



Metropolia

Abishek Chandra Mahato

Designing a Banking System

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

10 October 2021

Abstract

Author	Abishek Chandra Mahato
Title	Designing a Banking System
Number of Pages	47
Date	10 October 2021
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Supervisor	Janne Salonen, Head of School of ICT
<p>The purpose of the thesis was to develop a application(desktop) for a business company by using our expertise on IT tools so far. Large amount of research and data collections were done for traditional banking and the modern ones. The main goal was to provide a smoothly running banking system so that general users and the professionals could track the transaction through the business activities. The main technologies/tools to be used were C#, MS SQL, ADO.NET, .Net framework, MS Visual Studio, Windows 10 OS...etc</p> <p>The idea/research is based on the requirements and necessities of the client including the users based on different types of roles of the stakeholders. The features were based on the level of technical knowledge of the users, questionnaires, interviews, presentations, graphical representation and so on.</p> <p>The results of the survey and other data obtaining tools made me believe that banking system is a complicated thing for application that includes number of regulations and legislations. So, we tried to develop a system like it but not as advanced as the real world banking system. It can be taken as a mini version/comceptual of the system by which general users and professionals are able to track the transactions efficiently and act accordingly.</p> <p>Based on the research done, our system and the data can be used as an intermediate representation on how a banking system works with its common features and functionalities. Digitalization of a local business was developed through the development of the proposed system.</p>	
Keywords	WindowsDesktopApplication,Banking managementSystem, MS SQL, Ms visual Studio, C#, ADO.NET, .Net framework

Table of Contents

List of Abbreviations

1	Introduction	6
2	Project Analysis/Specifications	7
2.1	Proposition	8
2.2	Key Sources	8
2.2.1	Meetings	9
2.2.2	Interviews/Data Collection	10
3	System Structural/Functional detailing	10
3.1	UML (Unified Modelling Language) Representation	10
3.1.1	Use Case Diagram	11
3.1.2	Sequence Diagram	14
3.1.3	Activity Diagram	19
3.1.4	Class Diagram	20
3.1.5	Entity Relationship Diagram (ER)/ Database Design	22
3.2	GUI (Graphical User Interface) Design	23
3.2.1	Main Interface	23
3.2.2	Account/Profile Interface	24
3.2.3	Registration interface	25
4	System Implementation	26
4.1	Structure of the Project Implementation	27
4.2	Associated Functions and its implementation	28
4.2.1	User Login	28
4.2.2	User Registration	29
4.2.3	Update and Delete account	32
4.2.4	Perform transactions	33
4.2.5	Display Users Logs	34
5	System/Application Testing	34
5.1	User Login/ Main display	35
5.2	User Registration	38
5.3	Deposit and withdrawal from the bank account	39

5.4	User transactions/Logs	43
6	Conclusion	44
7	Work in Future	45
	References	46

List Of Abbreviations

MS SQL	Microsoft Structured Query Language
ADO.NET	ActiveX Data Objects
UML	Unified Modelling Language
GUI	Graphical User Interface
VS	Visual Studio
DBMS	Database Management System
SHA	Secure Hash Algorithm

1 Introduction

Banking is something what we do daily by direct process or indirect process. It is an important integral part of our life, as money revolves throughout our aspects of our life. Developing from the traditional cash-based system (declining in nature) to digitalization of the money, is one of the fastest ways to interact with external world with money in our hand flying through wires in a blink of an eye. The main aim is to develop a banking system for general customers of an imaginary company.

The key point for banking systems is reliability, security, safe transactions, secure data passages, accessibility and processing of huge amount of data to and from various known or unknown sources. That's a challenging job considering the main data of processing is the real money itself and on top of that comes a governmental legislations/regulation.

The proposed system is supposed to mock a real time banking management system which we/organisations use in daily life. Basically, the system is for an imaginary company with whom we have a deal to develop and implement the banking system for them and ultimately for their customers. The company can be considered as a small-scale bank.

The main objective of the project is to propose a system to the company through different tools and methodologies that we have. We aim to have proper interactions with the stakeholders of the company to understand their necessities, desires, technological level through our motivation, meetings, resources (technical or non-technical), documentations and other industry standards. The system will be developed under .Net framework using tools like C# programming languages, MS visual Studio, Windows 10 OS, MS SQL (data storing and processing), ADO.NET.... etc. Ultimately, we achieve a proper functioning desktop application which has features of a general banking system.

2 Project Analysis/Specifications

On the basis of some various visits and meetings with stakeholders of the company the specifications/requirements from both parties including the budget were concluded.

- End users/customers:

Varieties of end users, front desk officers/cashiers and administrators.

- The scope of the system was analyzed based on their scalability relative to currently running business.
- The system will consist of all the general functions that a bank has especially focusing on the types of transactions and their flow.

1. The proposed system should be based on the level of perceiving those technicalities. From different interviews and questionnaire with end users the level of perceiving and the general required functions were deduced. In general, following functions were deduced:

- Registration and login systems with authentication accountability.
- Hierarchical division of each role and accessibility to the system.
- General and specific function design for each end users.
- Accurate data storage of each processed info/interactions.

- Proper tracking system of each transaction with its authenticity and accountability.

2.1 Proposition

A customized banking system will be developed in a small scale for the company. By analyzing the requirements/specifications of the company, we came-up with following solutions which were done through different meetings, presentations and finalization:

- A system incorporating the general functions of a bank and its services.
- Presenting the working mechanism through UML diagram representation.
- Budget finalization to €15000.
- Time required agreed to 3 months.
- Designing Platform: UML tools.
- Regular updates will be provided to the authorities to know the designing phase.

These analysis and conclusions are based on:

- Different level meetings and interviews with the stake holders and the end users.
- Proper budget finalization based on required human resource and infrastructure.
- The level of expertise and scalability of the project.
- Graphical presentation/simulations of the initial stages of the project.
- Research on small scale financial banking software.

2.2 Key Sources

This is a methodology by which we are using different available resources to us and by the company to understand their real requirements, expertise in

technology, usages, necessities and desires. The main ones are meetings, interviews, data collection and usages.

2.2.1 Meetings

a. Meeting 1:

1. Meeting agenda:

- Knowing the requirements of each stakeholder for each defined role in the proposed system.
- Presenting and agreeing on the available resources & technicalities.
- Budget presentation and agreement for a go ahead.

2. Summary:

At this stage, all the vital stakeholders from lower level to authoritarian level are requested to be present. A lot of decisions and agreement must be finalized on vital entities.

b. Meeting 2:

1. Meeting agenda:

- Presenting the latest progress on the system.
- Familiarizing each role and functions defined to end users.
- UML presentation
- Allowing room for any addition or changes with effective interaction.
- Final Agreement

2. Summary:

Presentation on the interaction of each end user to the system.
Discussion for any entities to be added or excluded or changed.

1.2.2 Interviews/Data Collection

- Firstly, the level of knowledge and usage of technical software by the end users were known. They were given various tools like mini version of different software which could be like the system. Simulations of those usage were created to get a bigger picture.
- Various customized forms were distributed among the users to fill out their thoughts and expectations from the system. Then, they were rendered based on our development.
- Number of presentations were done to visually make the users what they are going to get.

3 System Structural/Functional detailing

After understanding the real requirements and needs by the stakeholders for the system we use design methodology to understand the system in different design angles. All the conceptual designs and graphical interpretations are overviewed.

3.1 UML (Unified Modelling Language) Representation

The UML is a graphical tool used to represent different functions, processes, users and subsystems of a System. They show the interaction between these and the actors. The actors are the one who use different entities of a system.

In our case we have 3 main actors namely, Customer, Cashier (front desk) and Admin. The following are the functional and structural representation of the proposed system.

3.1.1 Use Case Diagram

The diagram basically shows how the actors (Customer, Cashier and Admin) interact with the main system and between each other based on their roles. User stories are defined for each actor, which lists the exact activity done with the system. Finally, there is Use cases which are defined processes/functions for actors. Combining actors, user stories and use cases gives us a Use case diagram. The following are the listings of user stories and use cases for each actor.

User Stories

a. Customer:

- As a customer I want to book an appointment for opening an account and get info on various account types with details.
- I will give my personal details, different customizations for my account and add an opening balance.
- I would like to have options for different electronic cards and choose it.
- I should be able to do my daily transactions like paying my bills online with my login info integrated with security.
- I should be able to get frequent news and updates of the bank, my account as far as concerned.

b. Cashier:

- I should be able to recognize customers data with authentication and register it to the bank database for further processing.
- Allowing customers to choose between different account types.
- Verifying amount of money added or deducted from customers account physically or electronically.
- I need to track all the transactions related to and from customers with its accountability.

- Alerting customers and the admin with concerns for any usual unusual activity by customers.
- I will record all the information concerned with the interaction with customers and send it to admin for final verification.

b. Admin:

- I should be able to verify and authenticate different date of customers and integrate it to the system which was initially passed on by cashier.
- Generating customers and general staff login system with prior registrations.
- Authenticating and allowing each transaction done by customers with accountability.
- Tracking all the general activities of each end users with updates, alert and any required changes.

Use cases:

1. Registration, authentication and verification:

Each data of customers and cashiers should be registered to generate a login system for each defined role with proper documentation. The data should be authenticated and verified by admin for final integration to the system.

2. General and specific functions associated with roles of each end users:

Customers should be able to register itself and get login info for doing monetary transactions. A proper track of those transactions should be done by admin with its accountability. Cashiers should be able to open an account, integrate the customer data and pass those to admin for finalization.

