Development of a Speed Cash System

Prepared By

Group 14:

Akash Kumar Roy	(Reg. No. 141220110007 of 2014-15)
Chandrayan Sengupta	(Reg. No. 141220110018 of 2014-15)
Debanjan Choudhury	(Reg. No. 141220110019 of 2014-15)
Diptak Banerjee	(Reg. No. 141220110024 of 2014-15)

INDEX

Contents Page No.

1. Abstract

1.1. Problem Statement

Design and development of Speed Cash System [SCS]

1.2. Objective of the project

The Speed Cash System is used to transfer money from one place to another within a day. This is basically used to speed up the money transfer. The necessary information for the money transfer from the source bank to the destination bank is sent in the form of file on daily basis. This file contains the information like remitter details, beneficiary details and DD (Demand Draft) details, etc.

Basically, the remitter is a person who sends the money and the beneficiary is the person who receives the money. If the remitter has already an account with the bank, the deduction at the back end should happen instead of cash dealings. Once the file is received, it is processed and the data is put into the database. Then it is again processed and DD is printed. The printed DD will be handed over to the concerned person.

2. Introduction

With the onset of the Information Age, our nation is becoming increasingly dependent upon network communications. Computer-based technology is significantly impacting our ability to access, store, and distribute information. Among the most important uses of this technology is electronic commerce: performing financial transactions via electronic information exchanged over telecommunications lines. A key requirement for electronic commerce is the development of secure and efficient electronic payment systems. The need for security is highlighted by the rise of the Internet, which promises to be a leading medium for future electronic commerce.
Electronic payment systems come in many forms including digital checks, debit cards, credit cards, and stored value cards. The usual security features for such systems are privacy (protection from eavesdropping), authenticity (provides user identification and message integrity), and no repudiation (prevention of later denying having performed a transaction).
The main aim of developing this Speed Cash Maintenance System is to provide effective money transfer tool for customers in banking system. This project has implemented on visual C#.net and SQL Server. The main advantage of this final year system is to send money with in seconds in intranet and over the internet.
This project looks like user friendly way and any new customer who doesn't know computer basics can access this tool to send money to small towns and rural areas. The previous developed Cash system worked with MS Access database, so large amount of data storage is not possible and it was available in banking branch offices only.

Electronic Payment:

The term electronic commerce refers to any financial transaction involving the electronic transmission of information. The packets of information being transmitted are commonly called electronic tokens. One should not confuse the token, which is a sequence of bits, with the physical media used to store and transmit the information. We will refer to the storage medium as a card since it commonly takes the form of a wallet-sized card made of plastic or cardboard. (Two obvious examples are credit cards and ATM cards.) However, the "card" could also be something like a computer memory.

Software Requirements Specification

Software Requirements Specification

For

Speed Cash System

Version 1.0 approved

Prepared by

Diptak Banerjee (05)

Debanjan Choudhury (15)

Akash Kumar Roy (58)

Chandrayan Sengupta (68)

St.Thomas' College of Engineering and Technology

13th May, 2018

3. System Requirement Specifications

3.1. Introduction

3.1.1. Purpose

This Software Requirement Specification document presents a detailed description of the Speed Cash System. The aim is to provide banking application with quick access to the customers when transferring cash.

3.1.2. Document Convention

This document follows the IEEE format standard.

3.1.3. Project Scope

The purpose of this system is to provide the customers quick access to his or her banking actions when transferring cash from one bank account to another

3.2. Overall Description

The following description gives the general factors that affect system and its requirements,

- Users should have a proper valid bank account number to access the application.
- Only the admin has the full permission to update user information in the application, on user's request.
- Only the admin has the full permission to update user information in the application, on user's request.
- If a user enters the application, all details about the user's bank account will be added to the application's database.

3.3. Product Perspective

- Bank and Admin play a major role in the process.
- The necessary details of the user have to be submitted to the admin.
- Verification has been done by the bank and admin.
- User should be asked to validate his/her account number.

3.4. Actor Documentation

User Login

First to enter this system the users has to login to this system. Basically there are 2 types of users in this system.

- Admin users – The admin users can create / maintain the user information.
- Normal users – The normal user is similar to admin user, but can't create / maintain the user information. But they can't change their own password if needed.
- First a login screen should be displayed to get the user details. The user has to enter the user-id and the password. The password has to be masked with character '*'.
- The user details should be verified against the details in the user tables and if he / she is valid user, they should be entered into the system. Once entered, based on the user type access to the different modules to be enabled / disabled.
- If user enters wrong password for continuously 5 times, the account should be locked. The information about the login and logout details should be stored in a separate table.

User Details

This module contains 2 parts. User maintenance and password change.

- User maintenance is used to create a new user, modify the details of existing user or remove the existing user. Only the admin type user will have access to this part. Normal user won't have access to this. While storing into database the password should be encrypted and stored. The user can choose his / her own logic to encrypt the password.
- User password is used to change their own password. Both admin and normal user will have access to this.

Database Processing

The data received from source bank is processed as follows.

- The user's account number, account name, IFSC code of the corresponding bank and current account balance.
- The records are validated for various conditions. For e.g., validation on user's information, transaction information, etc.
- If the record is invalid it has be put into separate table.
- Valid records are kept separately for further processing – DD printing.

