UNITED FRONT FOR TRANSFORMATION(UFT) MEMBER ENROLLMENT SYSTEM

Software Design Document

GROUP G-4

NAME	REG NO.	STUDENT NO.
KALUUMA MAHAD	17/U/1405/EVE	217002461
LUBWAMA ISAAC	17/U/5739/PS	217017396
KINTU ARNOLD	17/U/5188/EVE	217014630
KYANZI HASSAN MUSISI	17/U/19979/EVE	217022320
KASOZI JAFFAR	17/U/4741/EVE	217002416

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to describe the architecture and system design of a member enrollment system of a new political party known as United Front for Transformation (UFT), that is described in the Software requirements document. The UFT member enrollment system shall be designed to enable the agents to register new members of the political party, the administrator to list members enrolled in a given period of time, organized by district, agent, month, to graph displays of hierarchy,to distribute money to each of the agents in the different districts ,to display payment statements of the agent ,upgrade members to become agents and also keep track of the funds as received from the well-wishers and how the funds are distributed among the agents.

1.2 Scope

The member enrollment system is a client-server-based system that runs on both a command line and graphical user interface, the latter being web-based. Its mode of operation is based on a set of functions that manipulate string data specifically, alphabetic and numeric as instructed by the agent, who utilizes the command line interface to issue tasks. The graphical user interface is utilized by the administrator for monitoring purposes. The system is aimed at achieving optimal performance while performing the following functions. Submitting a new member list for instance, the command “Addmember” inserts the member details, the command “Check-status” returns the list of members that have been registered, the command “get_statement” returns the payments of the logged-in users, the command “search” returns record from a file by name or date. At the administrator’s end, he will interact with a web interface to view the members enrolled in a given period of time organized by district,agent,month as selected by the administrator; Display hierarchy, at the top of each is an agent head displayed by district; Display distribution of money to each of the agents in the different districts; Display graphs of variation of funding per month and per period; Displays graphs showing donations by month and well-wisher as selected by the administrator; Display the total number of members; Display graph that shows variation in percentage change in enrollment figures.

The web interface shall also recommend the upgrading of members to agents in case the member recommends more than 10 members.

1.3 Overview

This document has been organized into chapters as described below

Chapter 1: describes the product scope, the purpose of this document, the reference material and abbreviations used with their meanings.

Chapter 2: describes the system overview thus the general description of the functionality, context and design of the UFT member enrollment system.

Chapter 3: describes the system architecture, which comprises the system architecture design, the decomposition of the system and justification for the choice of the design.

Chapter 4: describes the data design of the system, which shows how the information domain of the system is transformed into data structures.

Chapter 5: describes the component design of the system. It gives a functional description of each component in detail.

Chapter 6: describes the human interface design of the system, the way users interact with the system and provides screen images for each of the components.

1.4 Reference Material

[1] Pressman, Roger S, *Gary B. Shelly, Harry J. Rosenblatt "Software Engineering, A Practitioner's Approach (Eighth ed.), Systems Analysis and Design (Ninth ed.)*.

1.5 Definitions and Acronyms

UFT – United Front for Transformation

2. SYSTEM OVERVIEW

The UTF system is focused around performing a number of functions sent by both the agent and the administrator through the command line interface and the web interface to the server.

It shall be comprised of four modules including: the server, web interface, the database and the command line interface.

The server performs the tasks and returns the results for those that get processed. The agents are allocated to the districts as they are registered and once registered via the web application by the administrator, the system automatically allocates the agent to a district that either lacks agents or has the fewest agents. On registering an agent, a signature should be supplied and should be a single character. The same character should have been stored in the database during registration Each district should have a head of the agents if they are more than one attached to that district. Else, if there is one agent, the agent automatically becomes the head. An enrolled member may be upgraded to an agent by the system's administrator via a web interface. If a member introduces more than 40 new members, and there is an available position in any district, the web interface recommends the member to be upgraded to an agent.

The funds of the party is declared and registered into the web interface by the administrator as received from the well-wishers. The agents use a command-line client socket program to submit details of members who have been enrolled. The member details are stored in the system database.

Enrollment information is saved in a file (A file for each district) for validation by the agent head. Every 5 minutes, a scheduled job runs in the background, checking the files for each district for completeness and validity. In case an agent logs into the system via the C client, they should see status of the file, indicating who has and who has not signed.

The administrator can use the web application to view the following: -

i. List of members enrolled in a given period of time, organized by district, agent, month etc as selected by the administrator

ii. The graphical displays of hierarchy. At the top of each is an agent head, displayed by district

iii. The distribution of money to each of the agents in the different districts

iv. Graphs of variation of funding per month and per period

v. Graph showing donations by month, well-wisher as selected by the administrator

vi. Total number of members

vii. Graph, showing variation in Percentage change in enrollment figures. Ie, if March=30, April = 40, May=20 percentage change in March = 40 – 30 /30, percentage change in May = 20- 40/ 40

3.SYSTEM ARCHITECTURE’

3.1 Architectural design

The system architecture will comprise of 3 major components including the model which comprises of the districts ,System Database and Enrollment file, controller which comprises of the enrollment server and view which comprises of the agents and the administrator.

