**UNIVERSITY OF ZIMBABWE**

**University of Zimbabwe School of Technology**

****

**DIPLOMA IN APPLIED INFORMATION TECHNOLOG**Y

|  |  |
| --- | --- |
| **PROJECT NAME:** | **BANK INFORMATION SYSTEM** |
| **NAME:** | **MAPURANGA MICHAEL BOLTON** |
| **REG NUMBER:** | **R176785X** |
| **ADVISOR:** | **MR R. NYAMUKONDIWA** |

**MARCH 2018-MARCH 2019**

Table of Contents

[ABSTRACT iii](#_Toc874971)

[ACKNOWLEDGEMENT iv](#_Toc874972)

[LIST OF FIGURES v](#_Toc874973)

[CHAPTER 1: INTRODUCTION TO THE RESEARCH](#_Toc874974)

[1.1 Brief history of the organisation. 1](#_Toc874975)

[1.2 Problem Definition 1](#_Toc874976)

[1.3 Project Scope 1](#_Toc874977)

[1.4. Feasibility Study 2](#_Toc874978)

[1.4.1. Technical Feasibility 2](#_Toc874979)

[1.4.2. Economic Feasibility 2](#_Toc874980)

[1.4.3. Social Feasibility 2](#_Toc874981)

[CHAPTER 2: SYSTEM ANALYSIS AND DESIGN](#_Toc874982)

[2.1 Information Gathering Techniques 3](#_Toc874983)

[2.2. Required Analysis and Design 5](#_Toc874984)

[2.3. DFD Diagrams 5](#_Toc874985)

[CHAPTER 3: IMPLEMENTATION / CODING](#_Toc874986)

[3.1 Login Function 8](#_Toc874987)

[CHAPTER 4: SYSTEM TESTING](#_Toc874988)

[4.1 Testing Plan 18](#_Toc874989)

[4.2 System Testing 18](#_Toc874990)

[4.3 Dump screens 19](#_Toc874991)

[4.4 System Validation Procedures 23](#_Toc874992)

[4.4.1 Problems Faced 23](#_Toc874993)

[4.4.2 Limitations 23](#_Toc874994)

[4.4.3 Future Work 24](#_Toc874995)

[4.4.4 Information archiving 24](#_Toc874996)

[4.4.5 Learning Experience 24](#_Toc874997)

[4.4.6 Conclusion 24](#_Toc874998)

[REFERNCES 25](#_Toc874999)

# **ABSTRACT**

The report include a development presentation of a partly bank management system that undergoes account information application. It manage the customer, data within the bank premise. The system as such as it has been developed is called Partly Bank Management System but it consist on upgrading functionally related with the banking system which is Bank Account Information System. The choice of the programming tools are individual and particular. Bank Management System as a backbone of a company therefore the client play a major role in the succession of an organisation. By this software it makes it easy for the teller to keep tracking all the records. This software allow the teller to add records, to show records, to search records, to update records, to delete records even to close the application. Each client in the data base is associated with a position to know or verify his or her transactions. Most of all the teller can assign and access the progress in order to keep the client records

A flexible and easy to use software application as a solution to minimize and provides modules for personal information management whereby the organisation and company are able to manage the crucial client assets. The combination of this modules and supranational single module of Bank Account information System into one application assures the perfect platform for reengineering and aligning with the business goals. This system brings about an easy way of maintaining the details of clients.

It is simple to understand and can be used by anyone who is not even familiar with programming. It is user friendly and just ask for the user to follow step by step operations by giving each to follow options. It is fast and can perform many operations for the bank.

The goal of this project is to design and develop a Bank Account Information System to fill existing gape in electronic management of banking.

# **ACKNOWLEDGEMENT**

I am greatly indented to the following for the contribution to my work:

The almighty God who give me the insight and perseverance to accomplish this work. My parents for all the support and my siblings. Bonfire who I never have time for due to my business busy schedule. My supervisor who soundly advised, helped me stir this project in the right project, a work I will cherish. Members of staff at the UZ SOT department for the solid knowledge based that enables me carry out the research. Finally my class mates whose technical and moral support throughout my stay at the software studies department was of great help.

# **LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Figure 2.1. | Activity diagram for Bank Account Information System | 5 |
| Figure 2.2. | Use Case diagram for Bank Account Information System | 6 |
| Figure 2.3. | Analysis Class diagram for Bank Account Information System | 6 |
| Figure 2.4. | Design Class diagram for Bank Account Information System | 7 |
| Figure 2.5. | Sequence diagram for Bank Account Information System | 7 |
| Figure 4.1-4.10 | Dump Screens | 19-23 |

# **CHAPTER 1: INTRODUCTION TO THE RESEARCH**

## 1.1 Brief History Of The Organisation.

Youth Centric Bank was established in 2014 by a certain organisation for the benefit of students’ unemployed, school leavers’ youth to be financially equipped in terms of loans, fees funding and proposed projects. It is located in Muzarabani rural community where no one have ever heard of but we want to make it being published to the well-known place. It consist of three board members and few staff that are striving to make the bank stand on.

## 1.2 Problem Definition

It is evident that procedures are taken are similarly not comprehensive and protective to the client information and banking details. There are some human errors in the system that the teller have the rights to update , to show , and to add the records in which it is not secure , the teller may end up stealing the money from that saving account by manipulating the accounts. Other challenge is that multinational will have all the clients’ stored at the headquarters of the company made it difficult to access the account information when needed on a short notice.

The mentioned problems can be tackled by designing and implementing a bank Account Information System that will only base on client records. The system will maintain the information in a file for full privacy and authority access. The project is aimed at setting up employee information system about the status of a client, background in order to monitor through a password protected system.

## 1.3 Project Scope

As the backbone of the company we need to access if this system can be implemented to play a major role in easing the tracking of all records. It allow the teller to add, show, search, update and delete using various passwords undergoes an admin password.