Transaction Maintenance

- This module is used to maintain the transaction related information like sender's bank account number, receiver's bank account number and amount of the transaction...Only Admin users have the access to this module.

- To add the transaction information the following information are captured. For e.g., transaction serial No, account details of the sender and the receiver.

Transaction Reports

This module is used to generate last five (5) transaction's reports.

- User details report – This contains various information about the users available.
- Transaction details report – This gives the details of transaction related details
- Valid User report – This gives the details about the remaining account balance after the transaction of the sender and the receiver.
- Invalid User report – This gives the details about the report on various invalid users.

3.5. External Interface Requirements

3.5.1. User Interfaces:

Our system will make use of the Microsoft Visual Studio.

3.5.2. Hardware Interfaces:

PC with 160 GB hard-disk and 2 GB RAM.

3.5.3. Software Interfaces:

1. Windows 7/8/10.
2. Microsoft Access Database
3. Microsoft Visual Studio
4. Microsoft Dot Net Framework (version 3.5 and above)

3.6. System Features

This section provides detailed requirements for the application design, including functional requirements. The requirements are following:

- The system should have a **Log In**. A login box should appear when the system is invoked.
- The system should have **Help** screens. Help about the various features of the system should be provided in sufficient detail in a Q&A format. The vacancy creation, application creation, attaching applicant to a vacancy should also be part of the help.
- The **Vacancy** should have the following details:
  - User should give a valid bank account number. This should be linked with a verified bank account.
  - Owned by – This should default to the admin creating the user account. This user will become the member of the application. User can deactivate his/her account whenever he/she wants.
- **Transaction process** data should have the following:
  - This should display a verified sender account.
  - This should display a verified receiver account.
  - Display the transaction details with both account numbers and the amount transferred.
- As soon as all the criteria are fulfilled, then the transaction will proceed else, will fail.
- After the transaction, a transaction report will be generated.

3.7. Non-Functional Requirements

3.7.1. Performance Requirements:

1. The system will update all user record every 5 minutes.
2. The system shall generate a transaction report immediately after it receives a successful transaction

3.7.2. Security Requirements:

- All exchange from one user to the other should have the highest available level of secure connection.
- Account number shall be verified before grant the access.

3.7.3. Software Quality Attributes:

- **Usability:** The system is user friendly and self -explanatory.
- **Adequacy:** The system requires the user details only when the user wants to apply for the transaction.

DESIGN & METHODOLOGY

4. Data Flow Diagrams (DFD)

A Data Flow Diagram is one of the functional Models which is used to represent the flow of information in any computer-based system. The data flow diagram depicts the information flow and Transforms that are applied on data.

Assumptions

1. Bank database is initially populated
2. There are two different types of users – admin and normal.
3. When account is initially created, the bank details are fetched and stored locally.
4. A single bank account cannot be linked to more than one user.
5. Different user accounts cannot share the same phone number.
6. Different user accounts cannot share the same username.