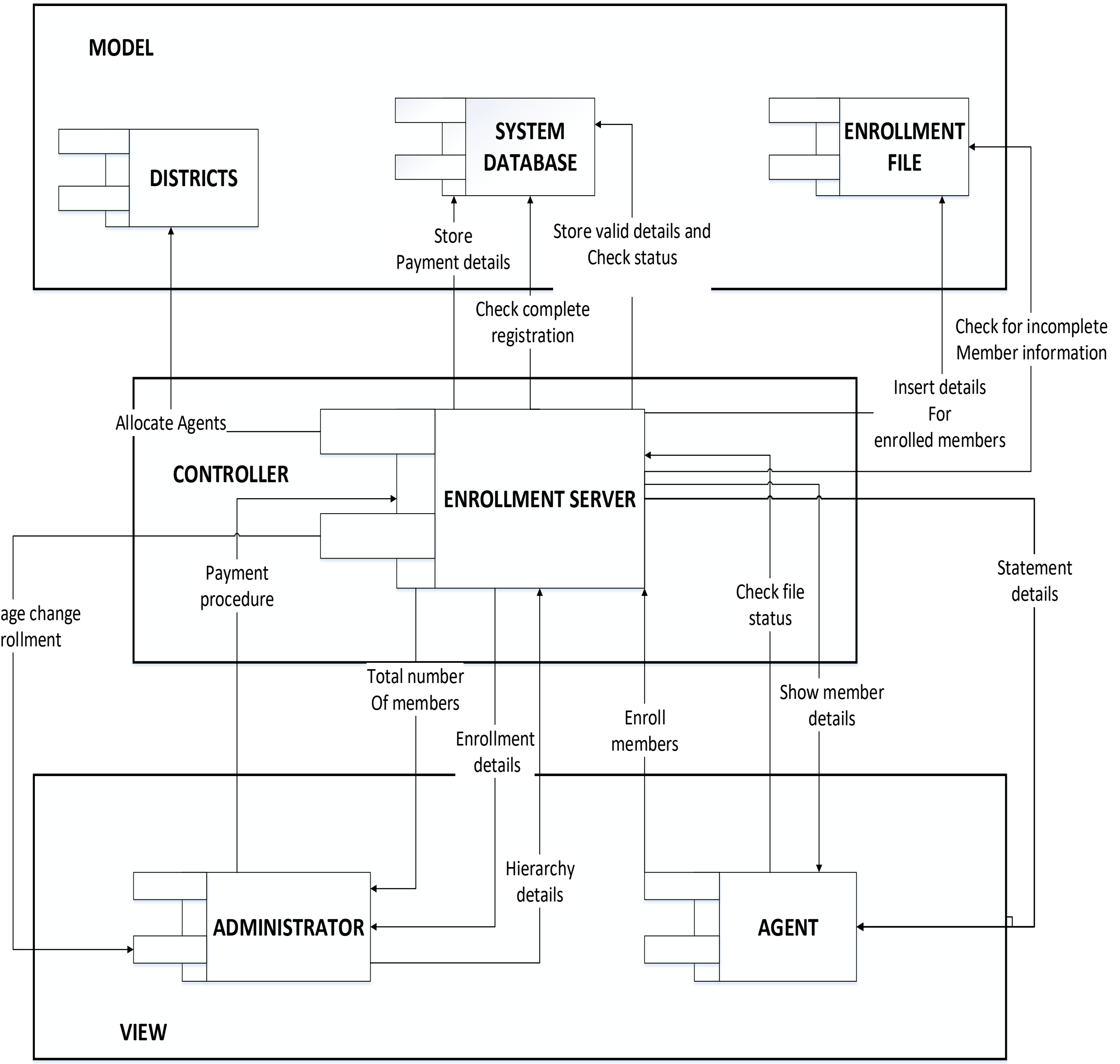


Figure1: Model-view-Controller Architectural Diagram

3.2 Decomposition Description

The overall system will follow a structured approach to design The command line interface will run on a Linux system which shall be developed using C programming language, a structured language.

