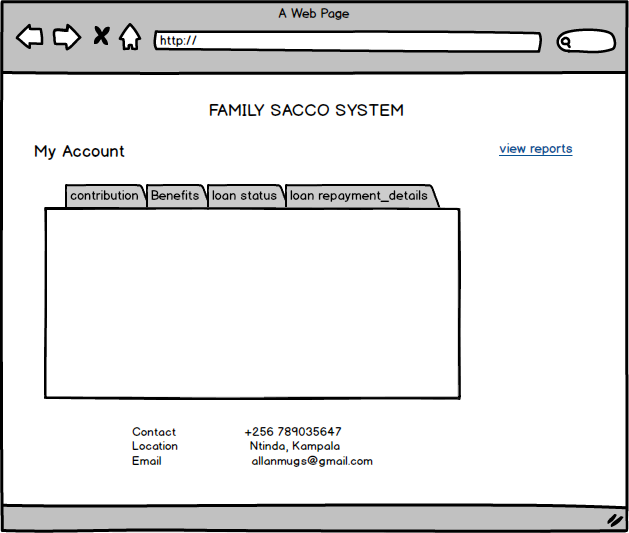


Figure 6.2.6 member account report

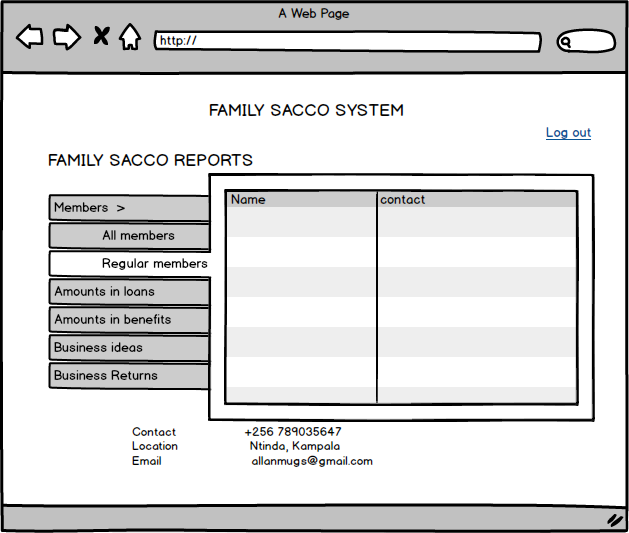


Figure 6.2.7 Other reports

## Screen Objects and Actions

Figure 6.1.2

When Login button is clicked, depending on the user, pages in *figure 6.1.3* and *figure 6.1.6* will be displayed.

Figure 6.1.4

When the Register button is clicked, data filled in the fields will be submitted to the database.

Figure 6.1.4

When the accept or deny buttons are clicked, records will be submitted in the database

When the log out button is clicked, the user at that time will be signed out.