4.1. Context Level DFD

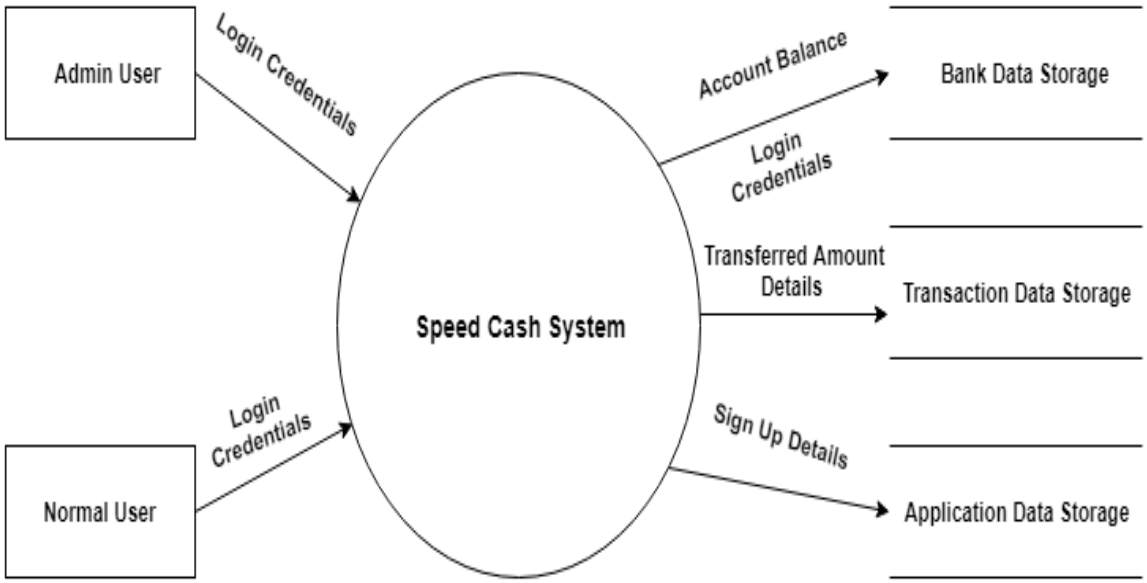
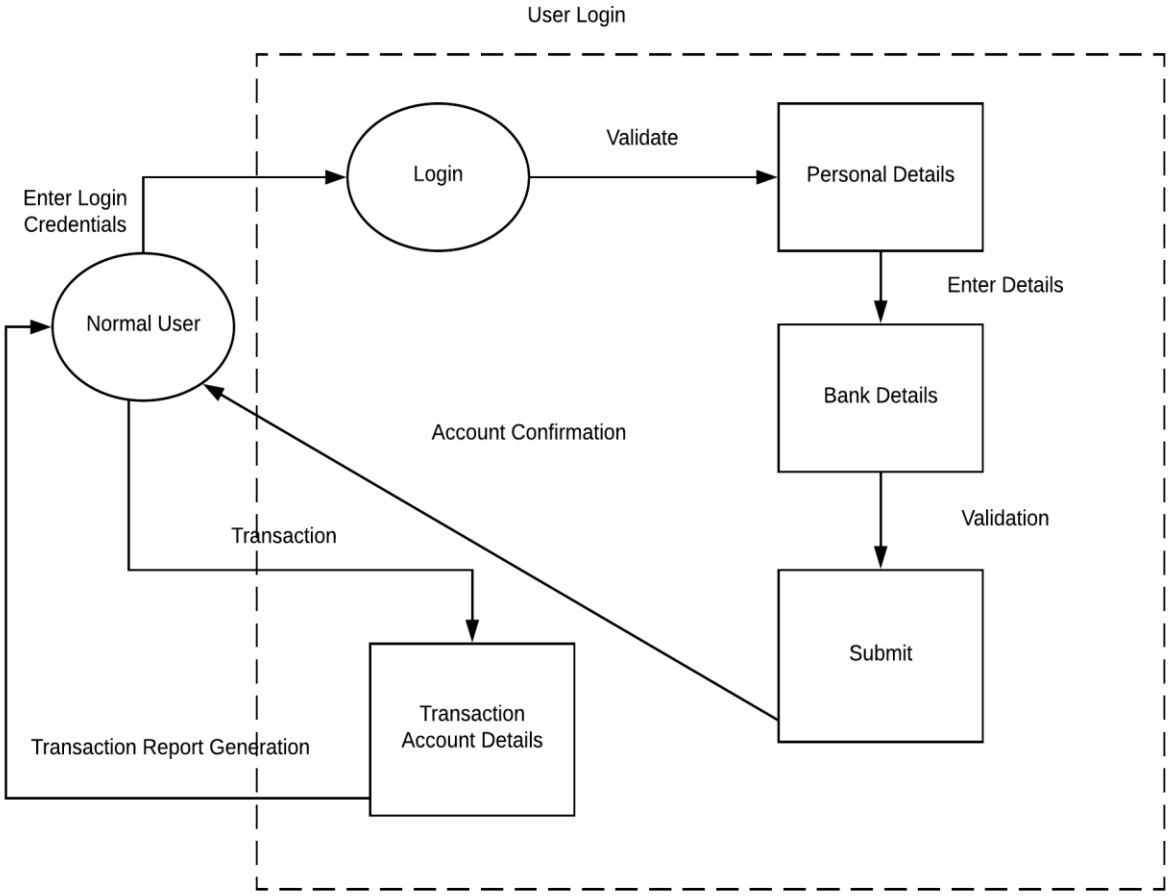


Figure 1: Context Level Diagram

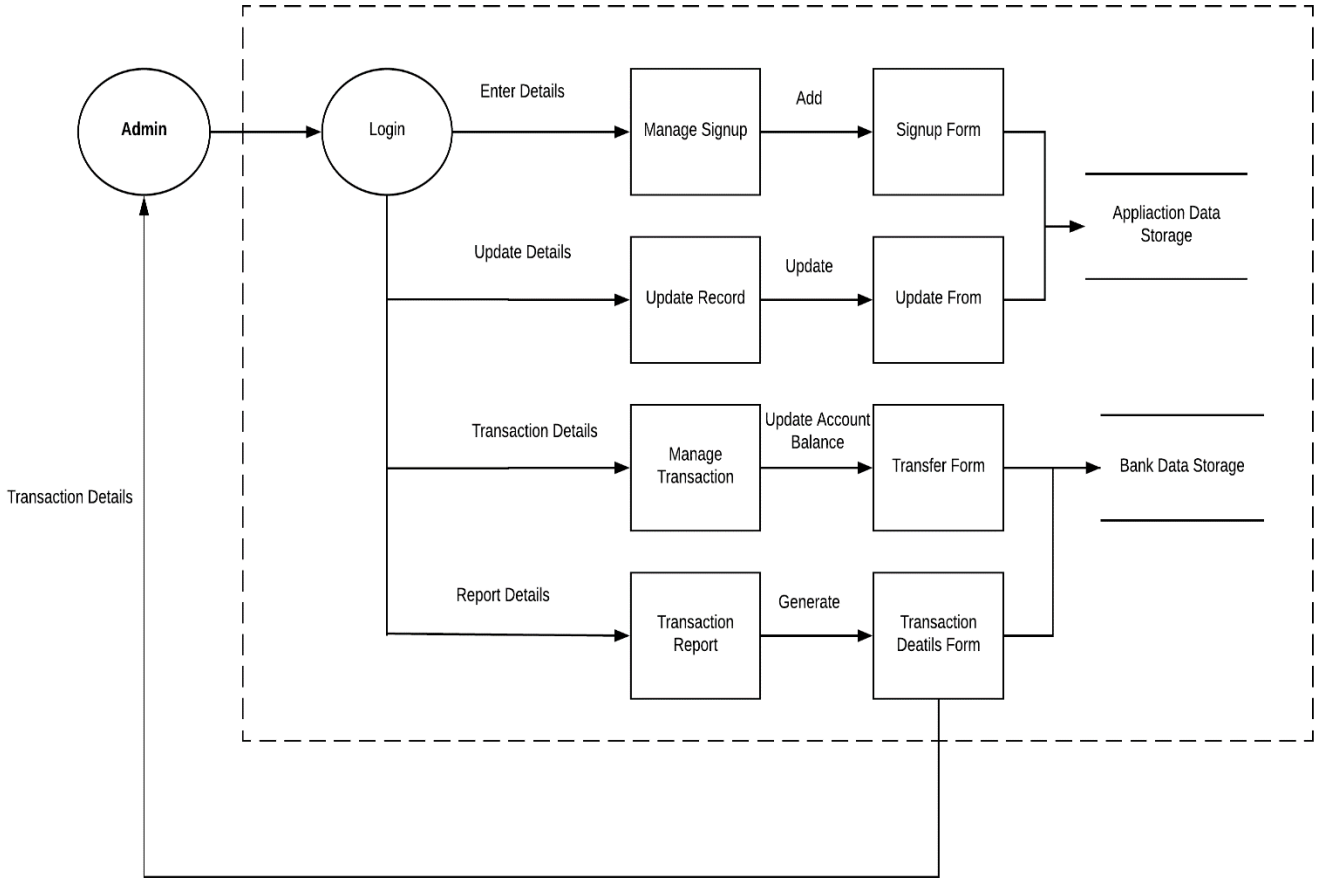
Context Level DFD shows the data system as a whole and emphasize the way it interacts with external entities. This Level 0 DFD of Speed Cash System shows how this application might function in transaction process.