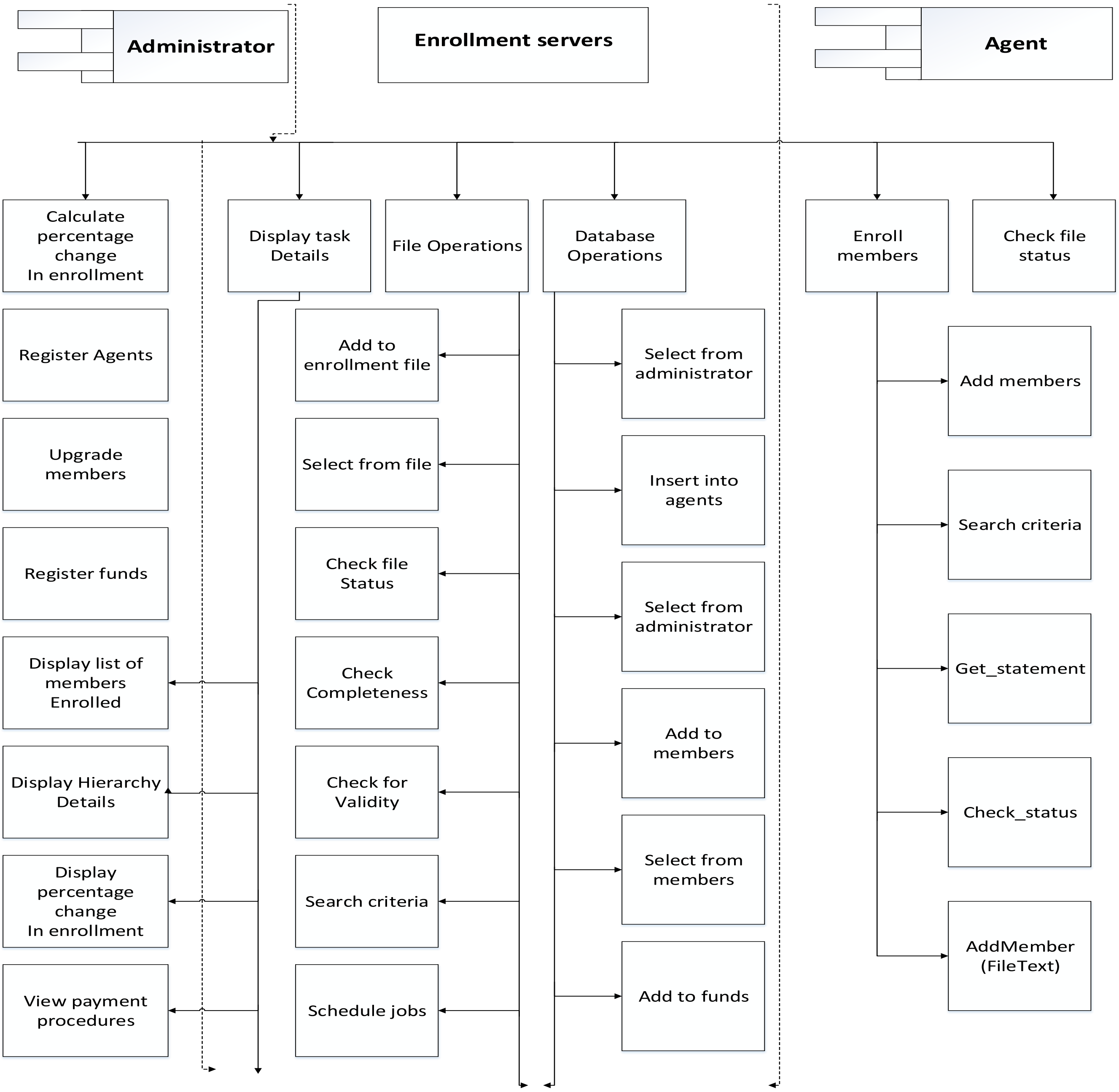


Figure 2: Decomposition Diagram

The Context Diagram of the System

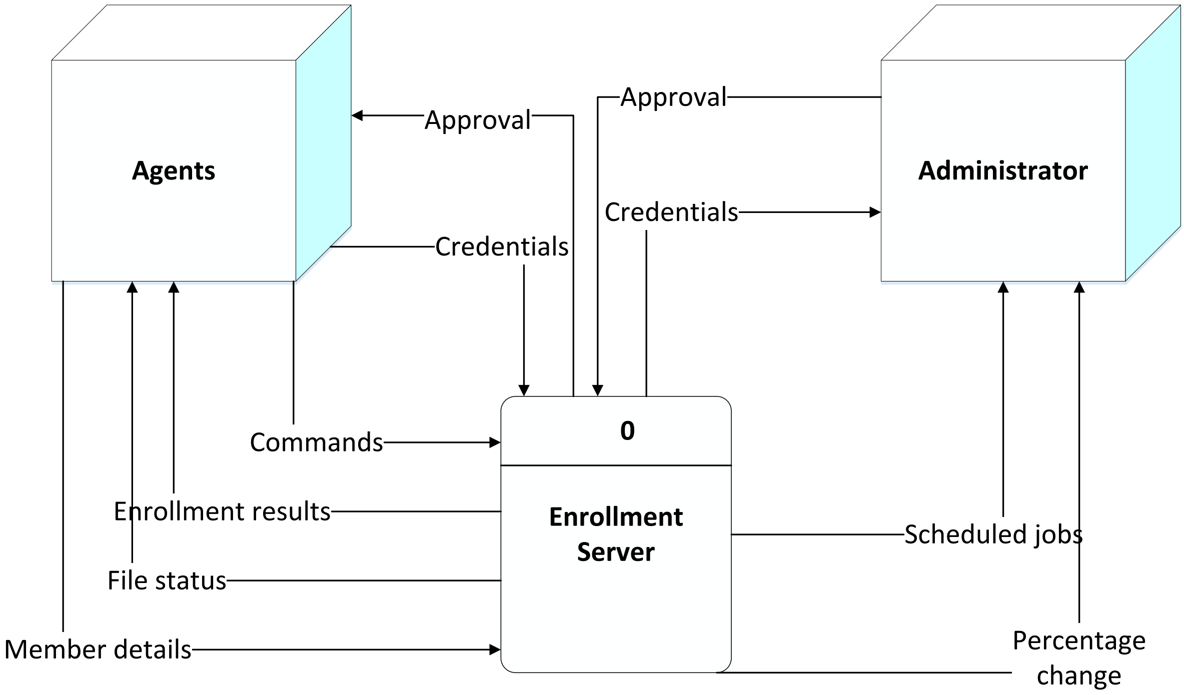


Figure 3: Context Diagram of the System

Data Flow Diagram

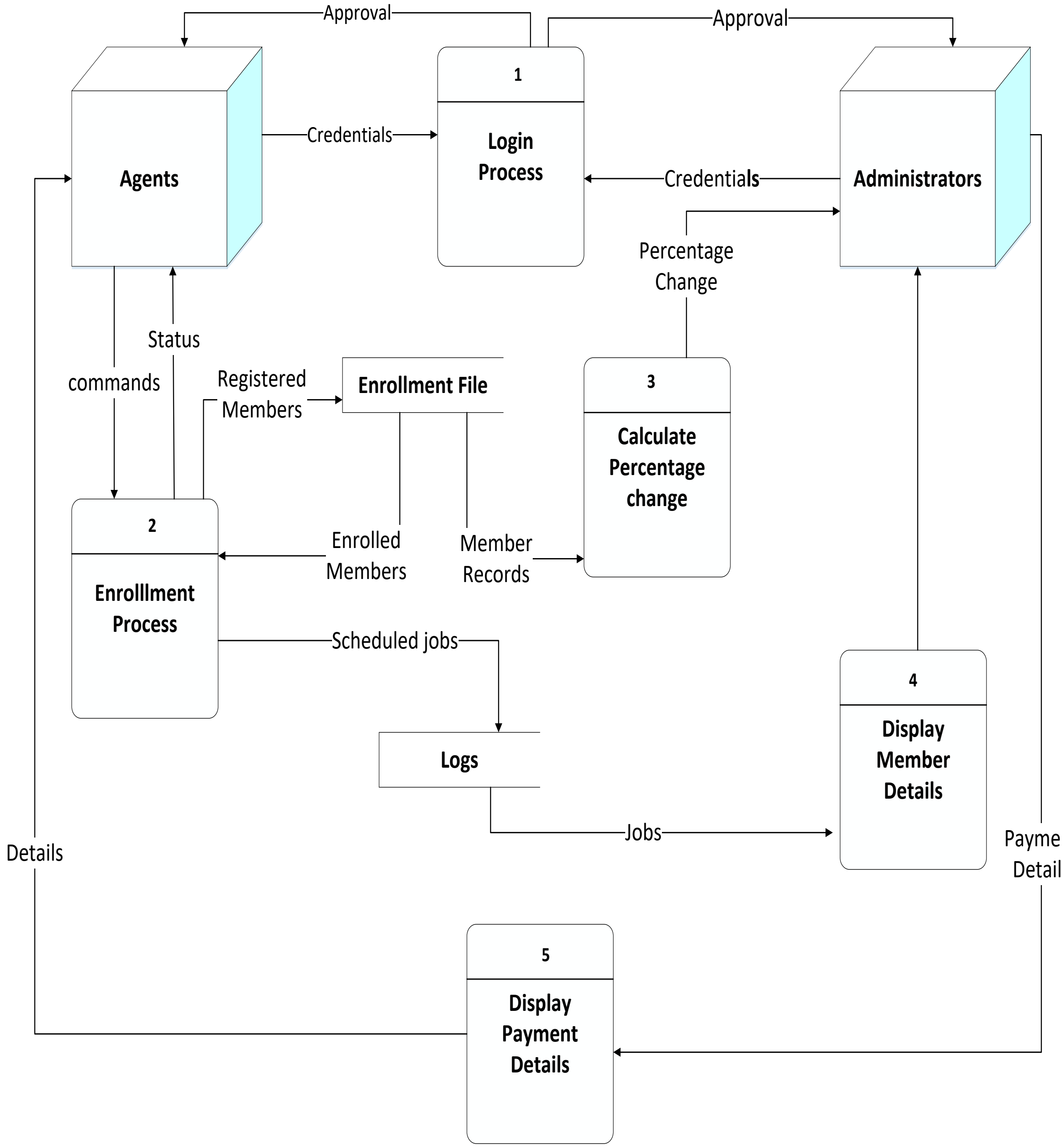


Figure 4: Data Flow Diagram

3.3 Design Rationale

An observation was made, indicating that the model-view-controller pattern would illustrate the system better. This is because the latter gives a more descriptive break-down of the system into components, catering for the storage of system data and how it is coordinated with the respective users. The flow pattern of the data is also indicated in the model-view-controller pattern