The scope of this project will be limited by the following:

* Client profile
* Client default account
* Client additional of details
* It is a complete elimination of theft and management of client funds, also by establishment of passwords that will be sent to a validated customer.

Requirements and constraints

Authentication

Login – Admin password is required with his or her username

Log In – teller also have to log in using username and password

Log in – customer must also login to his or her own account

Log in fail – If the user does not exist in the database or not authorised by the bank will not log in but if the user already exist in the system but fail to log in three times in the system will bared him or her out.

Authorisation

User role check – after log in the user will use his privileges and the user interface will display according to the role type

Data processing

Add record – the user have the role to add the new record according to client requirements and procedures

Display record – the user with defined role can display the content of the record being more specifically

Edit and delete record - the user can edit the record only if the admin put a part for him or her

Search record – it also the user password for him to have privilege to search for the specific record

## 1.4. Feasibility Study

It’s an analysis used in measuring the ability and likelihood to complete a project successfully including all relevant factors. It determine positive and negative outcomes of a project before investing considerable amount of time and money into it. It focuses on the following:

* User requirements and best utilization of available resources
* Develop a cost effective and technically feasible system

### 1.4.1. Technical Feasibility

It focuses on the on the technical resources available, it helps to determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into a working system. Technical feasibility always involves the evaluation of the hardware, software and other technology requirements of the proposed system

### 1.4.2. Economic Feasibility

Generally it means whether a business or project is feasible cost wise and logistically. It involves analysing the costs and revenues a business would incur by undertaking a certain project to determine whether or not it is logical and possible to complete.

### 1.4.3. Social Feasibility

This new system targeted youth, tellers and managers in order:

* To provide good quality of unique services, flexible also targeted youth and tellers to have quicker access to needed information
* To easily identify what to offer for specific field, for example hardware, software or network books.

Output under social feasibility recommendation

* It will be easy flow of work for users
* Provide less work for all bank tellers and fast, reliable, smart and unique to use.

# **CHAPTER 2: SYSTEM ANALYSIS AND DESIGN**

The system focuses on major discipline and particular basic with the information technology field whereas the programming data processing system comprise of enterprise resource price software. The bank have a large volume of data and can be daunting without a more sophisticated tool to store and retrieve data. The various levels of sophisticated can be aspects of technology [2]. The focus have shifted to atomizing as many transactions as possible to achieve effectiveness and efficiency. The technology of the future will be able to speed up access of current information and reliable to access this information via multi system will give organisation a strategic age. The new system will not necessarily mean reduction of information but will able to focus on transforming information into knowledgeable information that can be used by an organisation to making decision.

## 2.1 Information Gathering Techniques

The methodology of this system is a collection of procedures, tools, documentation aid that will help the developer to implement their information system [3]. There are a number of system to be used related to the project are factors that are considered e.g. time, cost, incorporation of requirements changes during the development process, system complex, communication between customers and bank

There are five commonly regarded methods of gathering the requirements of a new system that maybe relevant to the project at hand

* Research
* Interviews
* Sample documents
* Questionnaires
* Observation

Many projects are usually taken by the third parties that have very limited understanding of the youth centric bank. Research provided the opportunity for the analysis, process and to gauge an understanding of the business activities, processes and practices that may go on within the bank. Research is always a great necessity of high quality in order to produce knowledge that is applicable outside of the research setting. The result of the study may have implications for future project implementation.

Interviews

Interview is a conversation where questions are asked to elicit information. The interviewer is a professional or a paid researcher, sometimes trained, who poses questions to the interviewee in an alternating series of usually brief questions and answers. The advantages of interviews include the possibilities of collecting detailed information about research questions. It is easy to capture the requirements, suggestions and the change that the customers and employers need for an effective and competitive system.

Sample documents

Document analysis is the form of quantitative research in which documents are interpreted by the researcher to give a voice and meaning around an assessment topic. You can gather relevant text and explore the context and also authenticity of documents within the banking hall. Sample documents are useful when designing the database as they outline the exact data that should be stored in the database. Sampling the documents is less time consuming, scope of sampling is high and accuracy of data is high. However it is risk to fully rely on the sample documents since it’s difficult to truly select a representative sample, there is need to specific knowledge and there are impossibilities in sampling.

Questionnaires

These can be classified as both quantitative and qualitative method depending on the nature of the questions. This provides access to the questioner to obtain multiple choice answers, it has advantages of increased speed of data collection, no cost requirements, higher level of objectivity compared to many alternative methods of primary data collection

Disadvantages

Selection of random answers. No possibility for respondent to express the additional thoughts about the system due to the absence of a relevant question

Observation

The main purpose of the analyst on this technique was to gather the first-hand information to aid the analyst in analysing the system under study. The systems analyst participated in and watched people performing activities as a way to learn about the current system. The technique entails the observation of events as they unfold and produce results afterwards. Through this fact finding technique the system analyst observed the following

* Traditional banking habits
* Poor security
* Time consuming
* Long waiting lines in the banking halls
* Slow bank tellers
* Long processes

Strength of observation

* Direct access to research phenomena
* High level of flexibility in terms of application and generation of an effective system

Existing system

When records are accessed frequently, managing such records becomes difficult. Therefore organizing data becomes difficult. If any old data or information is to be fetched then it is a great problem for user to get the information in short span of time as to get information from files is not an easy task. The existing system requires that there be an improved system that can help the user or customer perform a task more easily and efficiently.

Proposed System

It creates a user friendly environment where a customer can access all the benefits of the system, with improved fraud identification and prevention. It provides security from unauthorized access, only the admin and authorized users are access granted to the system. Increased efficiency and reduce transportation costs. Up-to-date records of the customer are maintained by the authority and less time consuming, hence time wastage can be avoided. To provide flexibility for secure and save transaction and elimination of man power and human errors should be able to maintain information and able to keep records of banking events.