4.2. Level 1 DFD

1.1 Normal User

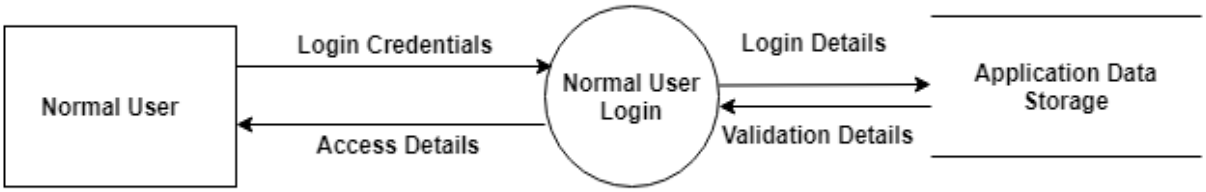


1.2 Admin User

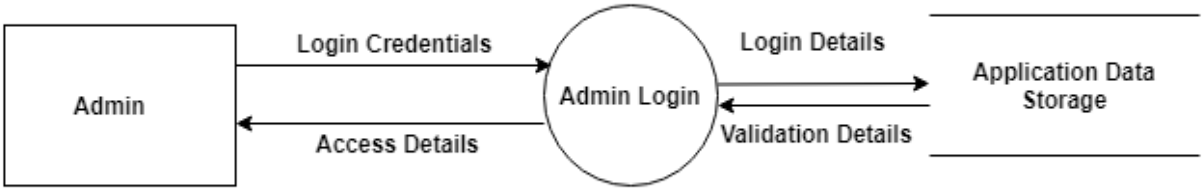


4.2. Level 2 DFD

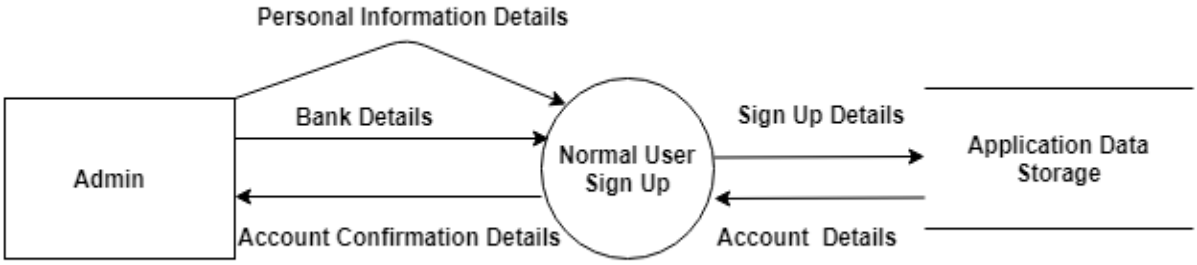
2.1 Normal User Login



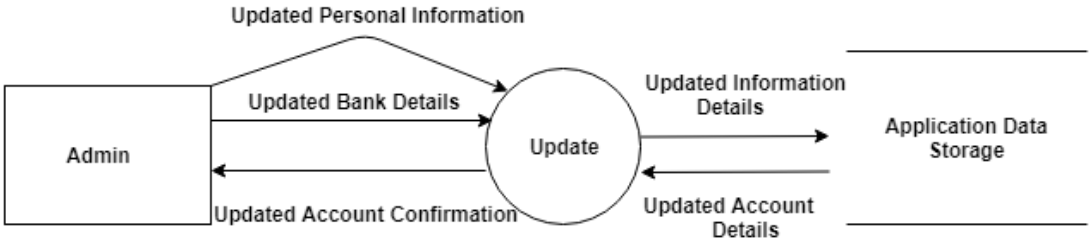
2.2 Admin Login



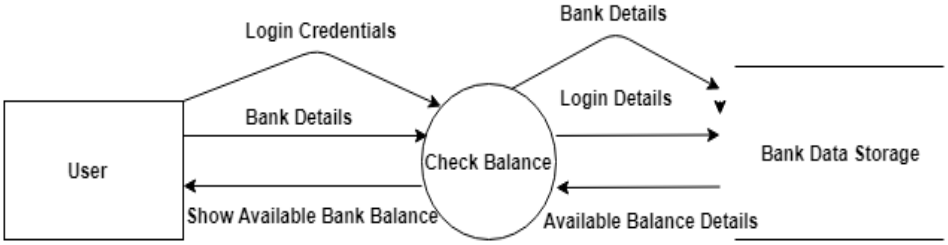
2.3 Creation of new user



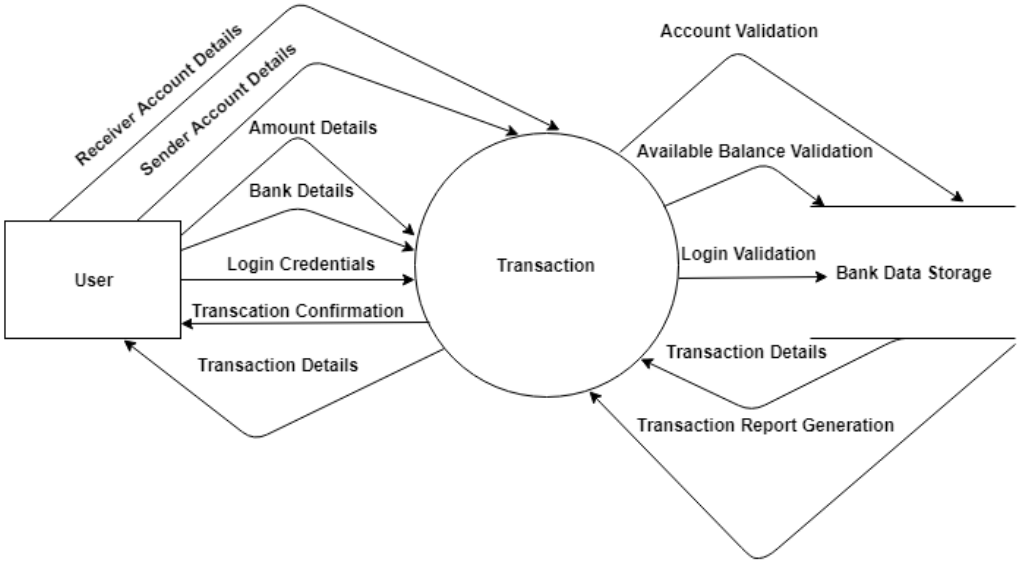
2.4 Update of User Details



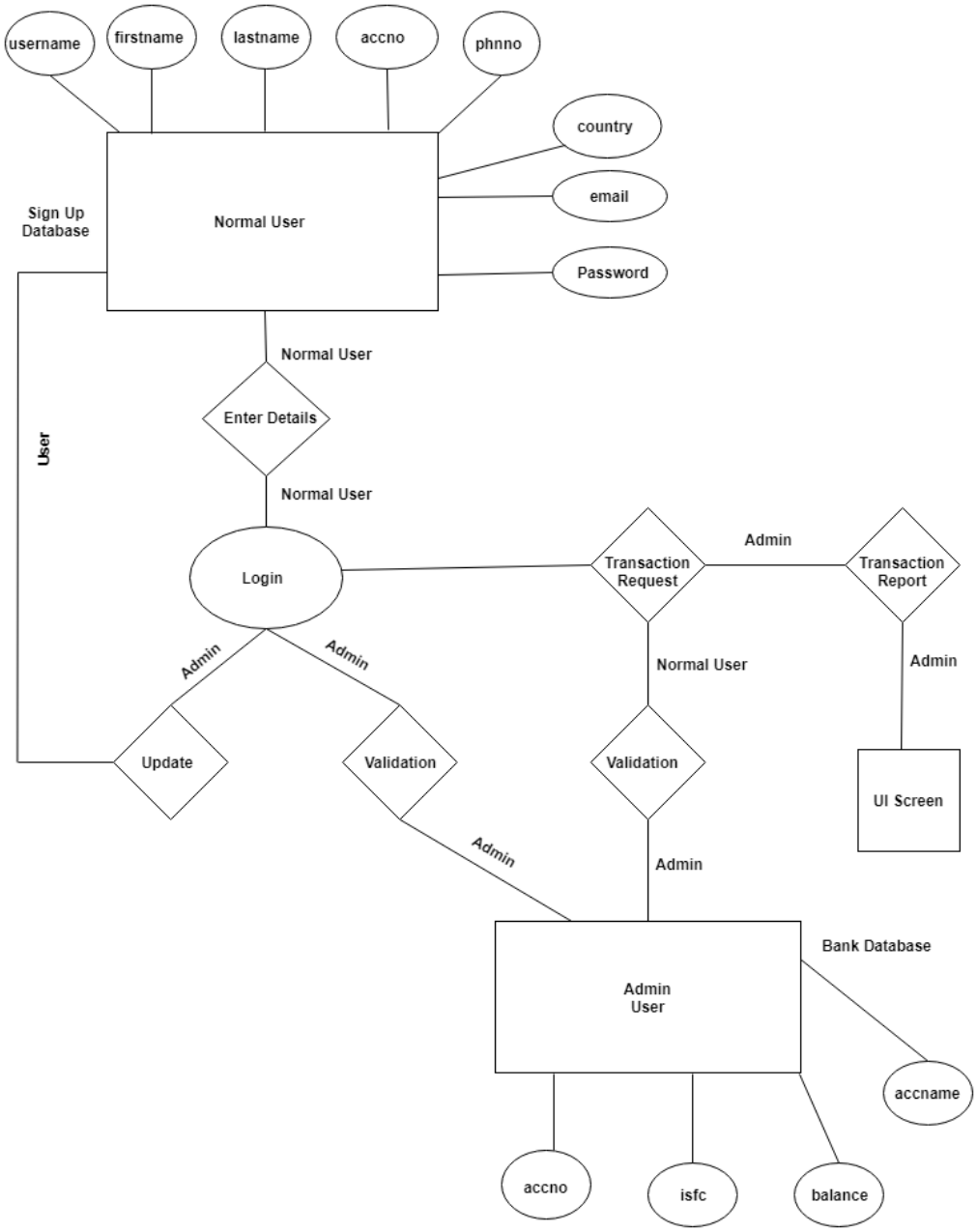
2.5 Balance Checking