Client-Server Model Diagram

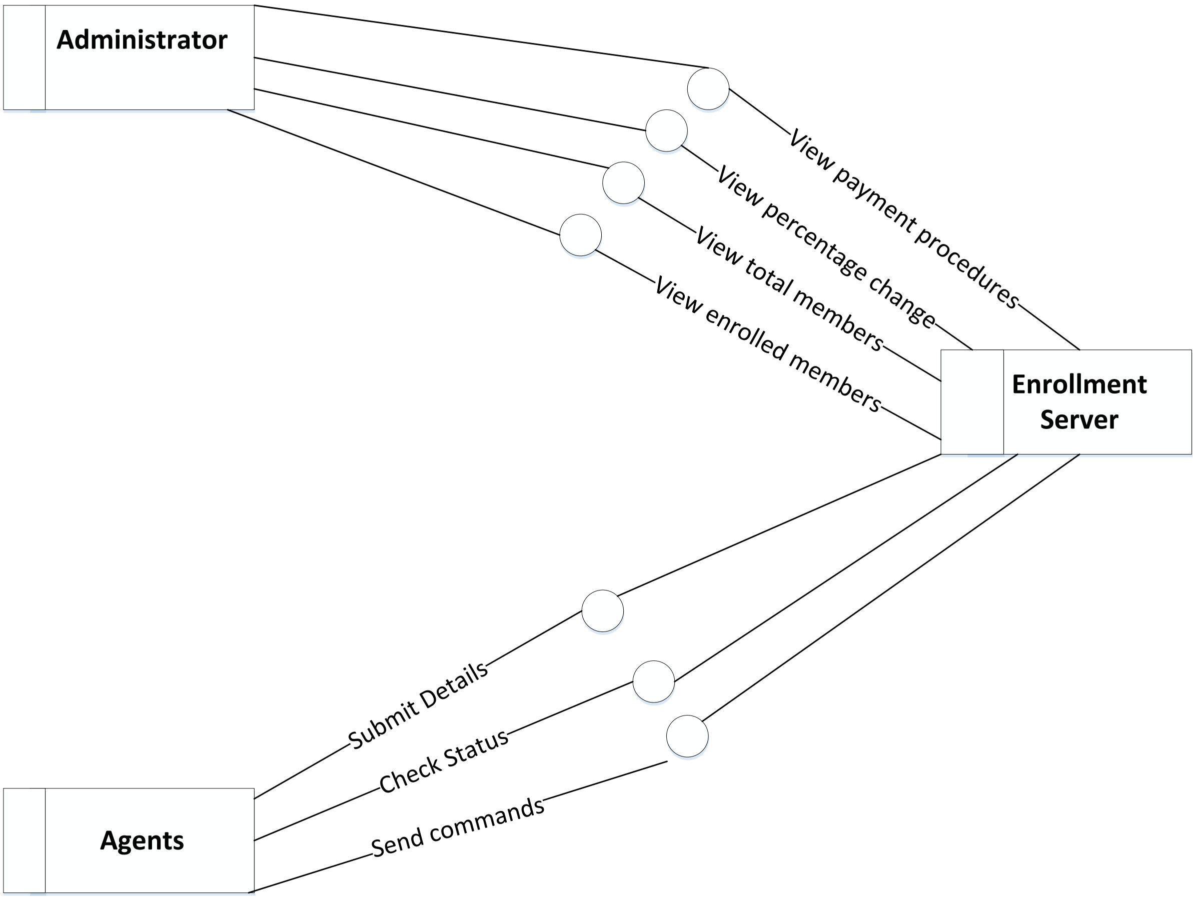


Figure 5: Client Server Diagram

4. DATA DESIGN

4.1 Data Description

From the information domain, both enrollment file and database data structures are realized. There is only one file data structure which is the **enrollment file**.

The database is named UFT Database and data is stored and organized in four relational tables; Agents, Administrator, Members and funds.

Looking at the file data structure in more detail:

An enrollment file is the file which is both **valid** and **complete**.

A **complete** file is the one that has all signatures for members attached to the bottom of the file.

A **valid** file is one, where all signatures are correct. If the agent's signatures match with those in the database, the details are stored in the database accordingly.

The Administrator table in the database is a table which will contain the administrator's verification details, that is the username and password.

Agents is the database table that will contain the agent's verification details, that is, the agent name, username, district and password.

Members is the database table that contains the details of members belonging to that political party, that is, members 'name, id, recommender, username and gender

Funds is the database table that contains the amount of money available on the political party's account as declared by the well-wishers and registered by the administrator, that is, names of well-wishers, total amount, distributions i.e.

In the enrollment file these are the fields and their denotations; Signatures of the agent who sent the file, names of the registered members, Date of registration, Name of the members and finally the Usernames of the registered of the members