## 2.2. Required Analysis and Design

System Analysis

System analysis is a process of gathering and interpreting facts, diagnosing problems and information [4] about the Youth Centric Bank System to recommend improvements of the system by allowing an update on a specific module of the system which is Bank Account Information System in terms of records.

Systems design

Systems design is the process of planning a new system, replace or complement an existing system. By the edification of DFD diagrams comprise of activity, user case, analysis, design and sequence illustrations,

## 2.3. DFD Diagrams

Proposed system model diagrams

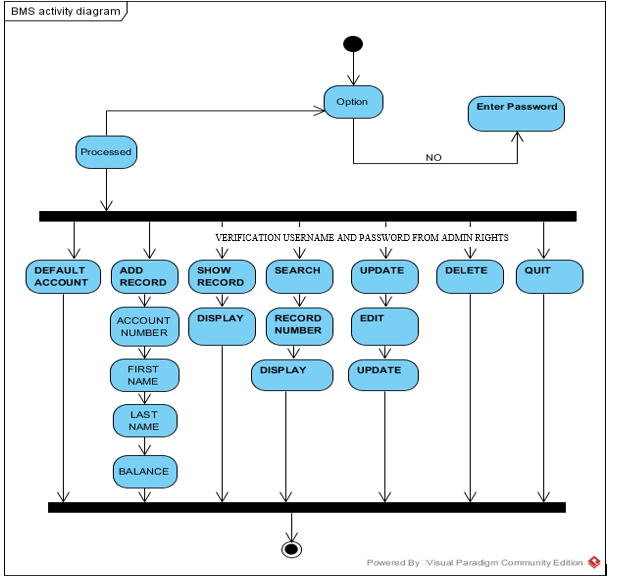


Figure 2.1 Activity Diagram for Bank Account Information Application

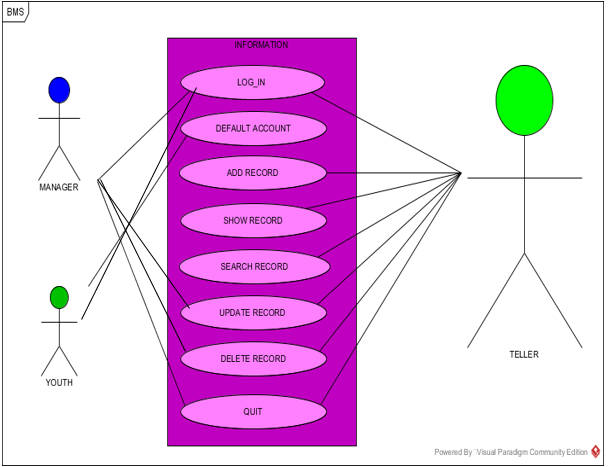


Figure 2.2 Use-Case Diagram for Bank Account Information System

* The user can perform various functions

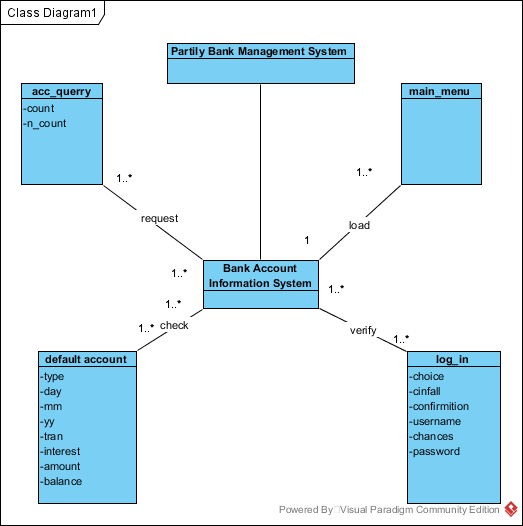


Figure 2.3 Analysis Diagram for Account Information System

A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams involve global system description, such as the system architecture, and detail aspects such as the attributes and operations within a class as well. The most common contents of a class diagram are:

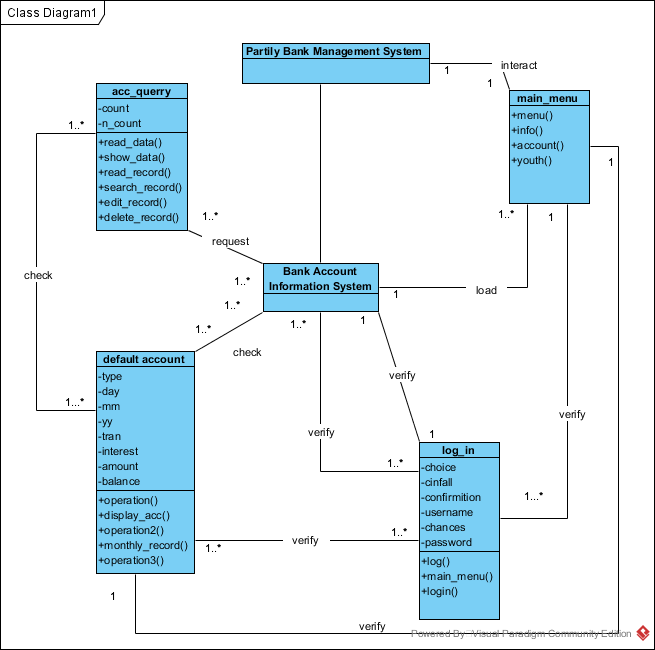
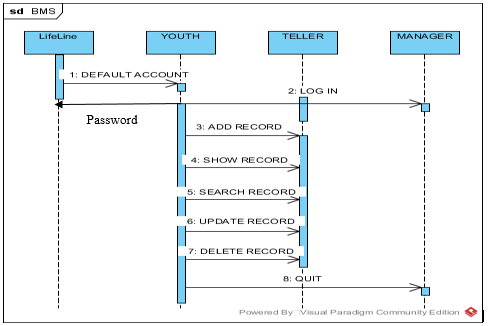