3. Data storage and information processing:

All the interaction between admin, customers and cashier should be structurally stored by each of them based on the roles like online transactions of each customer. All the information should be processed with its authentication, accountability and final verification by admin before the storage process.

Based on the above analysis, the diagram below is formed:

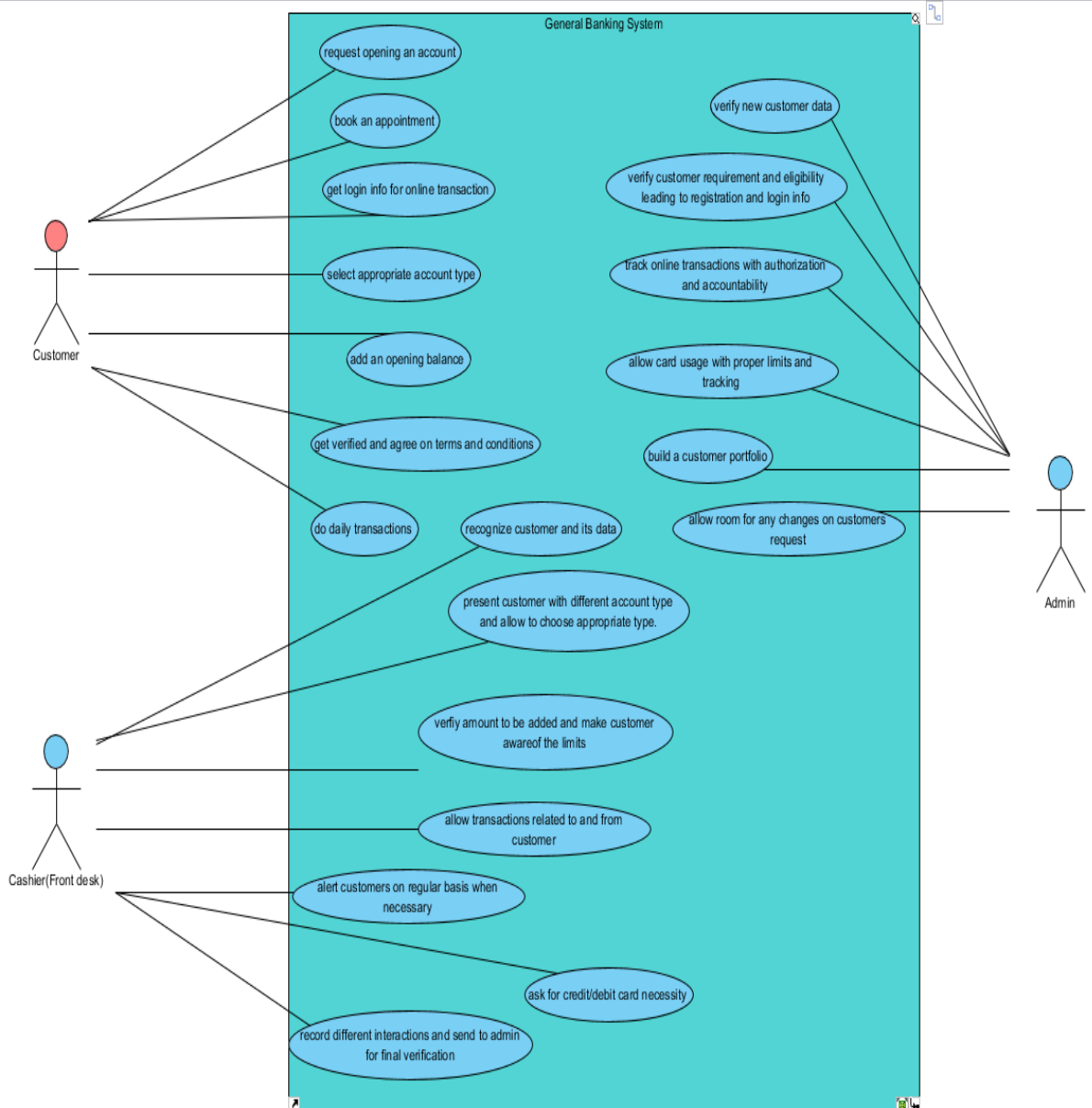


Figure 1. Cashier, Customer and Admin Use case diagram

3.1.2 Sequence Diagram

The diagram simply shows the sequential order interaction between the objects. Event diagrams or event scenarios are also used to refer this diagram. The description of way and order in which the objects function with

the system are shown. It is used by developers for documentation of the requirements of the proposed system or existing system. There are different sequence diagrams based on different activities performed with the systems. Based on the analysis and requirements, they are listed below.

Creating Account Sequence Diagram

This shows how a general user gets his account opened, while it provides the details and further it is verified by the bank servers.

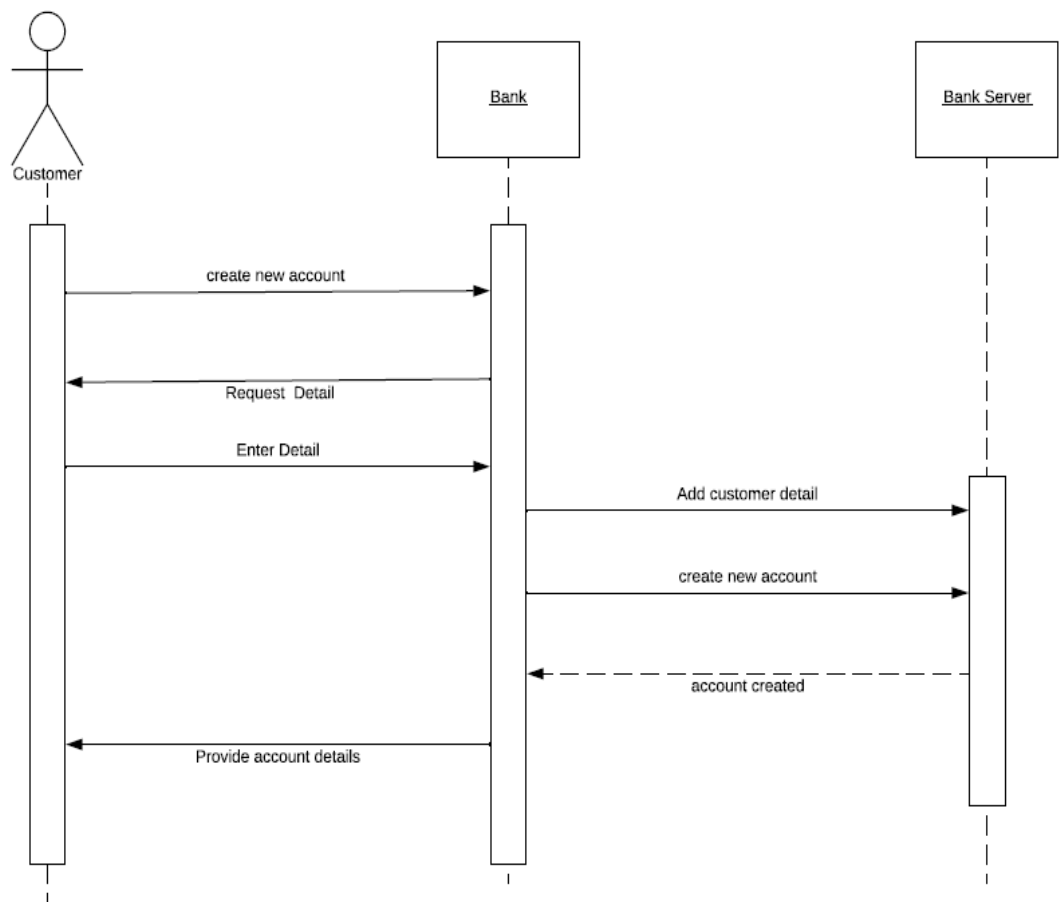


Figure 2. Creating Account Sequence Diagram

Payment Process Sequence Diagram

It shows how a user can do his daily life payments with the available balance under the authentication process of the bank.