ENTITY DIAGRAM OF THE UNITED FRONT FOR TRANSFORMATION (UFT) MEMBER ENROLLMENT SYSTEM

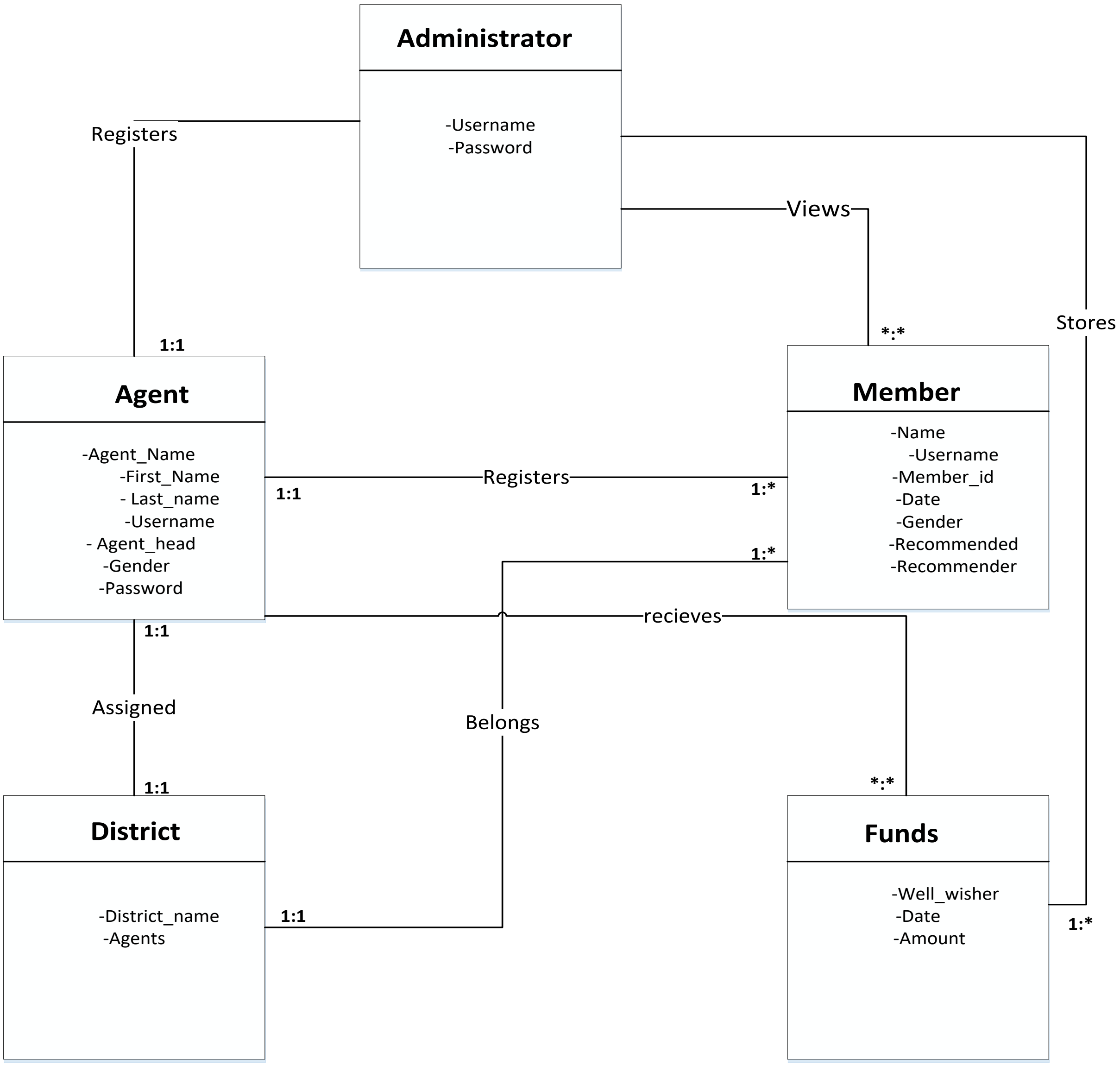


Figure 6: Entity Diagram

4.2 Data Dictionary

This is an individual description of the data elements that are used while design the data structure of the system. It includes the label given to the data element, alternative name, a description, datatype used, the type of data element and attributes.

Table 1: Data Dictionary

LABEL	**ALTERNATIVE NAME**	**DESCRIPTION**	**DATATYPE**	**ENTITY TYPE/DATA ELEMENT**	**ATTRIBUTES**
Administrator's table		Database table with administrator's details		table	Username Password
Password	Agent/ Administrator's Password	Agent/Administrator's pass for login	text	Table's field (Administrator table)	
Agents table		Database table with agent's details		table	District Username Signature Agent_head Password
Members table		Database table with member's details		table	Member_id Name Username Recommender Recommends Gender Date

LABEL	ALTERNATIVE NAME	DESCRIPTION	DATATYPE	ENTITY TYPE/DATA ELEMENT	ATTRIBUTES
Date		Date when the member was registered	Date/time	File and table field (Enrollment file, Members table, Funds table0	
Enrollment file		File that contains members details that are being registered		file	District Username Gender Members name Date
Username	Administrator's/ Agent's Username	Administrator identification name used for logging in	Text	Table Field (Administrator's table, Agent's table, member's table	
Funds table		Database table containing detail of funds held by the table		table	Total Well-wisher name, Well-wisher contributions, Distributions Date

5. Component Design

In this section, the functions of the System enrollment Server as illustrated in *Figure 2* are described in a more systematic way looking at how each interacts with data to give output under the respective component. From *Figure 2***,** some functions reside entirely on the server side without interaction with the two components while the **Display task Details** function is a shared function with its decompositions subdivided to their respective components.

General Functions:

File Operations

Add to Enrollment file

INSERT details into Enrollment file

Select from Enrollment file

```
WHILE signature valid & complete

        READ results
```

Delete from Enrollment file

Result ==0

Search criteria

Select from Enrollment file

Database Operations:

Insert into Funds

INSERT Result into Logs

Select from Funds

```
WHILE Result > 2,000,000
    READ Result
```

Insert into Agents

INSERT agent details into Agents

Select from Agents

SELECT Username, password

Select from Administrator

SELECT Username, Password

Insert into Members

```
INSERT name, recommender, district, username into members
```

Select from Members

```
SELECT recommender from Members
```

Agent Components:

Under the Agent, the following functions are stated;

Receive Jobs

```
READ Member details

IF server == busy
    Send to Busy_list

ELSE
    Perform tasks
```

Perform tasks

```
READ single task

Result = 0
```

Submitting Member list

```
IF job == Addmember
    Result = 0
```

Check file status

```
IF job == Check_status
    Result = File Status
```

Get Payment

```
IF job == get_payment
    Result = User payment statement
```

Submit member file

```
IF job == Addmember "filename.txt"

Result = 0
```

Search Records from file

```
IF job == Search_criteria(name/date)
        Result = Member records
```

Administrator Component:

Functions under the Administrator Component are:

Registering Agents

```
READ
Member details (district, Username, Password)
IF (District_Agents == 0)
        Member = Agent_head
ELSE
        Member = Agent
```

Upgrading Members to Agent

```
IF (recommended > 40)
        Member = Agent
ELSE
        Member = Member
```

Payment Procedure

```
IF (Treasury > 2,000,000)
Administrator = 0.5 *Agent_salary
        IF (Agent == Agent_head)
                Agent_head = 7/4*Agent_salary
        ELSE
                Agent = Agent_salary
ELSE
        No Payments Made
```

Percentage Enrollment

```
Previous_monthNo = n1
Current_monthNo = n2

Percentage_change = (n2 – n1)/n1
```

Description

Result: refers to a processed job

Busy_list: refers to a file containing jobs waiting to access the server

Username: Agent/Administrator names used to access log into the system

Password: Agent/Administrator pass to access the system

Treasury: Funds collected for the political party

District_agents: Agents available for a particular district

Recommended: Number of members recommended a member

Agent_Salary: Money earned by the party agent

Agent_head: Leader of agents in a particular district

Previous_monthNo: number of members registered in the previous month

Current_monthNo: number of members registered in the current month\

n1: a variable representing the Previous_monthNo

n2: a variable representing the Current_monthNo.

6. Human Interface Design

6.1 Overview of User Interface

The user interface features a web-based environment and command line interface. The command line interface is used by the agents while sending the tasks and progress commands, as well as receiving the results and status of scheduled tasks. The web-interface is used by the administrator to view the performance and corresponding reports of the system.