Figure 2.4 Design Diagram for Account Information System



Password

Figure 2.5 Sequence Diagram for Account Information System

Sequence diagrams are interaction diagrams that detail how operations are carried out

# CHAPTER 3: IMPLEMENTATION / CODING

## 3.1 Login Function

The user will be prompted to enter password to the system. If he or she enters the wrong password within three chances, the program will terminate [1]

// Function used to do screening

class main\_menu

{

public :

void menu() ;

void intro() ;

protected :

//void edit\_menu() ;

void edit\_account() ;

void edit\_youth() ;

} ;

// Class contains the customers daily transaction entry

class account

{

public:

//void new\_account(void); // Function to create a new account

void close\_account(); // Function to close an account

void display\_account(); // Function to display the accounts

//void transaction(); // To display the transaction process

void clear(); // Function to perform a clear screen function

void month\_report(); // Function to list monthWise transaction report

private:

void add\_to\_file();

// Function to add transaction records

//void delete\_account(); // Function to delete a transaction record

int no\_of\_days(); // Function to find the total days

float calculate\_interest();

// Function for calculating interest of anaccount

void display(int); // Function to display a transaction account

//void box\_for\_display(); // Function for displaying box

int accno;

char type[10]; // Account type as Cheque or Cash

int dd, mm, yy; // To store the system date/ Enter date

char tran; // As the account type is Deposit or Withdraw

float interest, amount, balance;

};

class account\_query : public account

{

private:

char account\_number[20];

char firstName[10];

char lastName[10];

float total\_Balance;

public:

void read\_data();

void show\_data();

void write\_rec();

void read\_rec();

void search\_rec();

void edit\_rec();

void delete\_rec();

};

class login : public account

{

public:

void mainmenu();

void Login();

int choice;

bool cinfail;

int confirmation;

char username;

void log(){

cout << "LOG IN SUCCESSFUL!";}

};

void Login()

{

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"#####\*\*WELCOME YOUTH CENTRIC BANK\*\*#####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"#####\*\* ACCOUNT INFORMATION SYSTEM \*\*#####\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"######\*\*DESIGNED BY : BOLTON MICHAEL\*\*###\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\t\t\t\*\*\*PASSWORD NEED FOR CONFIRMATION AND SYSTEM ACCESS\*\*\*\n";

cout << "\t\t\t\*\*\*\*\*\*\*\*\*\* ALLOWED THREE TIMES ONLY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

int chances = 3;

//char default\_Password[15] = "bolton";

password\_re:

char password[15];

char username[30];

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout << "\t\t\t>>>>>>>>>>>>>>>>>>>>ENTER PASSWORD: ";

cin >>password;

cout << "\t\t\t>>>>>>>>>>>>>>>>>>>>ENTER USERNAME: ";

cin >> username;

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

if (strcmp(password , "bolton") == 0) {

if(strcmp(username, "michael")== 0)

cout << "\t\t\tSUCCESSFULLY ACCESSED\n";

return;

}

else {

--chances;

if (chances == 0){

cout << "\t\t\tINVALID PASSWORD\n";

cout << "\t\t\t PROGRAM EXECUTION.\n";

cout << "\t\t\t CLOSE THE APPLICATION.\n";

cout << "\t\t\t SORRY TRY AGAIN.\n";

exit(EXIT\_FAILURE);

}

cout << "\t\t\t! Wrong Password . Please Try Again\n";

goto password\_re;

}

}

/\*\*\*\*\*TRY TO TEST DO-WHILE LOOP FOR THE LOGIN FUNCTION WHICH DID NOT EXECUTE\*\*\*\*\*/

void account\_query::read\_data()

{

cout<<"\nEnter Account Number: ";

cin>>account\_number;

cout<<"Enter First Name: ";

cin>>firstName;

cout<<"Enter Last Name: ";

cin>>lastName;

cout<<"Enter Balance: ";

cin>>total\_Balance;

cout<<endl;

}

void account\_query::show\_data()

{

cout<<"Account Number: "<<account\_number<<endl;

cout<<"First Name: "<<firstName<<endl;

cout<<"Last Name: "<<lastName<<endl;

cout<<"Current Balance: BOND. "<<total\_Balance<<endl;

cout<<"-------------------------------"<<endl;

}

void account\_query::write\_rec()

{

ofstream outfile;

outfile.open("record.bank", ios::binary|ios::app);

read\_data();

outfile.write(reinterpret\_cast<char \*>(this), sizeof(\*this));

outfile.close();

}

void account\_query::read\_rec()

{

ifstream infile;

infile.open("record.bank", ios::binary);

if(!infile)

{

cout<<"INVALID NOT FOUND!"<<endl;

return;

}

cout<<"\n\*\*\*\*Data from file\*\*\*\*"<<endl;

while(!infile.eof())

{

if(infile.read(reinterpret\_cast<char\*>(this), sizeof(\*this))>0)

{

show\_data();

}

}

infile.close();

}

void account\_query::search\_rec()

{

int n;

ifstream infile;

infile.open("record.bank", ios::binary);

if(!infile)

{

cout<<"\nINVALID NOT FOUND!"<<endl;

return;

}

infile.seekg(0,ios::end);

int count = infile.tellg()/sizeof(\*this);

cout<<"\n There are "<<count<<" record in the file";

cout<<"\n Enter Record Number to Search: ";

cin>>n;

infile.seekg((n-1)\*sizeof(\*this));

infile.read(reinterpret\_cast<char\*>(this), sizeof(\*this));

show\_data();