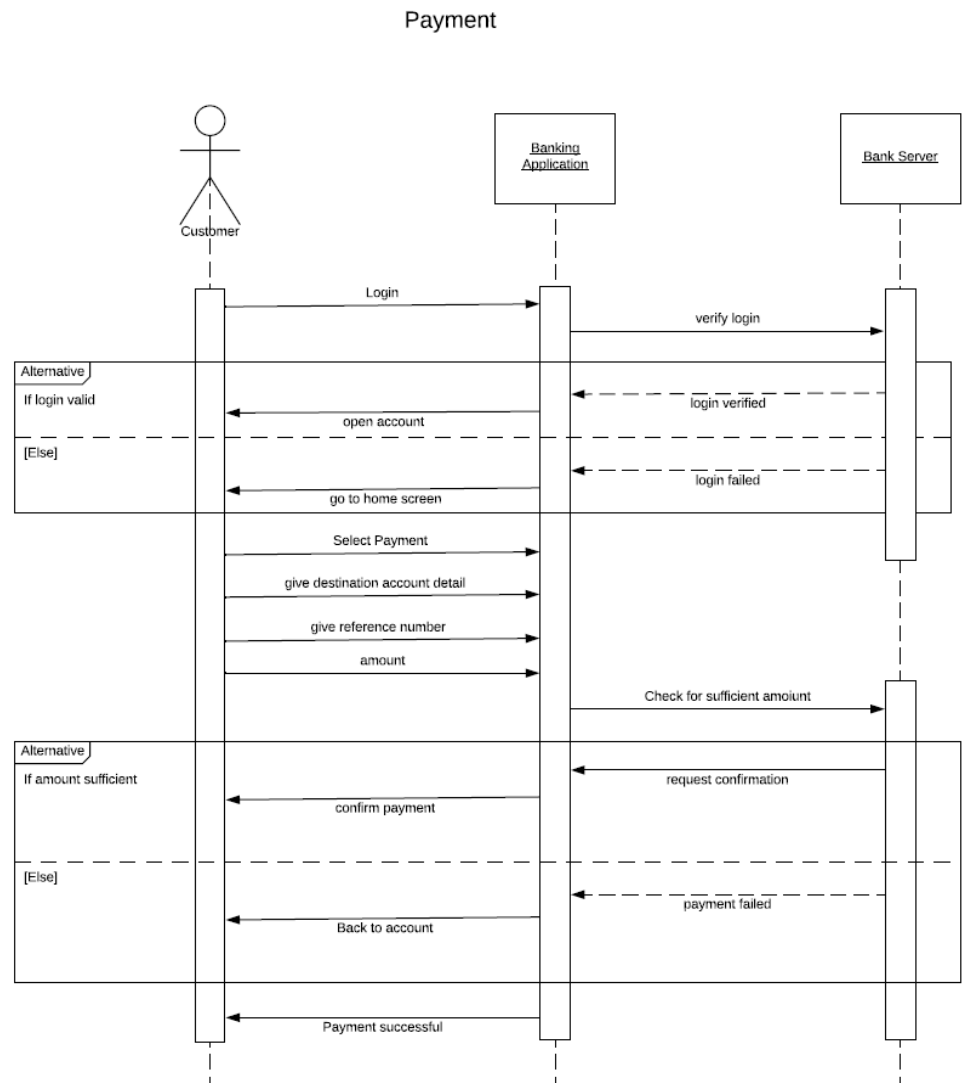


Figure 3. Payment Process Sequence Diagram

Transfer Fund Sequence Diagram

The diagram shows the process in which a user can transfer the money from one account to the other with a proper secured verification from the bank.

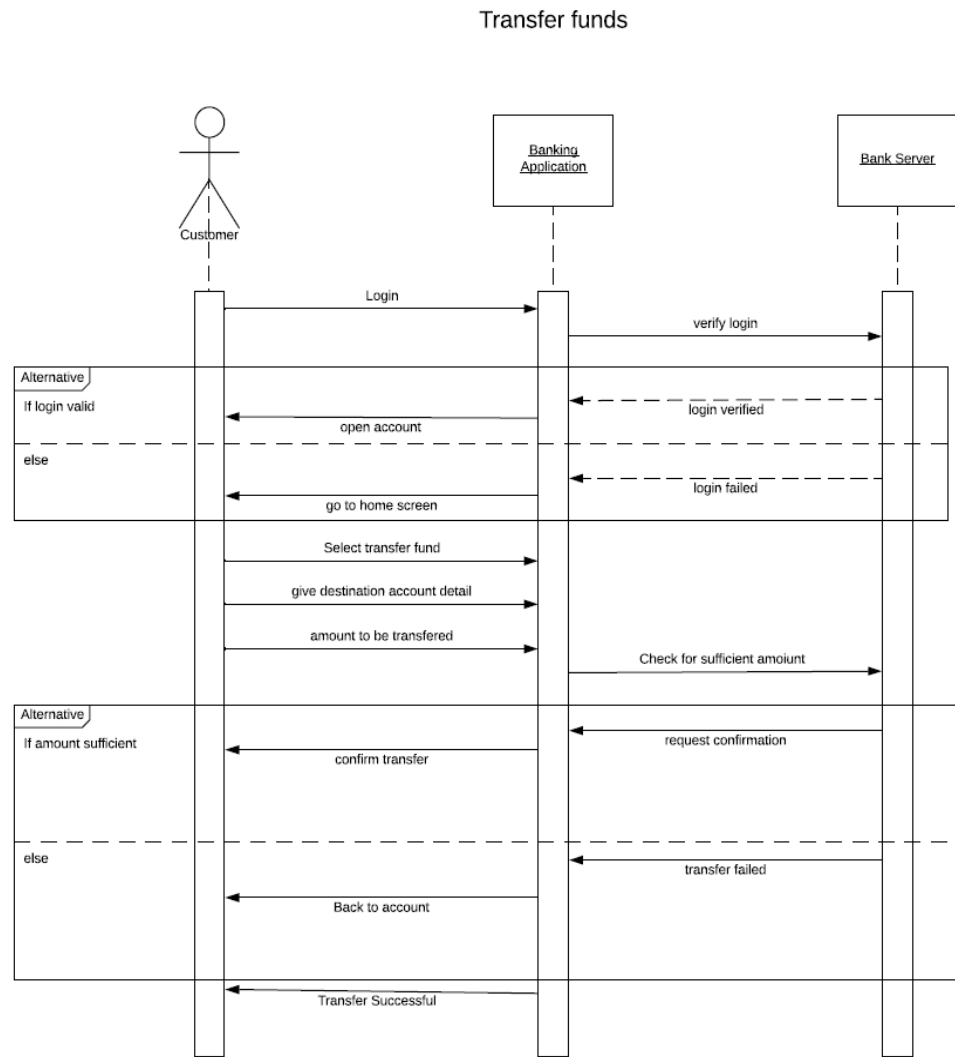


Figure 4. Transfer Funds Sequence Diagram

View Transactions Sequence Diagram

The diagram shows how a user can view all the money matters that has been dealt with the bank.

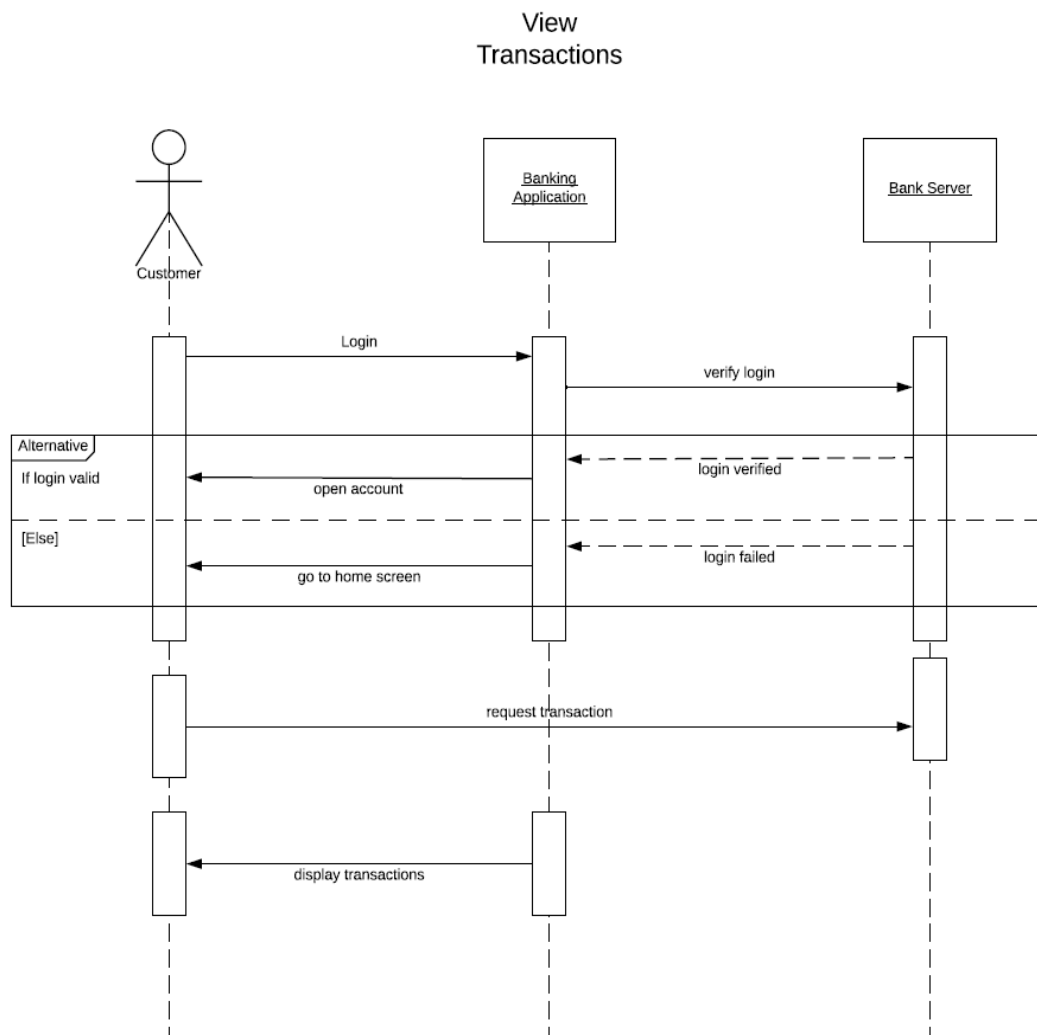
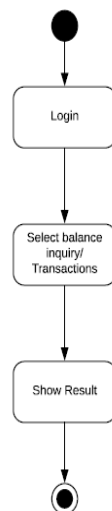


Fig.5 View Transactions Sequence Diagram

3.1.3 Activity Diagram

The diagram presents a behavioural diagram depicting the behaviour of a system. It portrays the control flow from starting point to a finishing point showing the different decision paths that's exists while an activity is in process. Basically, they are used in business and process modelling where the main use is to see the dynamic aspects of a system. Based on the requirements and analysis of the proposed system activity diagrams for balance enquiry, transactions, transfer funds and balance withdraw have been produced. All the figures relating to it are shown below.