6.2 Screen Images

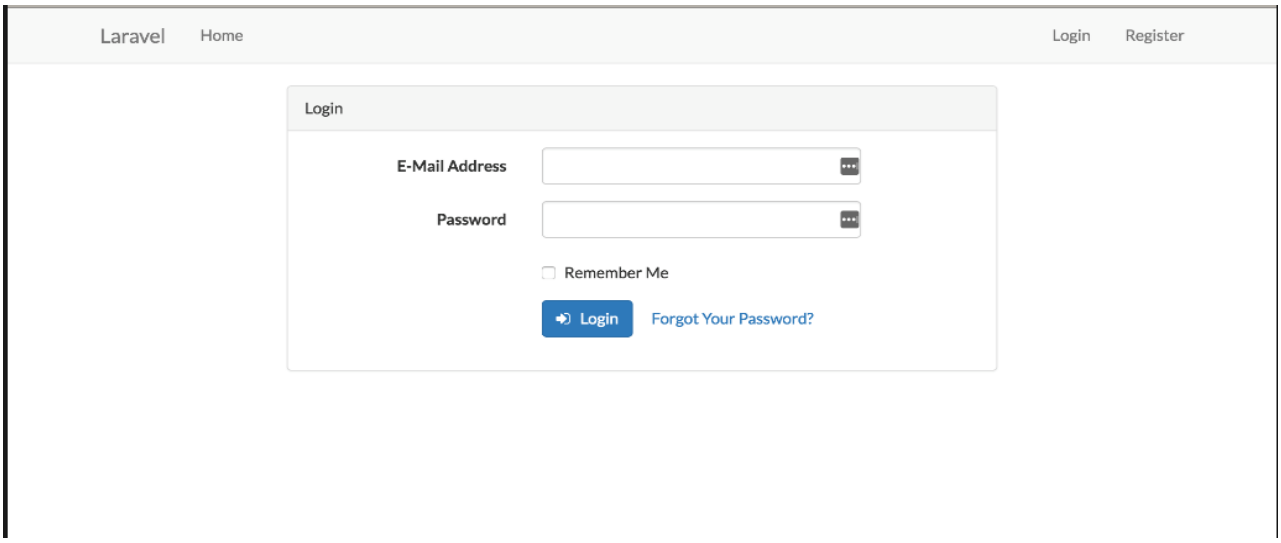


Figure 7: Administrator's Login Page

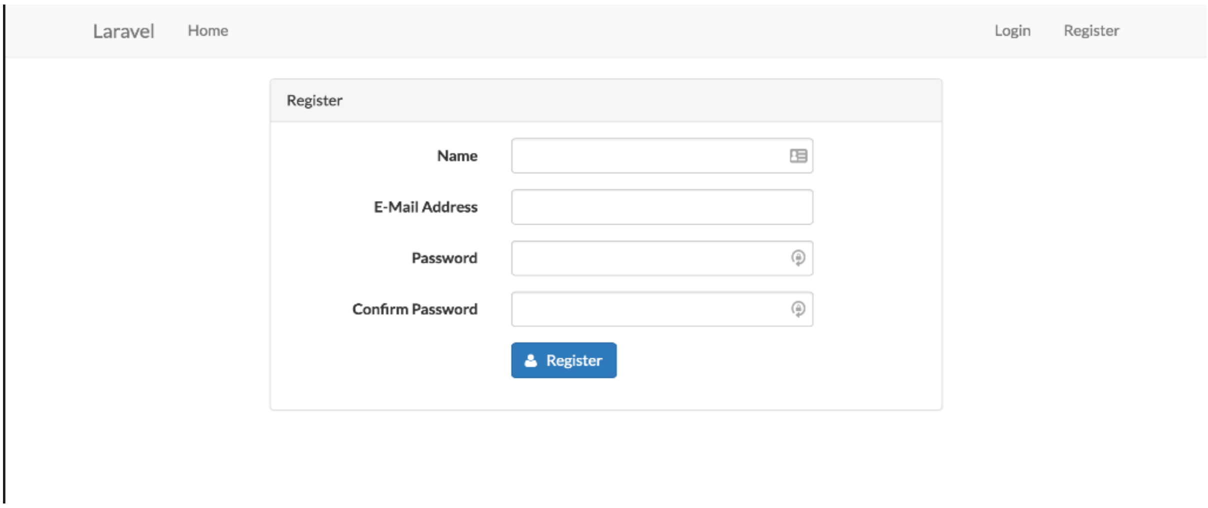


Figure 8: Administrator's page for registering the agents

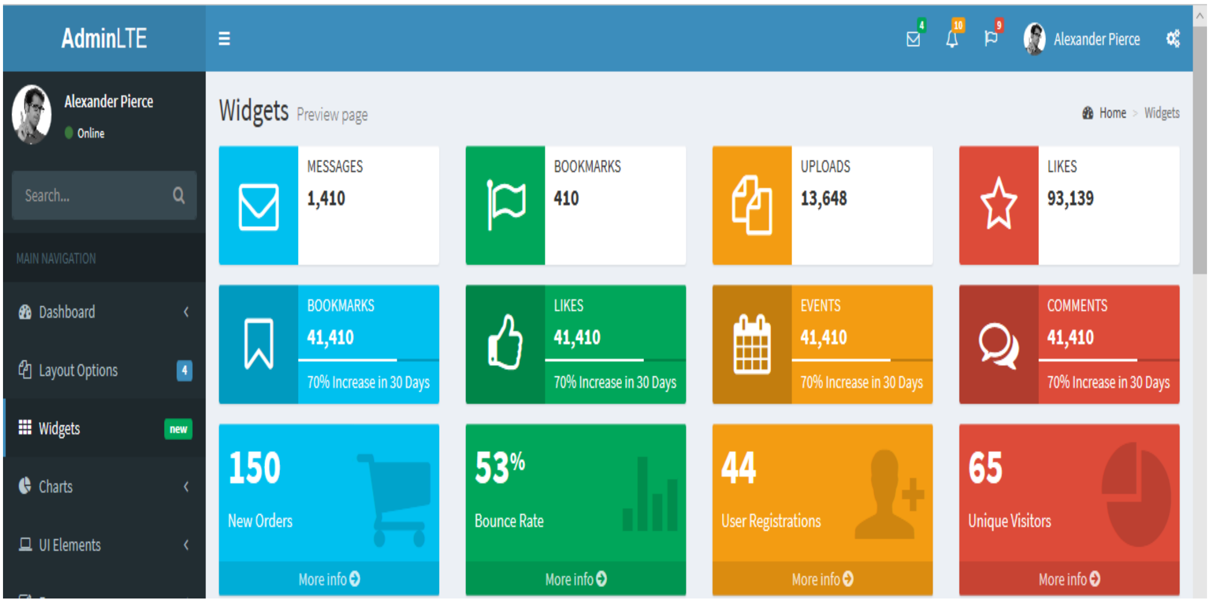
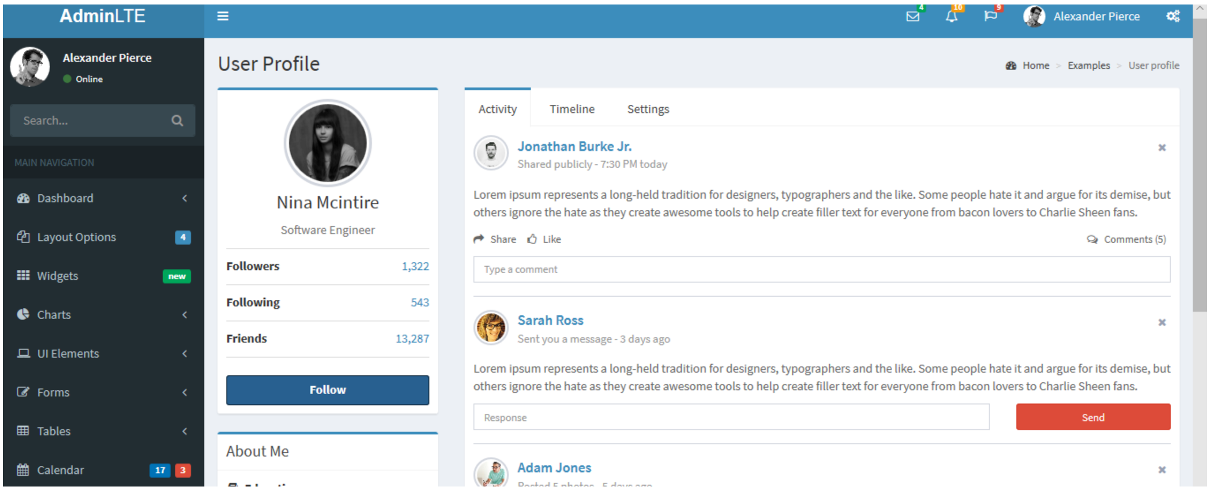