}

void account\_query::edit\_rec()

{

int n;

fstream iofile;

iofile.open("record.bank", ios::in|ios::binary);

if(!iofile)

{

cout<<"\nINVALID NOT FOUND!!"<<endl;

return;

}

iofile.seekg(0, ios::end);

int count = iofile.tellg()/sizeof(\*this);

cout<<"\n There are "<<count<<" record in the file";

cout<<"\n Enter Record Number to edit: ";

cin>>n;

iofile.seekg((n-1)\*sizeof(\*this));

iofile.read(reinterpret\_cast<char\*>(this), sizeof(\*this));

cout<<"Record "<<n<<" has following data"<<endl;

show\_data();

iofile.close();

iofile.open("record.bank", ios::out|ios::in|ios::binary);

iofile.seekp((n-1)\*sizeof(\*this));

cout<<"\nEnter data to Modify "<<endl;

read\_data();

iofile.write(reinterpret\_cast<char\*>(this), sizeof(\*this));

}

void account\_query::delete\_rec()

{

int n;

ifstream infile;

infile.open("record.bank", ios::binary);

if(!infile)

{

cout<<"\nINVALID FILE...NOT FOUND!!"<<endl;

return;

}

infile.seekg(0,ios::end);

int count = infile.tellg()/sizeof(\*this);

cout<<"\n There are "<<count<<" record in the file";

cout<<"\n Enter Record Number to Delete: ";

cin>>n;

fstream tmpfile;

tmpfile.open("tmpfile.bank", ios::out|ios::binary);

infile.seekg(0);

for(int i=0; i<count; i++)

{

infile.read(reinterpret\_cast<char\*>(this),sizeof(\*this));

if(i==(n-1))

continue;

tmpfile.write(reinterpret\_cast<char\*>(this), sizeof(\*this));

}

infile.close();

tmpfile.close();

remove("record.bank");

rename("tmpfile.bank", "record.bank");

}

int main()

{

Login();

account\_query A;

int choice;

cout<<"\*\*\*WELCOME TO YOUTH CENTRIC BANK\*\*\*"<<endl;

cout<<"\*\*\*RESTRICTED FORM PEOPLE UNDER THE AGE OF 40\*\*\*"<<endl;

cout<<"\*\*\*FOR THE BRIGHT FUTURE AHEARD\*\*\*"<<endl;

while(true)

{

cout<<"SELECT OPTION: ";

cout<<"\n\t1-->DEFAULT ACCOUNT";

cout<<"\n\t2-->ADD RECORD";

cout<<"\n\t3-->SHOW RECORD FROM FILE";

cout<<"\n\t4-->SEARCH RECORD FROM FILE";

cout<<"\n\t5-->UPDATE RECORD";

cout<<"\n\t6-->DELETE RECORD";

cout<<"\n\t7-->QUIT";

cout<<"\nEnter your choice: ";

cin>>choice;

switch(choice)

{

case 1:

A.write\_rec();

break;

case 2:

A.write\_rec();

break;

case 3:

A.read\_rec();

break;

case 4:

A.search\_rec();

break;

case 5:

A.edit\_rec();

break;

case 6:

A.delete\_rec();

break;

case 7:

exit(0);

break;

default:

cout<<"\n CORRECT CHOICE REQUIRED";

exit(0);

}

}

system("pause");

return 0;

}

# **CHAPTER 4: SYSTEM TESTING**

## 4.1 Testing Plan

The testing Scope consisted of a series of different tests that fully exercised the system. The primary purpose of these tests was to uncover the limitations of the systems and measure its full capabilities. A list of the various planned test types and a brief explanation follows. Objective of this test was to ensure that all project functionality works without break. Includes validation Testing-- that is intensive testing of the Front-end fields and screens considering the bank requirements [5].

This test ensured that all the modules were integrated and communicated with each other in the way they are expected to be. This test ensured consistent flow of data across the modules. This test ensured that only authorized users can access the system. This test ensured that the user doesn’t have access to any modules, which they are not supposed to have. This test checked the overall performance of the system and response time for loading screens, creating a record, updating the record and deleting the record. This test ensured the response time of the system in multi-user environment

The test procedure was conducted on a 1-year base from the implementation of the system. The system was introduced to a few employees who were test employees, and their reaction to the new system was observed over this one-year period. The employees accepted the new system with enthusiasm as it had completely defined all the proposed requirements. An addition of 2 months was given to see how the system would work if it were to be used by every employee this made it clear that the new system was fully functional and could be used by the whole bank itself as it became successful

The developer was responsible for the assurance and control processes that formed the basis of the quality plan. The developer had the overall responsibility for quality assurance. This included establishing and maintaining acceptable methodologies, processes, standards and guidelines for the performance of work during all life cycle stages. The team leader was responsible to ensure that all aspects of the Quality Plan are performed and took the appropriate action when the Quality Plan is not followed. Developers themselves were responsible for project quality.

## 4.2 System Testing

Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **TEST CASES** | **EXPECTED RESULT** | **ACTUAL RESULT** | **SUCCESS** | **FAILURE** |
| 1. | LOG IN | To see if the manager and teller have the log in access | Output as expected. |  |  |
|  |  | Admin log in |  |  |  |
|  |  | Teller login to change |  |  |  |
|  |  | Default account |  |  |  |
| 2. | DEFAULT ACCOUNT | Intact information | Output as expected. |  |  |
|  |  | Individual log in |  |  |  |
| 2. | ADD RECORD | Edit information | Output as expected. |  |  |
| 3. | SHOW RECORD | Show record information only. | Input is prompted. |  |  |
|  |  | Login user |  |  |  |
| 4. | SEARCH RECORD | Search requirement. | Output as expected. |  |  |
|  |  | Login user |  |  |  |
| 5. | UPDATE RECORD | Update information. | Output as expected. |  |  |
|  |  | User login |  |  |  |
| 6. | DELETE RECORD | Delete required record. | Output as expected. |  |  |
| 7. | QUIT | Close application | Output as expected. |  |  |