Activity diagram for balance inquiry and transactions



Activity diagram for transfer funds

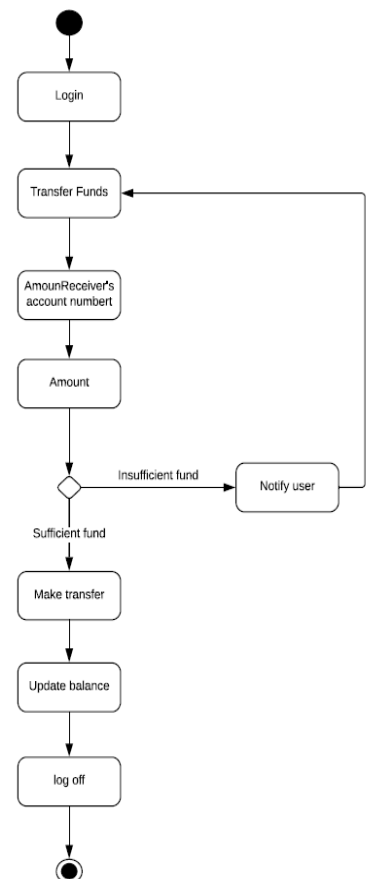


Fig.6 Activity diagrams for balance enquiry, transactions and transfer funds

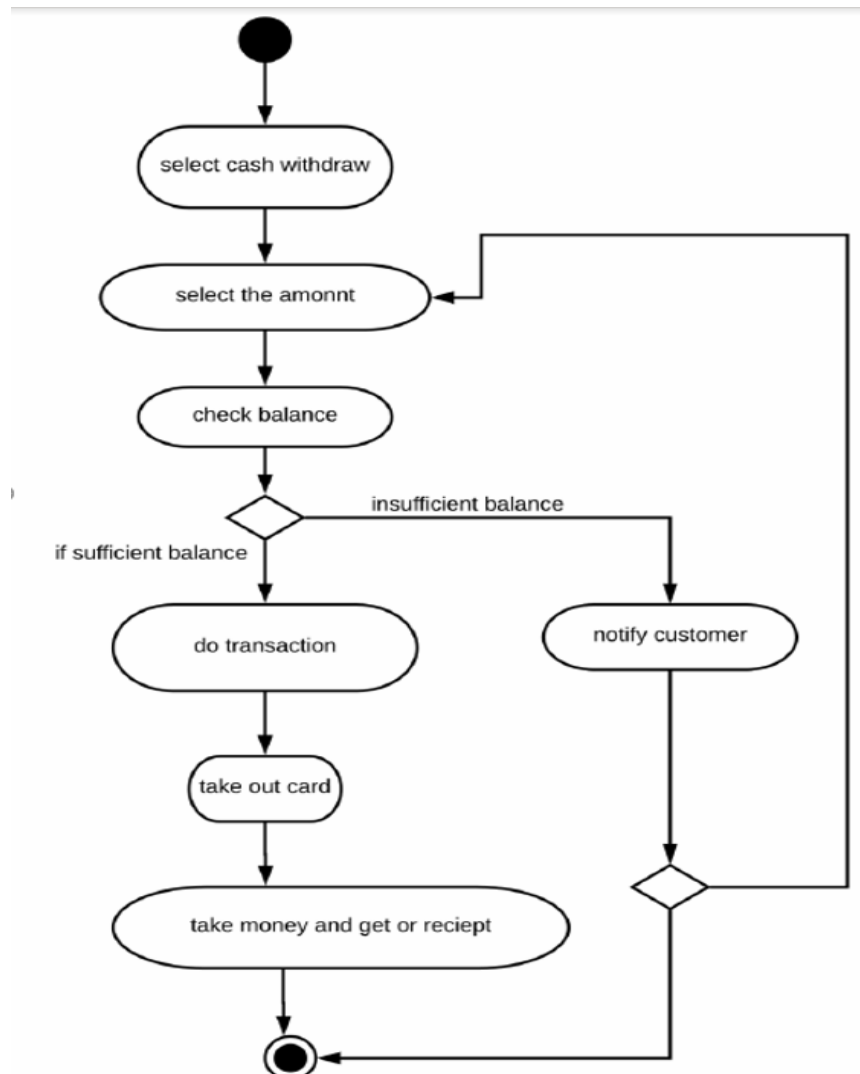


Fig.7 Activity Diagram for Balance Withdrawal

3.1.4 Class Diagram

This diagram is usually considered as building blocks of object-oriented methods. It can be used to show the different classes, relationships, interface, association and collaborations. It presents various kinds of objects and the static relationship between them.

The main idea for the diagram is to:

1. Appropriately depict various aspects of OOPs concept.
2. The design and analysis of a system becomes efficient and faster.
3. It can be considered as a base for deployment and component diagram

In this case there are main entities to be considered like Bank, Cashier, Transfer, Customer, Account, Loan, Current account, Saving account, Fixed account, Account, Deposit. These all entities have their own properties and associated functions. In addition, these all entities are independent classes which have relationships with each other during various processing. Graphically they are represented below.

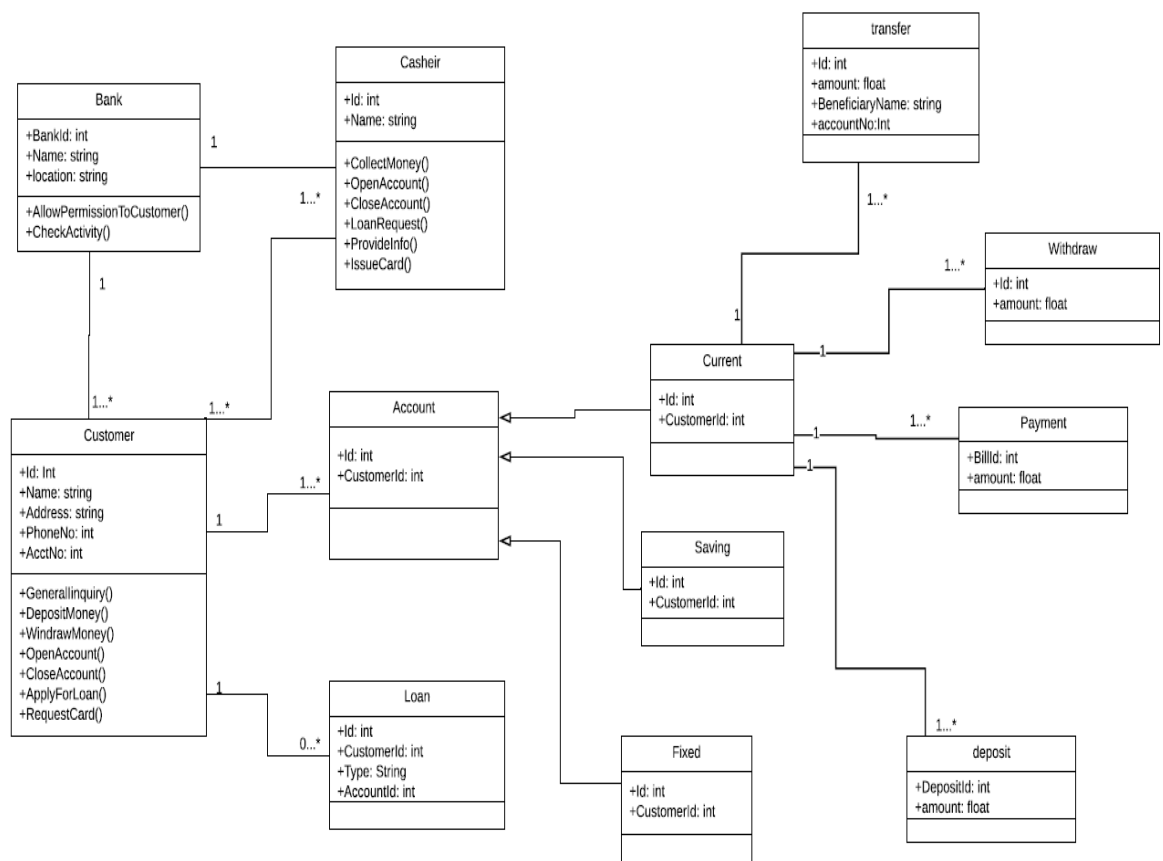


Fig. 8 Class Diagram Bank System

3.1.5 Entity Relationship Diagram (ER)/ Database Design

It is a model that establishes relationships between people, objects, places, concepts or events within a system. It is very vital to project data modelling for the database. It also helps in defining processes and develop relational database. Basically, it helps us to create a database model for our system including different entities revolving in our system.

There are five basic components for this diagram. They are listed below:

- Entities, these are concepts or objects that can have data stored about them. They are referred as tables used in database.
- Attributes, these are properties or characteristics of entities.
- The relationships among and between those entities.
- Actions, that describes how entities share information in the database
- Connecting lines

The following diagram represents the ER model of the banking system that we are developing.