AdminLTE
Alexander Pierce
Alexander Pierce
Online
Search...
MAIN NAVIGATION
Dashboard
Layout Options 4
Widgets new
Charts
UI Elements
Widgets Preview page
Home > Widgets
MESSAGES 1,410
BOOKMARKS 410
UPLOADS 13,648
LIKES 93,139
BOOKMARKS 41,410 70% Increase in 30 Days
LIKES 41,410 70% Increase in 30 Days
EVENTS 41,410 70% Increase in 30 Days
COMMENTS 41,410 70% Increase in 30 Days
150 New Orders More info
53% Bounce Rate More info
44 User Registrations More info
65 Unique Visitors More info

Figure 9: Administrator Dashboard



Lorem ipsum represents a long-held tradition for designers, typographers and the like. Some people hate it and argue for its demise, but others ignore the hate as they create awesome tools to help create filler text for everyone from bacon lovers to Charlie Sheen fans.

Lorem ipsum represents a long-held tradition for designers, typographers and the like. Some people hate it and argue for its demise, but others ignore the hate as they create awesome tools to help create filler text for everyone from bacon lovers to Charlie Sheen fans.

Figure 10: Member Profiles

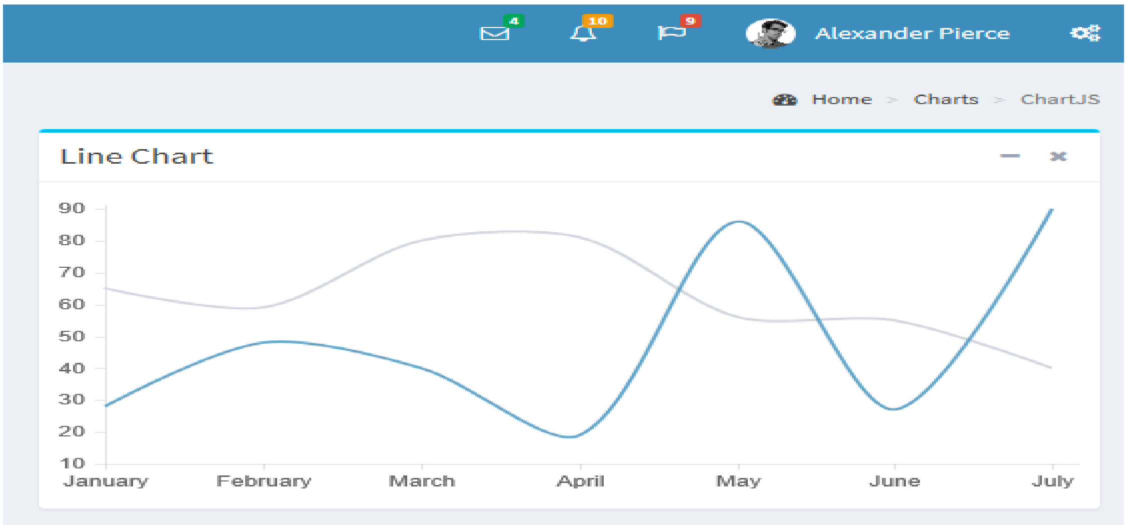
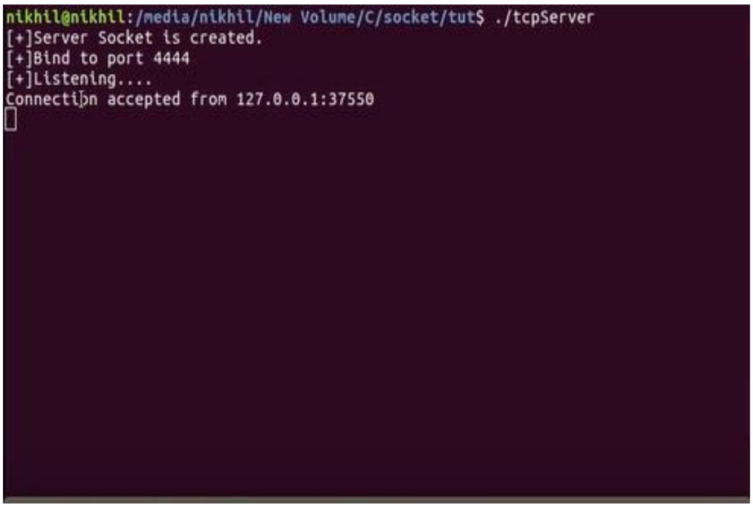