## 4.3 Dump screens

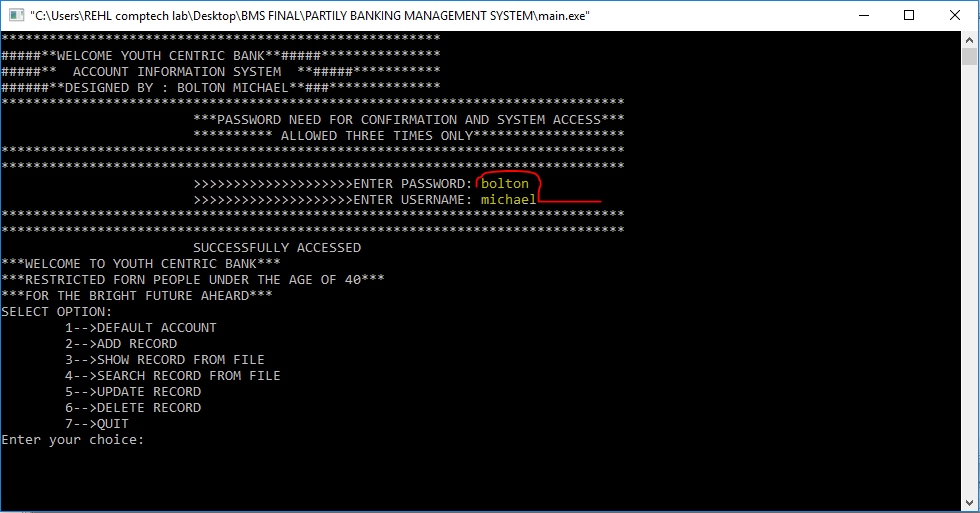


Figure 4.1 log in with correct password

According to the Account Information System, the manager or teller will input the username and password as default inserted in by the programmer. As stated above. If he/ she input the wrong username or password the system will output a message as stated below:

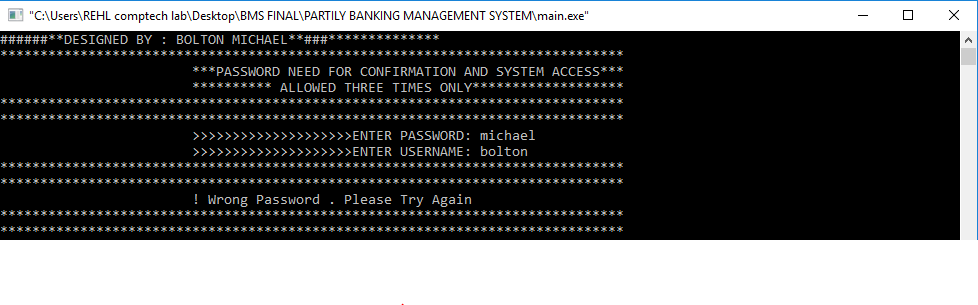


Figure 4.2 log in with wrong password

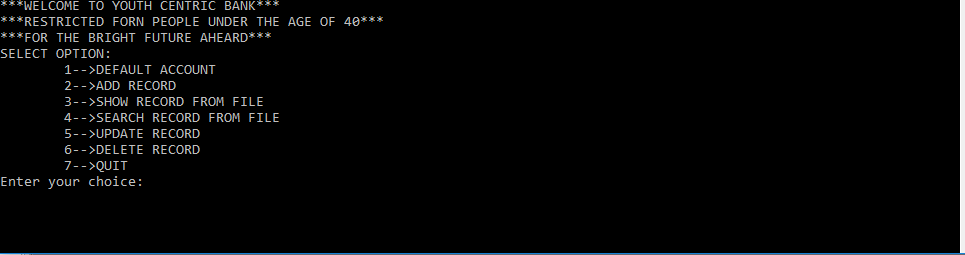


Figure 4.3 Option platform

In this option the manager and teller have the privilege to select an option as stated above:

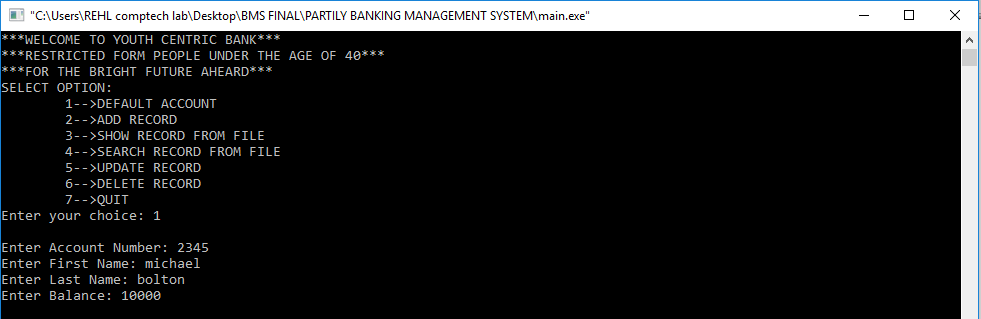


Figure 4.4 Add record

As stated above, the system will roll in on figure 4.1 for default account. In which it will inquire the account number, first and last name then the balance needed as default inserted.