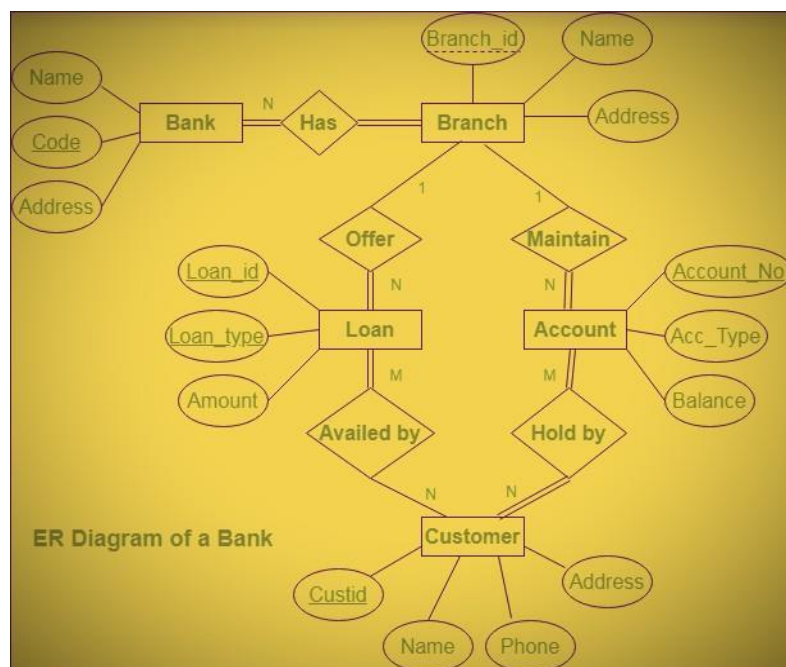


Fig. 8 Entity Relationship Diagram

For the working database the local database from MS SQL server is used for the entire system.

3.2 GUI (Graphical User Interface) Design

It is a type of user interface by which any user interacts with the system via visual indicator presentation. It is considered as use centered design in software programming which basically means that it is an interface for a system graphically design so that a user can access the services as accurately as possible. The users should understand what they are doing and what they are getting from the system as different type of user-friendly responses graphically.

Considering the level of technological knowledge of the stakeholders of the company and the requirements a simple and easily accessible GUI is developed for this system. The GUI is developed using various tools and features available in windows form platform. There are different sections of GUI which are defined and graphically represented below.

3.2.1 Main Interface

It is first interface that opens when we load the application. It consists of a login section where users, cashiers and admin can login into the system. There are other divided sections such as Registration, Exchange Rates, contact us, about us and exit when clicked can go to their own section.

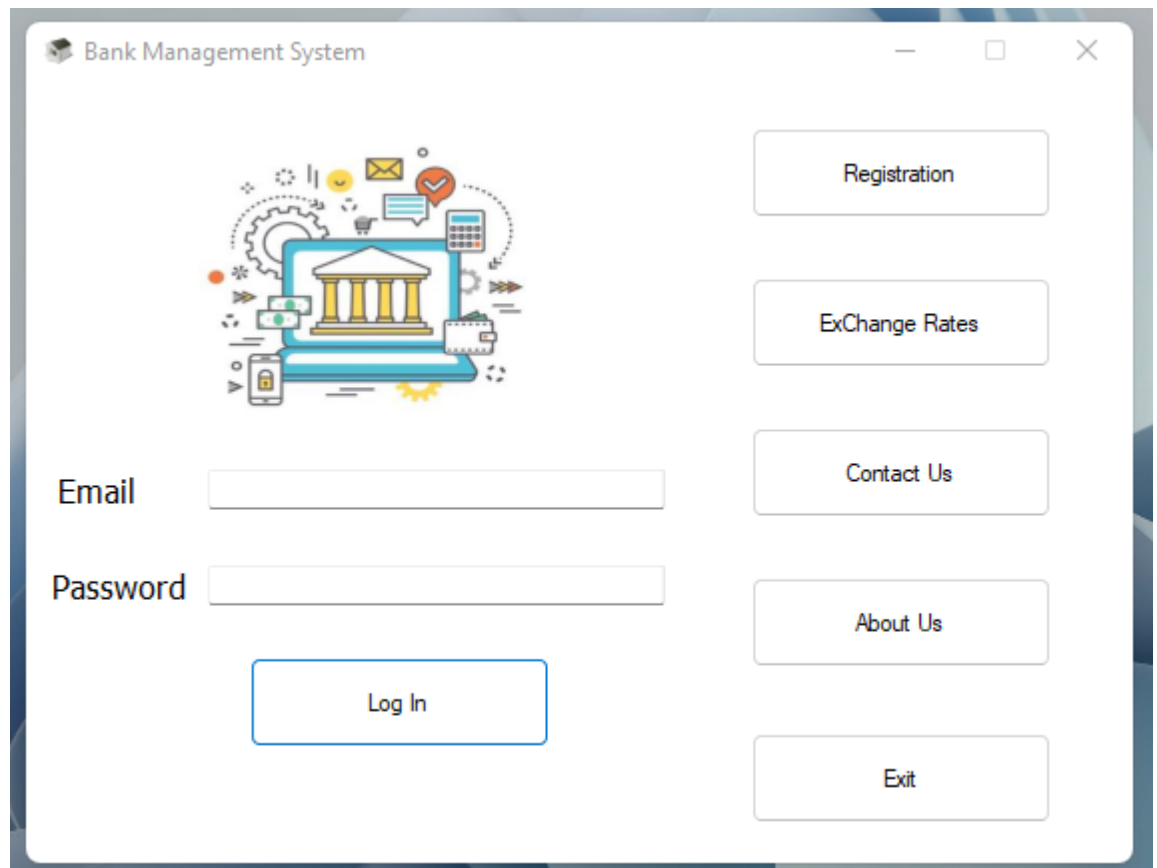


Fig. 9 Main Page interface

3.2.2 Account/Profile Interface

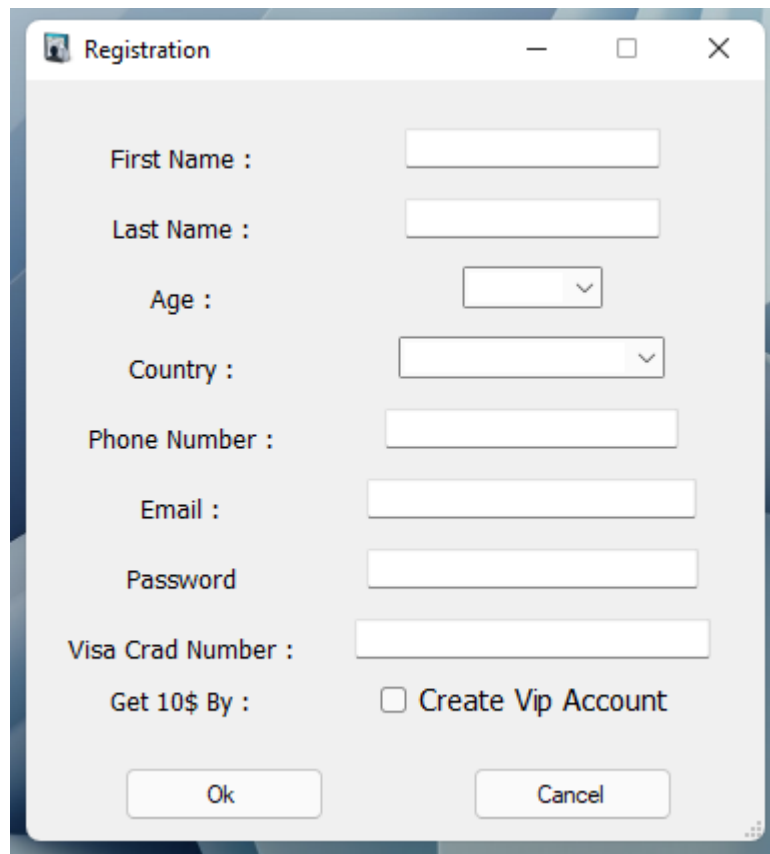
This is the interface user gets when they login from the main page with correct credentials. It consists of sections like delete my account, Deposit, Withdraw, Transfer money and Payment service which are also the functions that are available for the general users. Basic information is also displayed of the user with a log table consisting of all the transactions done.

Log	Value
Deposit To The Account :	500\$
Deposit To The Account :	56256\$
Payment For Electricity/Water Bill :	1000\$
Deposit To The Account :	4525\$
Withdrawal From The Account :	5252\$
Withdrawal From The Account :	89\$
Transfer To Another The Account :	5000\$
Payment For Mobile Recharge :	500\$
Deposit To The Account :	500\$

Fig. 10 Account Interface

3.2.3 Registration interface

The interface allows to register as a user and open an account providing our personal details. Then we are allowed to login to the system.



The image shows a registration window titled "Registration". It contains the following fields and options:

- First Name : [text input]
- Last Name : [text input]
- Age : [dropdown menu]
- Country : [dropdown menu]
- Phone Number : [text input]
- Email : [text input]
- Password : [text input]
- Visa Card Number : [text input]
- Get 10\$ By : Create Vip Account

At the bottom, there are two buttons: "Ok" and "Cancel".

Fig. 11 Registration interface

4 System Implementation

Implementation is a process which outputs a low-level system entity in the system hierarchy of system (System Breakdown Structure) made, bought and re-used. The idea of system implementation designing and creating/fabricating a system element conforming to element's design requirements and/or properties. The element is generated employing proper technologies and practices of the industries. The system process bridges the system definition processes and process of the integration.

4.1 Structure of the Project Implementation

The system has been developed based on .Net framework, C# language and windows form applications. The following picture shows the actual implemented project structure from MS Visual Studio (developing environment).