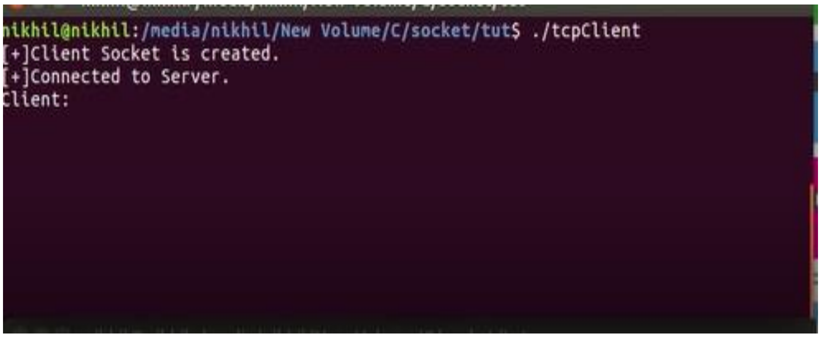
Figure 11: Percentage Change

Command line Interfaces



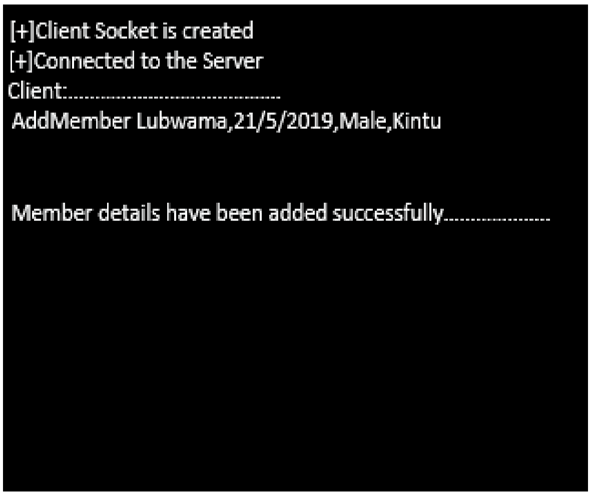
```
nikhil@nikhil:/media/nikhil/New Volume/C/socket/tut$ ./tcpServer
[+]Server Socket is created.
[+]Bind to port 4444
[+]Listening....
Connection accepted from 127.0.0.1:37550
```

Figure 12: Server socket



```
ikhil@nikhil:/media/nikhil/New Volume/C/socket/tut$ ./tcpClient
[+]Client Socket is created.
[+]Connected to Server.
Client:
```

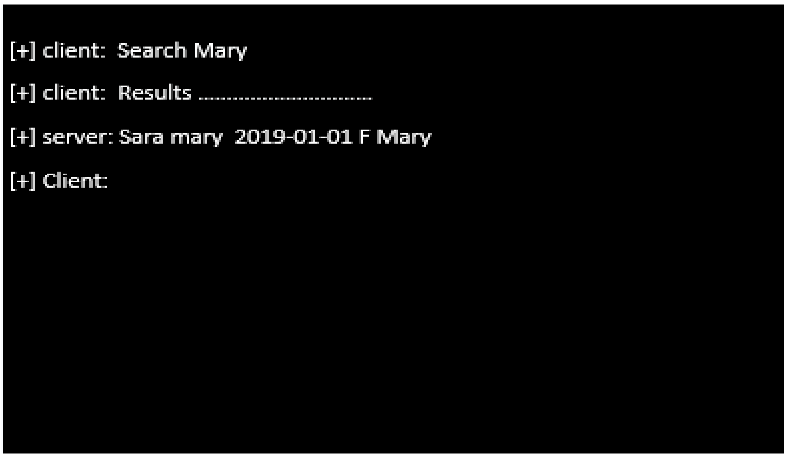
Figure 13: Client socket



```
[+]Client Socket is created
[+]Connected to the Server
Client:.........................................
 AddMember Lubwama,21/5/2019,Male,Kintu


 Member details have been added successfully.....................
```

Figure 14:Addmember command



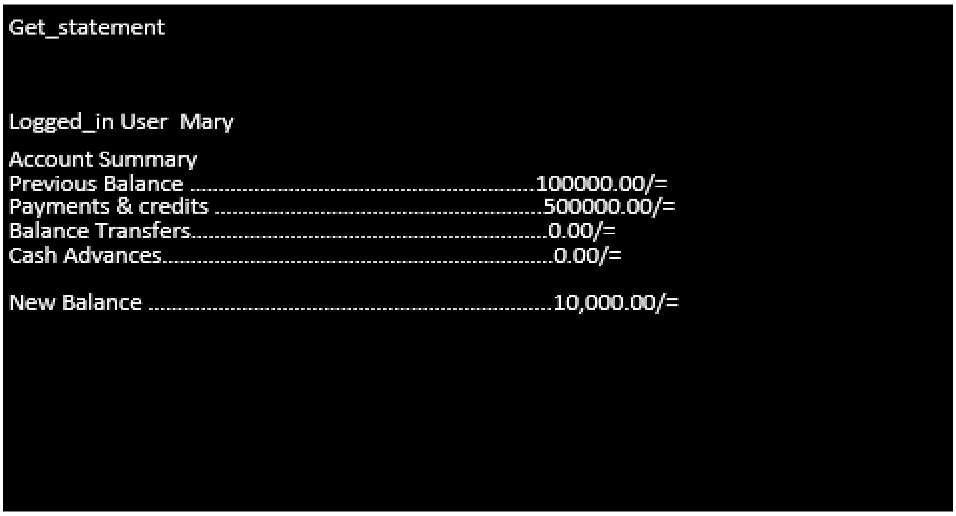
```
[+] client:  Search Mary

[+] client:  Results ..............................

[+] server: Sara mary  2019-01-01 F Mary

[+] Client:
```

Figure 15: Search command



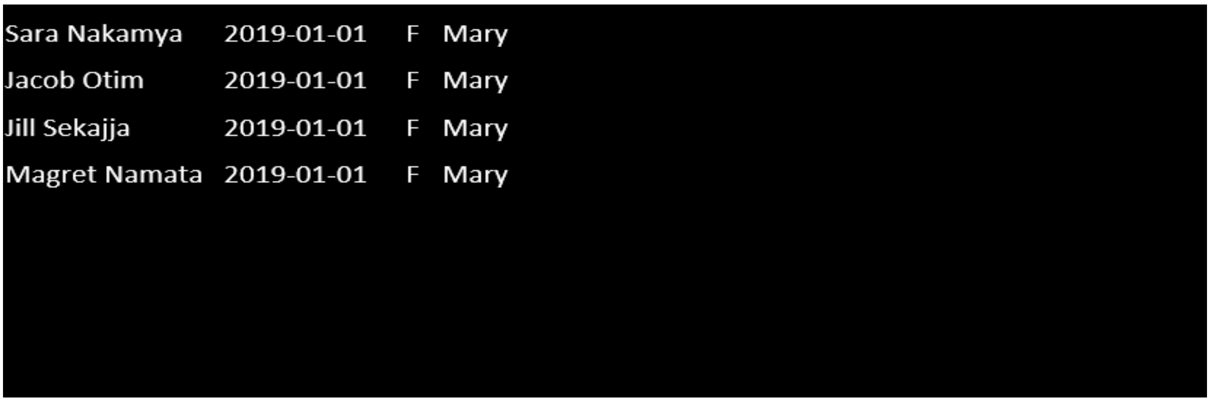
```
Get_statement


Logged_in User  Mary

Account Summary
Previous Balance ...........................................................100000.00/=
Payments & credits .........................................................500000.00/=
Balance Transfers................................................................0.00/=
Cash Advances.......................................................................0.00/=

New Balance ...........................................................................10,000.00/=
```

Figure16: Get Statement command



```
Sara Nakamya     2019-01-01     F    Mary
Jacob Otim       2019-01-01     F    Mary
Jill Sekajja     2019-01-01     F    Mary
Magret Namata    2019-01-01     F    Mary
```

Figure17: File Example