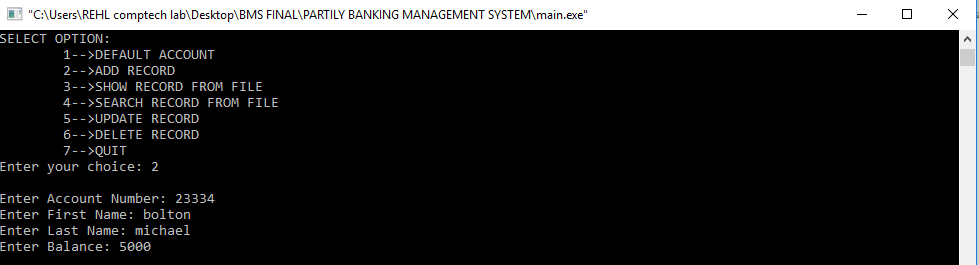


Figure 4.5 Show record

As stated above, the system will roll in on option 2 for add record, similar to figure 4.1.

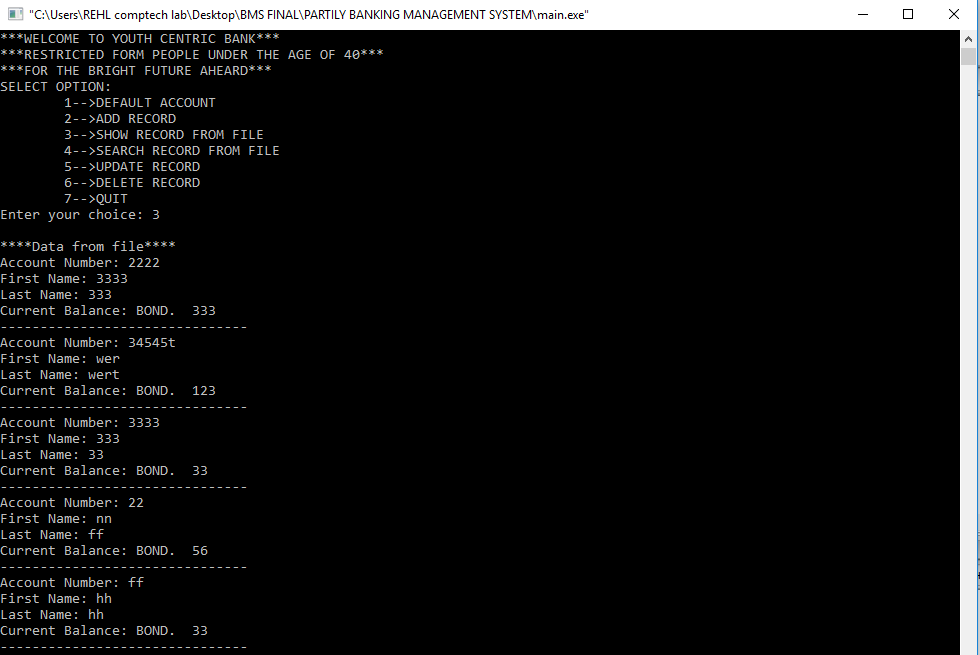


Figure 4.6 Show record

As stated above, the system will roll in on option 3 for show records which have been input into the file during figure 4.1 and 4.2

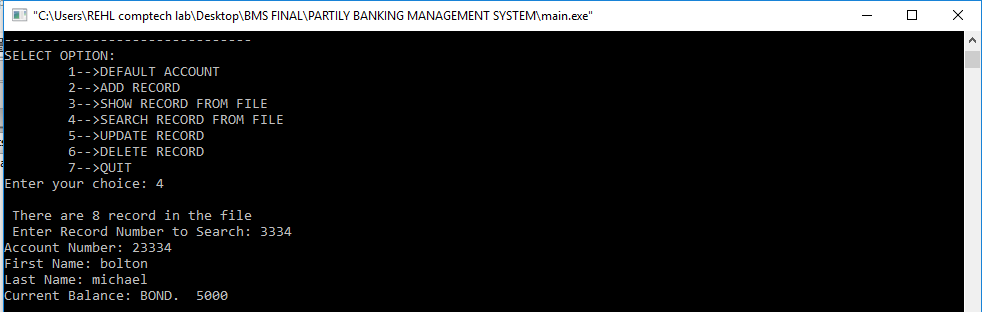


Figure 4.7 Search record

As stated above, the system will roll in on option 4 for search record from file which have been input during figure 4.1 and 4.2. Only a single search on particular record.

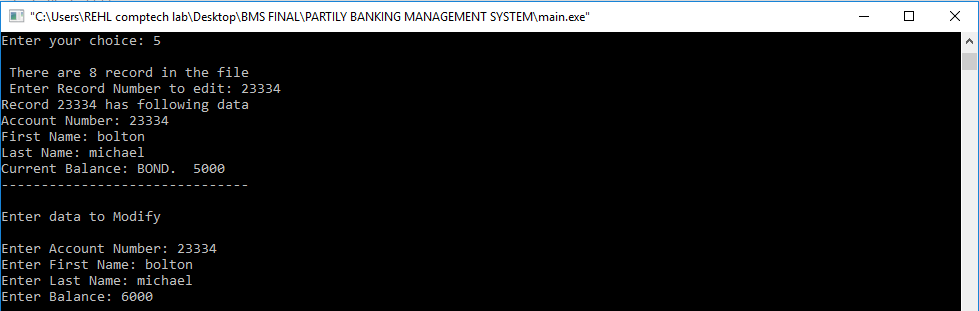


Figure 4.8 Update record

As stated above, the system will roll in on option 5 for update records which have been input into the file during figure 4.1 and 4.2.on this the existing will be update depending on the status of the information to be manipulated.