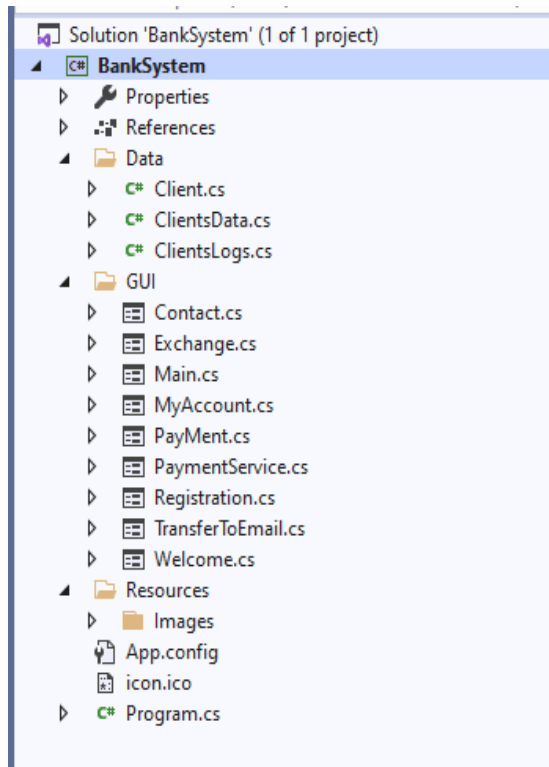


Fig. 12 Implemented Application Structure

Here is a brief description of the contents of each section from above figure

- Data: It deals with storing, manipulating and availing data of users as demanded. In addition, it also creates new data. There are multiple classes to deal with those.
- GUI: It consists of all the required interfaces for all the functionalities including the main interface view.
- Resources: It consists of all the required entities like external images for GUI.
- Program.cs: It is main entry point of the application.

4.2 Associated Functions and its implementation

The implementation of this application has based on functions are user login, registration, view and store user information, delete user, and Transactions by user. All the user graphical interfaces are built under windows forms. The logical implementations have been done by C# language and MS SQL database system. In the following sections, functional descriptions with code snippets are shown.

4.2.1 User Login

The function checks the user credentials of login email and password. Checks with the user registrations in database. If the login credentials match with the registration data, then the user account interface is opened where the user can access or request services from the bank. An error message is popped if the credentials are not correct with the database.

```
public static bool LogInChecker(string Email, string Pass)
{
    int Position = 100*HashFunction(Email);
    FileStream.Seek(Position, SeekOrigin.Begin);
    while (Br.PeekChar() == 'T')
    {
        int Temp = Position;
        Br.ReadChar();
        if (Br.ReadString().ToLower() == Email && Br.ReadString()
== Pass)
        {
            Cur = new Client(Email, Pass, Br.ReadString(),
Br.ReadString(), Br.ReadString(),
Br.ReadString(), Br.ReadInt32());
            CurPosition = Position;
            ClientsLogs.SetAdress(Temp);
            return true;
        }
        Position += 100;
    }
}
```

```

        FileStream.Seek(Position, SeekOrigin.Begin);
    }
    return false;
}

```

Code Snippet 1. User Login

4.2.2 User Registration

Different personnel information is gathered from a user to register them in database. It is also checked that if the user already exists through the email given by the user. All the data provided by the user are properly checked for its integrity and after verifying all of them a new bank account for the user is created.

```

class Client
{
    protected readonly string FirstName, LastName, Age, Email,
Country, VisaCard, Password, PhonNumber;
    protected int Amount = 0;
    public Client(string Email, string Password, string FirstName,
string LastName, string Country,
                string Age, string VisaCard, string PhonNumber, bool
Vip)
    {
        this.FirstName = FirstName;
        this.LastName = LastName;
        this.VisaCard = VisaCard;
        this.Email = Email;
        this.Country = Country;
        this.Age = Age;
        this.Password = Password;
        this.PhonNumber = PhonNumber;
        if (Vip) Amount = 10;
    }
    public Client(string Email, string Password, string FirstName,
string LastName, string Country,

```

```

        string Age, string VisaCard, string PhonNumber, int
Amount)
    {
        this.FirstName = FirstName;
        this.LastName = LastName;
        this.VisaCard = VisaCard;
        this.Email = Email;
        this.Country = Country;
        this.Age = Age;
        this.Password = Password;
        this.PhonNumber = PhonNumber;
        this.Amount = Amount;
    }
    public string FName
    {
        get { return FirstName; }
    }
    public string LName
    {
        get { return LastName; }
    }
    public string EM
    {
        get { return Email; }
    }
    public string Ctry
    {
        get { return Country; }
    }
    public string Ag
    {
        get { return Age; }
    }
    public string Pass
    {

```

```

        get { return Password; }
    }
    public string Visa
    {
        get { return VisaCard; }
    }

    public string Phone
    {
        get { return PhonNumber; }
    }
    public int Am
    {
        set { Amount = value; }
        get { return Amount; }
    }
}

public static void AddUser(Client User)
{
    int Position = 100*HashFunction(User.EM.ToLower());
    FileStream.Seek(Position, SeekOrigin.Begin);
    while (Br.PeekChar() == 'T')
    {
        FileStream.Seek(100, SeekOrigin.Current);
    }
    Bw.Write('T');
    Bw.Write(User.EM);
    Bw.Write(User.Pass);
    Bw.Write(User.FName);
    Bw.Write(User.LName);
    Bw.Write(User.Ctry);
    Bw.Write(User.Ag);
    Bw.Write(User.Visa);
    Bw.Write(User.Phone);
    Bw.Write(User.Am);
    Bw.Flush();
}

public static bool Registered(string Email)
{
    int Position = 100*HashFunction(Email);
    FileStream.Seek(Position, SeekOrigin.Begin);

```

```

        while (Br.PeekChar() == 'T')
        {
            Br.ReadChar();
            if (Br.ReadString().ToLower() == Email)
            {
                return false;
            }
            Position += 100;
            FileStream.Seek(Position, SeekOrigin.Begin);
        }
        return true;
    }
}

```

Code Snippet 2. Creating and verifying a new User

4.2.3 Update and Delete account

The user can edit or delete its own data if there is a necessity. A user can delete its whole account too. Before allowing editing or deletion, the data are verified and then taken proper actions as allowing or disallowing.

```

public static void UpdateAccount()
{
    Bw.Seek(-4, SeekOrigin.Current);
    Bw.Write(Cur.Am);
    Bw.Flush();
}

public static void DeleteAccount()
{
    Bw.Seek(CurPosition, SeekOrigin.Begin);
    Bw.Write('F');
}

```

Code Snippet 3. Update or Delete an account

4.2.4 Perform transactions

A user can do different transactions through bank services such as money transfer, pay bills or invoices. The following code allows to do that.

```
public static bool Transfer(string RecieverEmail,int Amount)
{
    long Temp = FileStream.Position;
    int Position = 100 * HashFunction(RecieverEmail);
    FileStream.Seek(Position, SeekOrigin.Begin);
    while (Br.PeekChar() == 'T')
    {
        Br.ReadChar();
        if (Br.ReadString().ToLower() == RecieverEmail)
        {
            Br.ReadString(); Br.ReadString(); Br.ReadString();
Br.ReadString();

            Br.ReadString(); Br.ReadString(); Br.ReadString();
            int RecieverAmount = Br.ReadInt32();
            RecieverAmount += Amount;
            Bw.Seek(-4, SeekOrigin.Current);
            Bw.Write(RecieverAmount);
            FileStream.Position = Position;
            return true;
        }
        Position += 100;
        FileStream.Seek(Position, SeekOrigin.Begin);
    }
    FileStream.Position = Position;
    return false;
}
```

Code Snippet 4. Perform different transactions.

4.2.5 Display Users Logs

The user can see each activity performed at any given time in the system. For e.g., a user wants to see how much it spent money on paying a particular bill. So, they all are displayed.

```
public static void GetLogs(DataGridView dg)
{
    while (br.PeekChar() == '|')
    {
        br.ReadChar();
        dg.Rows.Add(br.ReadString(), br.ReadString());
    }
}
public static void AddLog(string s1,string s2)
{
    bw.Write('|');
    bw.Write(s1);
    bw.Write(s2);
    bw.Flush();
}
```

Code Snippet 5. Display Transaction/Logs

5 System/Application Testing

Even after intensive analysis of system requirements and implementation the real part comes up which is Actually testing the application implemented. The testing is done through various user interactions through the GUI and getting results from the backend implemented logics. We try to see that the implementation of GUI and functions have worked or not in real time interactions with the systems. Below, there are different graphical interactions and their results. The application was tested under the platform of windows 10 using Visual Studio 2019.

5.1 User Login/ Main display

The application should load the main GUI displaying different functionality and user login interface.