2.6 Transaction between two users



5. Entity Relationship Diagram (ERD)

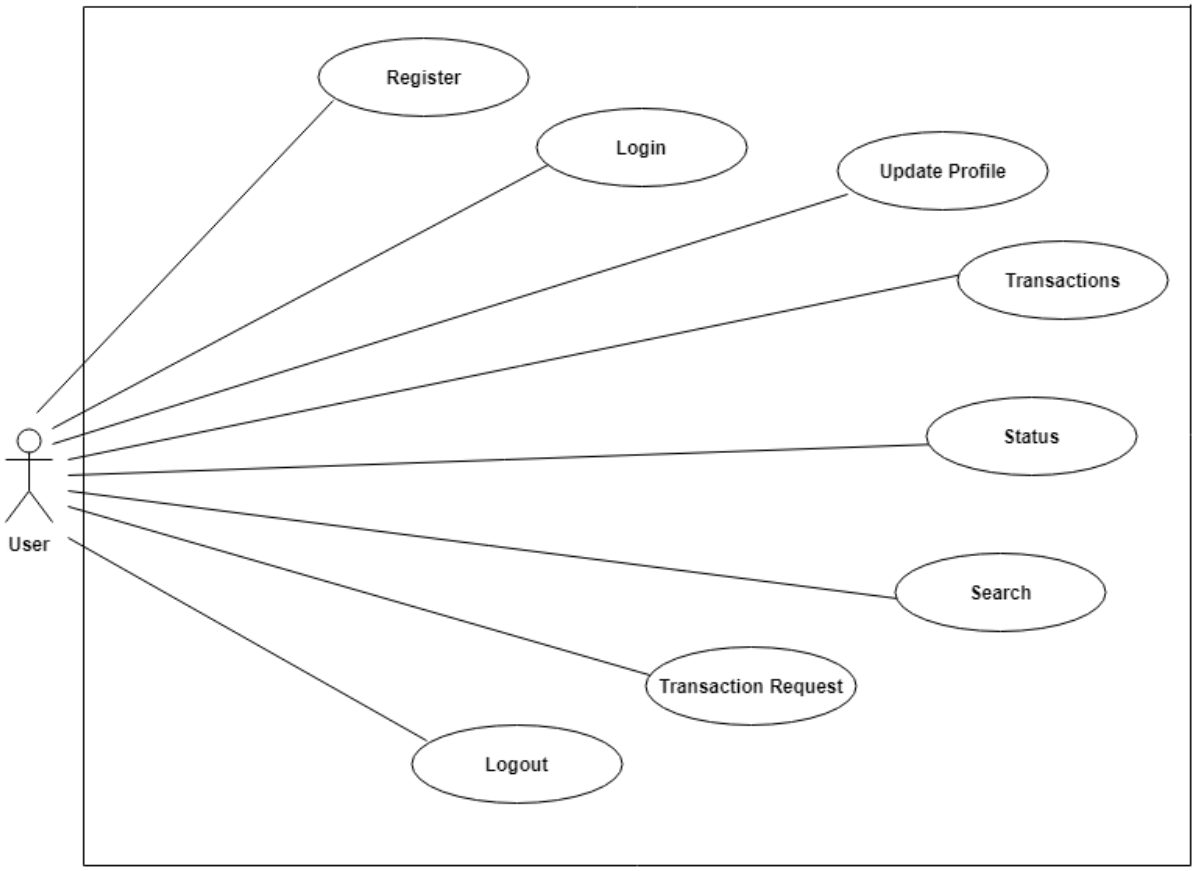


6. Normalization

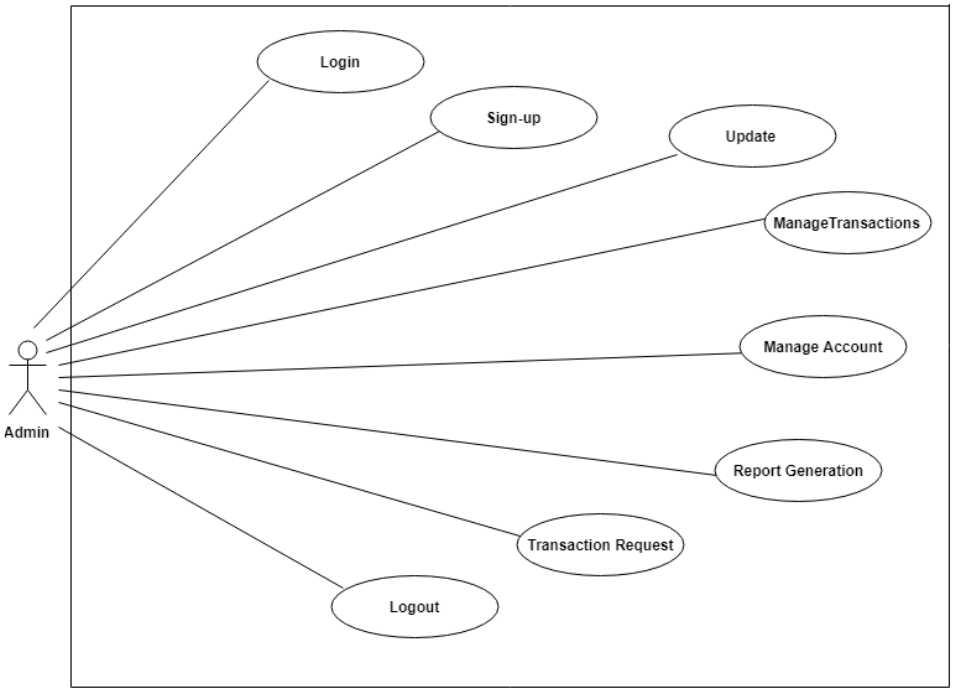
- There is no multi-valued dependency in the table structures. Hence the database is in 1NF.
- There is no partial dependency between attributes and tables. Hence the database is in 2NF
- There is no transitive dependency between attributes. Hence the database is in 3NF.

7. Use-Case Diagram

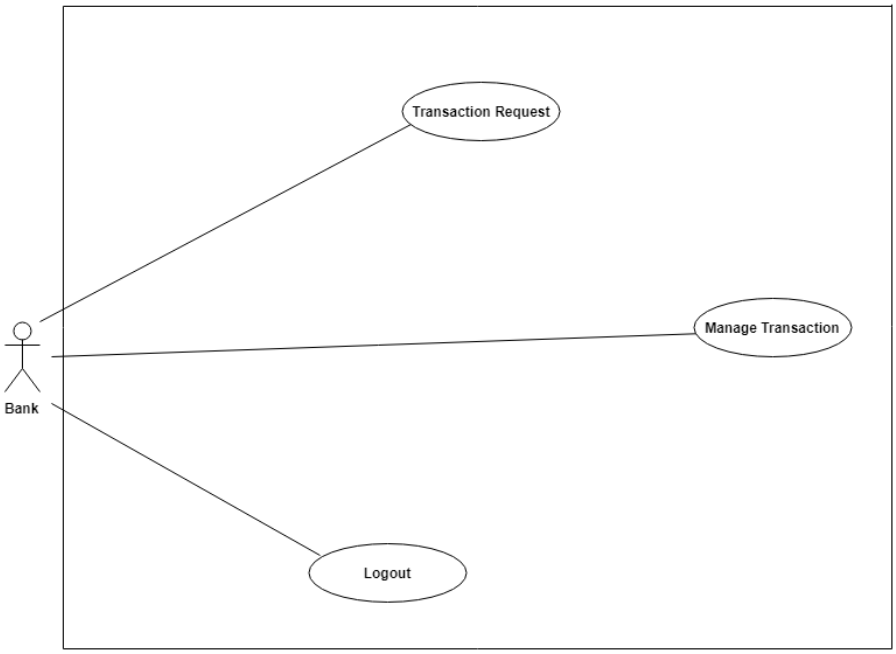
7.1 Normal User as an actor



7.2 Admin User as an actor



7.3 Bank Administrator as an actor



CODING

8. Working Scenario

- It is a tool that can help the users of the system to perform their respective tasks without interfe ring with each other.
- This also helps in anticipating resources that might be needed by the users of the system
- The Applicants can check for vacancies and apply for the same.
- The HR can create, modify or delete vacancies and check the applications.