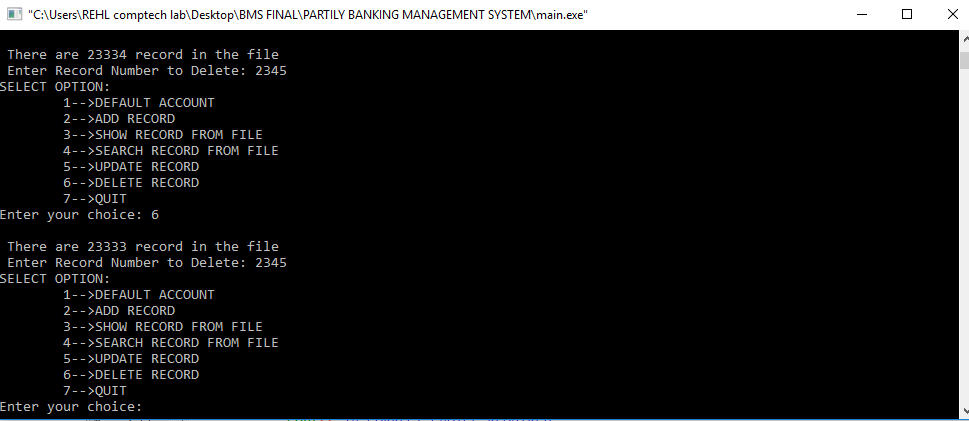


Figure 4.9 Delete record

As stated above, the system will roll in on option 6 for delete records which have been input into the file during figure b4.1 and 4.2 , the teller will either entered in wrongly or unwanted fake information.

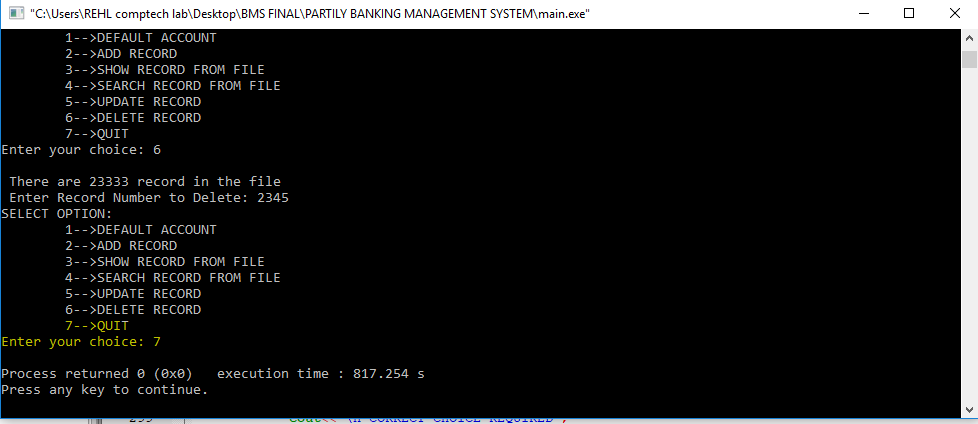


Figure 4.10 Quit

As stated above, the system will roll in on option 7 for quit. Close the application.

## 4.4 System Validation Procedures

The aim of this chapter is to draw conclusions of the work done or achieved and to give an assessment of the completed system, discuss the Problems faced, limitations of the system and give future recommendations on how the system can be improved. The software product produced was fairly good, it achieved most of the user requirements, the user interface is good and is very easy to navigate, and even novice users can find their way around the interface application easily. The client side validation is excellent but with a limitation in password fail. The lack of integration with a compliment is the major drawback and the system was also unable to generate structured reports i.e. reports based on specific information.

### 4.4.1 Problems Faced

The biggest challenge faced was getting hold of Bank Account Information Application. The project was initially aimed at the clients but that proved to be a challenge because client information is very critical. This led to the implementation of a generic system thereby drawing a few assumptions were possible meaning that requirements kept on changing as different views were put into consideration as compared to software which has a specific client.

### 4.4.2 Limitations

In the designed system, the manager can generate a report containing all the client information from personal profile, skills and educational background. Sometimes, the manager may be looking for specific client information e.g. his/ her background, technical skills or language and location in cases where a transfer is to be made, this poses a challenge because will always have to generate a report of all the information about an client, some of which may be not be needed.

### 4.4.3 Future Work

This feature is important because the manger should know which of his/her clients which to go on the note sheet of password access. In order for the system to be more comprehensive, I’d recommend an integration of the system. If further worked on, this functionality can assist in determining the performance based on their ability to finish tasks on time.

### 4.4.4 Information archiving

A system holding all the client information should have some form of archiving system so that retired, suspended or fired employees are archived rather than been completely deleted from the system where an employee did several projects and there details are required for future reference.

### 4.4.5 Learning Experience

This project assisted me to gain a practical experience and apply the knowledge assimilated from the previous courses undertook. Putting the knowledge gained earlier and applying different techniques from past courses was interesting and certain concepts, tools and techniques only made sense after seeing their application in a real world scenario. It was extremely challenging at times but it has been a great and worthwhile learning experience. There is not at all any doubt that the employee management system would be an asset to any company, small or large.

### 4.4.6 Conclusion

In this chapter, the results were discussed, problems faced and limitations were elaborated. Future recommendations for the extension and improvement of the system have also been discussed as well as well as an assessment of achieved functionality.

# REFERNCES

[1] OrangeHRM Open Source, Retrieved: November 4, 2013. From: http://www.orangehrm.com/open-source-product-features-pim.shtml

[2] A.S.SyedNavaz, A.S.SyedFiaz, C.Prabhadevi, V.Sangeetha, S.Gopalakrishnan, “Human Resource Management System”, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 8, Issue 4 (Jan. – Feb. 2013) Page 62-71.

[3] Ian Sommerville, “Software Engineering”, 9th Edition, Addison-Wesley, 2011.

[4] Avison, D. and Fitzgerald, G. (2003).Information systems Development Methodologies, Techniques and Tools.3rd Edition. McGraw-Hill Education Limited Bershire

[5] Zhiming, L, July 2002 ,Object-Oriented Software Development with UML Retrieved:

July 16th 2010. From:http://www.iist.unu.edu/www/docs/techreports/reports/report259.pdf