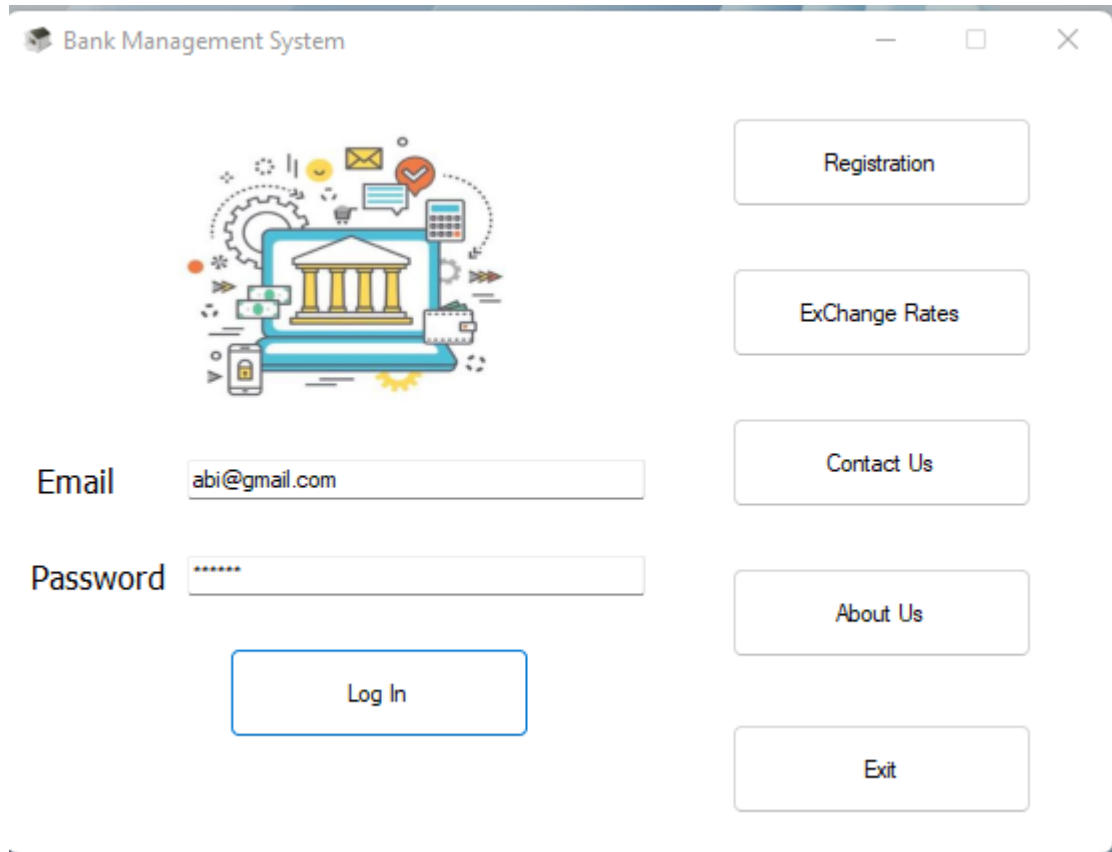


Fig 12. Main interface Display

The user can type its email and password, and then gets into its bank account profile if the credentials are correct.

The screenshot shows a web browser window titled "My Account". The user's email address is **abi@gmail.com**. The user's name is **Abishek Mahato**. The profile details are as follows:

- Age : **30**
- Country : **Finland**
- Phone Number : **0452163660**
- Crad Number : **1254874852**
- Amount : **401\$**

On the right side of the profile, there are five buttons: "Delete My Account", "Deposit", "Withdrawal", "Transfer Money", and "Payment Service".

Below the profile details is a table with two columns: "Log" and "Value".

Log	Value
▶ Deposit To The Account :	500\$
Deposit To The Account :	56256\$
Payment For Electricity/Water Bill :	1000\$
Deposit To The Account :	4525\$
Withdrawal From The Account :	5252\$
Withdrawal From The Account :	89\$
Transfer To Another The Account :	5000\$
Payment For Mobile Recharge :	500\$
Denosit To The Account :	500\$

Figure 13. User profile after correct Login

If the user enters incorrect email address or password, then following interface is displayed.

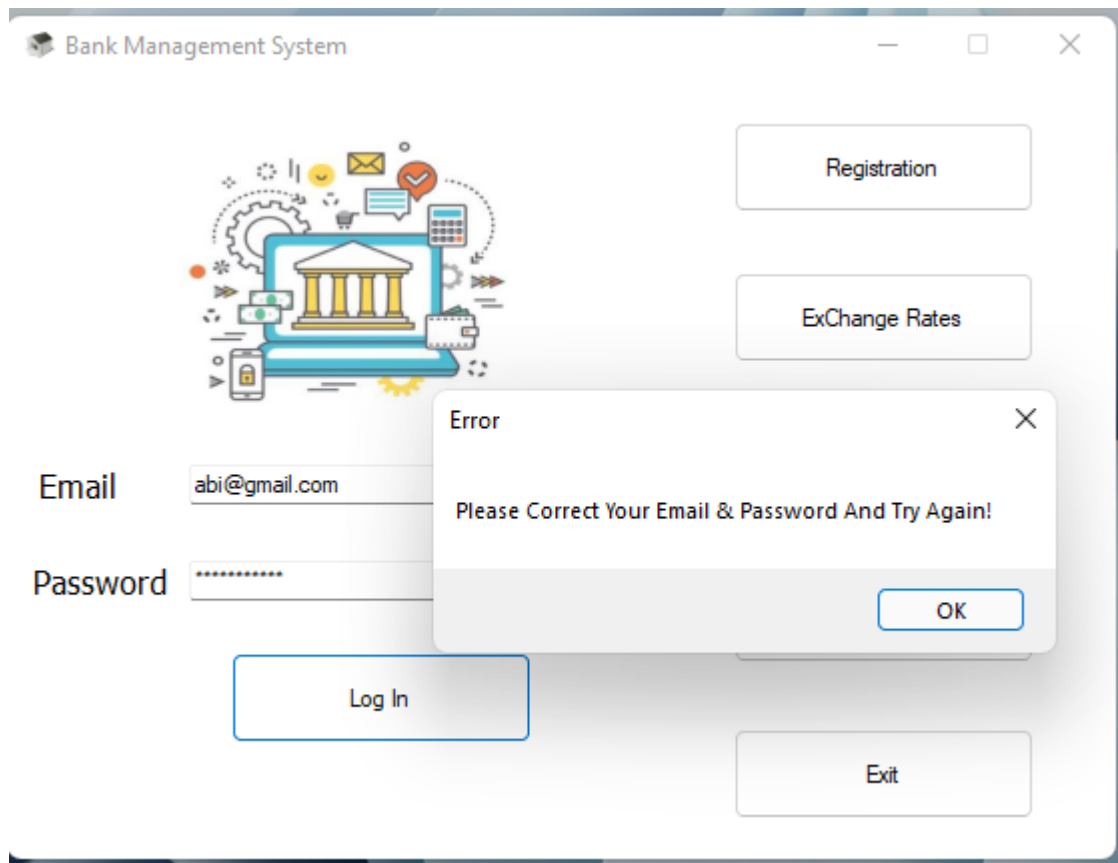
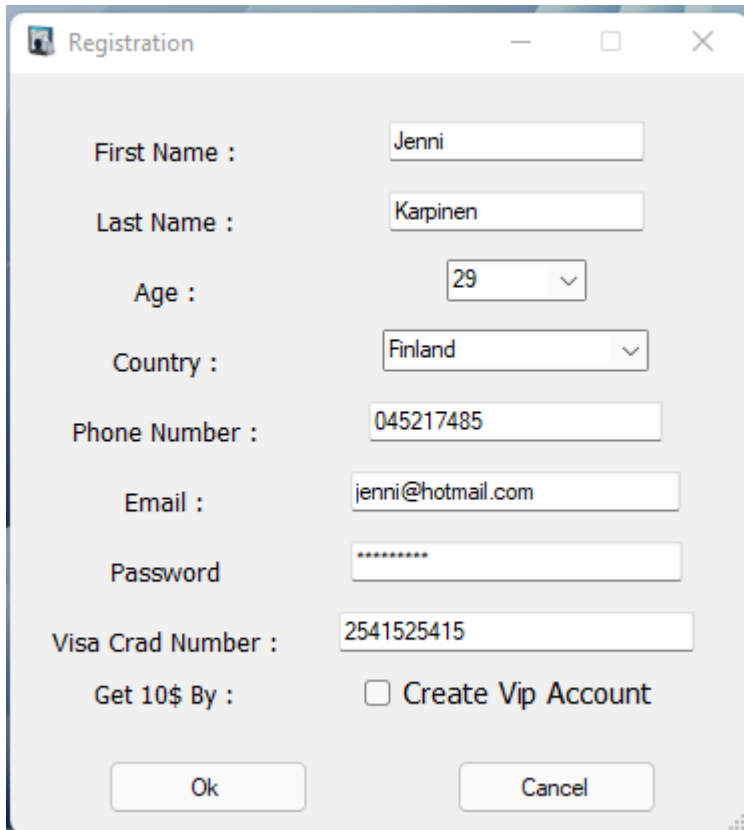


Figure 14. Error display, Login failed

5.2 User Registration

A user registration form should be displayed when clicked on registration section on the main display.



The image shows a screenshot of a web browser window titled "Registration". The form contains the following fields and options:

- First Name : Jenni
- Last Name : Karpinen
- Age : 29 (dropdown menu)
- Country : Finland (dropdown menu)
- Phone Number : 045217485
- Email : jenni@hotmail.com
- Password : [masked with asterisks]
- Visa Card Number : 2541525415
- Get 10\$ By : Create Vip Account

At the bottom of the form, there are two buttons: "Ok" and "Cancel".

Figure 15. User registration interface

After entering all the required information by the user, a message pops up saying the user has been registered, now it can log in.

The image shows a software window titled "Registration" with a standard Windows-style title bar (minimize, maximize, close buttons). Inside the window, there are five input fields: "First Name" with the value "Jenni", "Last Name" with "Karpinen", "Age" with a dropdown menu showing "29", "Country" with a dropdown menu showing "Finland", and "Phone Number" with "045217485". At the bottom of the window are "Ok" and "Cancel" buttons. A smaller dialog box titled "Congratulations" is overlaid on top of the registration form. It contains the text "You Have Been Registered Successfully You Can Login Now!" and an "OK" button.

Figure 16. Registration success

5.3 Deposit and withdrawal from the bank account

A user should be able to take out and deposit balance from its profile.

My Account

abi@gmail.com Delete My Account

Abishek Mahato Deposit

Age : **30**

Country : **Finland** Withdrawal

Phone Number : **0452163660** Transfer Money

Crad Number : **1254874852** Payment Service

Amount : **401\$**

Log	Deposit	5000\$
Depo		0\$
Depo	Please Enter Your Deposit Value :	256\$
Paym		00\$
Depo	50	25\$
Withd		52\$
Withd		\$
Transfer To Another The Account :		5000\$
Payment For Mobile Recharge :		500\$
Denosit To The Account :		500\$

Figure 17. Entering the deposit amount.

My Account

abi@gmail.com

Abishek Mahato

Age : **30**

Country : **Finland**

Phone Number : **0452163660**

Crad Number : **1254874852**

Amount : **451\$**

Delete My Account

Deposit

Withdrawal

Transfer Money

Payment Service

Log	Value
▶ Deposit To The Account :	500\$
Deposit To The Account :	56256\$
Payment For Electricity/Water Bill :	1000\$
Deposit To The Account :	4525\$
Withdrawal From The Account :	5252\$
Withdrawal From The Account :	89\$
Transfer To Another The Account :	5000\$
Payment For Mobile Recharge :	500\$
Deposit To The Account :	500\$

Figure 18. Showing the Deposited balanced

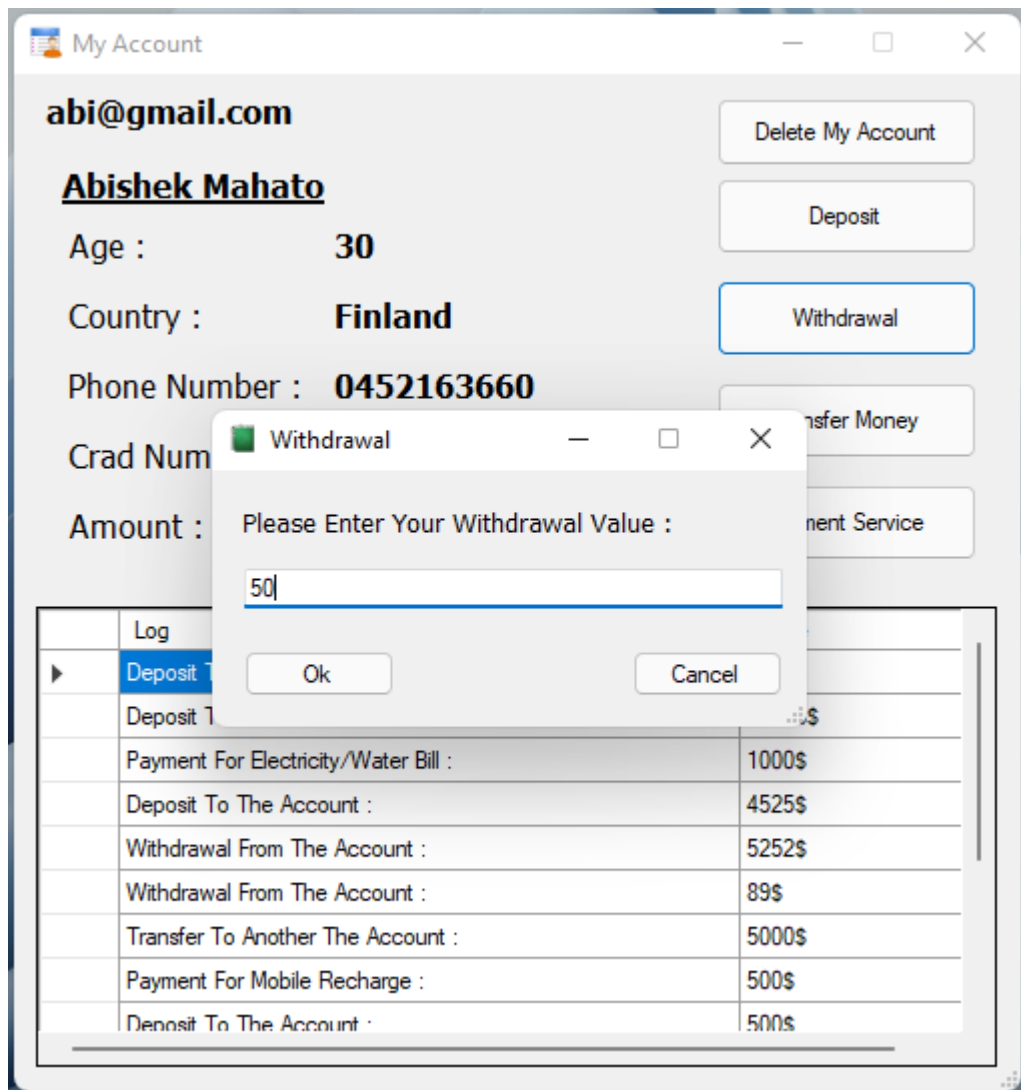


Figure 19. Showing the withdrawing amount.

The screenshot shows a web browser window titled "My Account". The user's email is **abi@gmail.com**. The user's name is **Abishek Mahato**. The user's details are: Age: **30**, Country: **Finland**, Phone Number: **0452163660**, and Crad Number: **1254874852**. The current balance is **401\$**. On the right side, there are buttons for "Delete My Account", "Deposit", "Withdrawal", "Transfer Money", and "Payment Service". Below the details is a table with two columns: "Log" and "Value".

Log	Value
▶ Deposit To The Account :	500\$
Deposit To The Account :	56256\$
Payment For Electricity/Water Bill :	1000\$
Deposit To The Account :	4525\$
Withdrawal From The Account :	5252\$
Withdrawal From The Account :	89\$
Transfer To Another The Account :	5000\$
Payment For Mobile Recharge :	500\$
Deposit To The Account :	500\$

Figure 20. Showing the Balance after withdrawal

5.4 User transactions/Logs

All the user activities/transactions are recorded and are available for viewing for future references.

Amount : **401\$** Payment Service

Log	Value
Payment For Mobile Recharge :	500\$
Deposit To The Account :	500\$
Withdrawal From The Account :	256\$
Deposit To The Account :	500\$
Withdrawal From The Account :	99\$
Deposit To The Account :	50\$
Withdrawal From The Account :	50\$
*	

Figure 21. User Logs/Transactions

6 Conclusion

The thesis project is directed to develop a system, which acts as an efficient and user friendly in giving the real time banking services. The aim is to satisfy different types of users including the stakeholders of the company. The application has provided different features, the main one being the transactions. A general user is continuously doing different transaction in the daily life. So, a very interactive graphical interface is provided so that the user can interact with the system properly and perform the desired activity. The general customers can open the bank account, get verified, check transaction deposit and withdraw money at any given time. The customers are also able to contact the bank. The banks stake holders such as administrators and Cahiers can view each transaction of each customer and give them proper guidance and security in each step. For achieving these goals, mainly the Microsoft technologies were used, like C# programming language, MS visual Studio, Windows OS platform, MS SQL, ADO.NET and so on. In addition to maintain the security credentials, encryption technology SHA-512 was used.

During the development, one of the main challenging parts was to analyse and conclude the requirements of each stakeholders involved. Different stakeholders of the company including the customers have different technological knowledge, so to be fair to each one, such a system must be thought upon where everyone feels comfortable. The interaction between the users and the interfaces of the banking services with proper applied working functions was thought upon a lot. In brief, finally a considerable and interactive banking management system was developed which focused on efficiency, security and way of use by different categories of users.

7 Work in Future

In addition to the system developed there are many shortcomings, if we compare it with the real-world banking systems. The legislation of the government is the main thing that needs to be considered while developing a bank system. An approval from government is needed of developing such systems. The security part of it also needs to be very strong, as it involves money of people. Securing each user, keeping track of all type of transactions and at the same time providing efficient available services is a big challenge. Additional features can be included such as loan management. In current context the system itself needs to be developed using web technologies, which makes it more efficient in accessibility by any users. Further, there are business perspective of developing system including the business constraints depending on the type of company.

References

1. Banking System: Definition & Types. Accessed 10 October 2021.
<https://study.com/academy/lesson/banking-system-definition-types.html>
2. The First 7 Steps You Must Take for Banking Software Development. Accessed 10 October 2021.
<https://cybercraftinc.com/the-first-7-steps-you-must-take-for-banking-software-development/>
3. Software Engineering | Software Design Process. Accessed 10 October 2021.
<https://www.geeksforgeeks.org/software-engineering-software-design-process/>
4. Software development process. Accessed 11 October 2021.
https://en.wikipedia.org/wiki/Software_development_process
5. The Unified Modelling Language. Accessed 15 October 2021.
<https://www.uml-diagrams.org/>
6. Types of UML Diagrams. Accessed 15 October 2021.
<https://www.lucidchart.com/blog/types-of-UML-diagrams>
7. Banking Web Application - GUI Design Sample. Accessed 20 October 2021.
<https://www.carettasoftware.com/gallery/banking-design.html>
8. Banking Software Application Project in c#. Accessed 22 October 2021.
<https://dotprogramming.blogspot.com/2016/01/banking-software-project-in-c-with.html>
9. The Evolution of Banking Over Time. Accessed 30 October 2021.
<https://www.investopedia.com/articles/07/banking.asp>