9. Environment

- Microsoft Access for Database
- Microsoft Visual Studio 2017

10. Sample Tables with Data

10.1 Schema

10.1.1 Sign up Table

Field Name	Data Type
accno	Number
username	Short Text
firstname	Short Text
lastname	Short Text
phno	Short Text
country	Short Text
email	Short Text
password	Short Text
adminr	Short Text

10.1.2 Bank Details Table

Field Name	Data Type
accno	Number
ifsc	Short Text
balance	Number
accname	Short Text

10.1.3 Transaction Details Table

Field Name	Data Type
ID	AutoNumber
accfrom	Number
accto	Number
amount	Number
cdate	Short Text
ctime	Short Text

10.2 Data Values

10.2.1 Sign up Table:

accno	username	firstname	lastname	phno	country	email	password	adminr
1000	diptakb	Diptak	Banerjee	9876123432	India	asdf@asdf.com	diptakb	no
1001	debanjanc	Debanjan	Choudhury	9878786543	India	dech@gmail.c	debanjanc	yes
1002	akashr	Akash	Roy	8987656765	India	asdfa@adsf.co	akashr	no
1003	suchitag	suchit	agarwal	9876565434	India	suchit@gmail.c	suchitag	no
1008	chandu	chandrayan	sengupta	9898765434	India	asdf@asd.com	chandu	no

10.2.2 Bank Details Table:

accno	ifsc	balance	accname
1000	9090	29500	Diptak
1001	9080	28011	Debanjan
1002	9070	42289	Akash
1003	9989	49000	Suchit
1008	8989	891200	Chandrayan

10.2.3 Transaction Table:

ID	accfrom	accto	amount	cdate	ctime
8	1001	1002	1000	04-05-2018	12:25:45 AM
9	1002	1001	500	04-05-2018	12:55:18 AM
10	1003	1001	1000	04-05-2018	11:50:21 AM
11	1000	1002	500	04-05-2018	2:29:33 PM
12	1001	1008	1200	04-05-2018	2:47:18 PM

11. Code Snippets

11.1 Main Form:

```
46          ////////////////////////////
47          flag = true;
48          menuStrip1.Size = new Size(977, 0);
49          pictureBox1.Location = new Point(455, 23);
50          pictureBox1.Image = Image.FromFile(@"F:\DesignLab\Final Project\images\imageedit_5_6548876356.png");
51
52          OleDbConnection connection = new OleDbConnection();
53          connection.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=F:\DesignLab\Final Project\database new\da
54          string ntext = "select firstname, lastname from signup where username='" + username+"'";
55          connection.Open();
56          OleDbCommand commandname = new OleDbCommand(ntext, connection);
57          OleDbDataReader rd = commandname.ExecuteReader();
58
59          string fname = "";
60          string lname = "";
61
62          while (rd.Read())
63          {
64
65              fname = rd.GetString(0);
66              lname = rd.GetString(1);
67
68          }
69
70          welcomelabel.Text = "Welcome " + fname + " "+lname+"!";
71
72          connection.Close();
73      }
74
75
76
```

11.2 Sign Up Form:

```
52
53
54
55
56
57          private void submitButton_Click(object sender, EventArgs e)
58          {
59
60              if ((string.IsNullOrWhiteSpace(firstnameText.Text)) || (string.IsNullOrWhiteSpace(lastnameText.Text)) || (string.IsNullO
61                  || (countryDropdown.SelectedIndex==-1) || (string.IsNullOrWhiteSpace(emailText.Text)) || (string.IsNullOrWhiteSpace(
62              {
63                  // Show message?
64                  MessageBox.Show("Please fill all the fields");
65
66                  return; // Don't process
67              }
68
69              else if (!(isValidPhno(phnoText.Text)))
70              {
71                  MessageBox.Show("Please check phone number");
72                  return;
73              }
74
75
76              else if(!(IsValidEmail(emailText.Text)))
77              {
78                  MessageBox.Show("Enter a valid email");
79                  return;
80              }
81
82              else if (!(isValidUsername(usernameText.Text.Trim())))
```

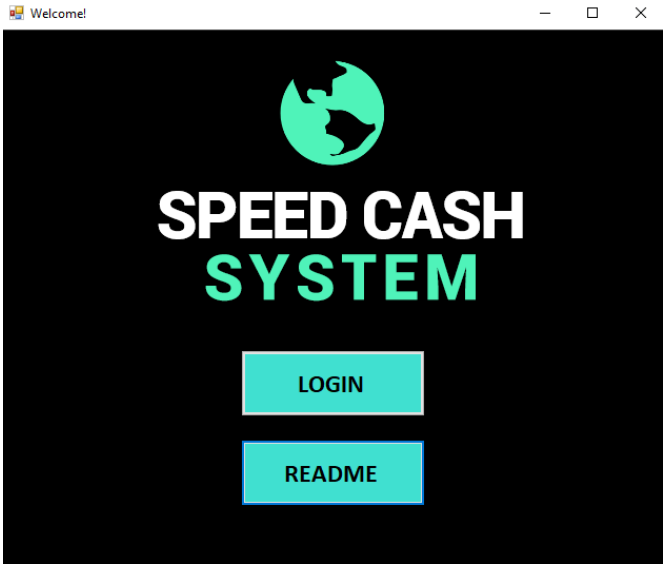
11.3 Transaction Details Form:

```
82
83          string text = "select top 5 * from transactiondetails where accfrom = " + tdaccno + " or accto=" + tdaccno + " order by
84          OleDbCommand command = new OleDbCommand(text, connection);
85
86          OleDbDataReader reader = command.ExecuteReader();
87
88          int i = 0;
89          while (reader.Read())
90          {
91              int accfrom = Convert.ToInt32(reader.GetValue(1));
92              int accto = Convert.ToInt32(reader.GetValue(2));
93              int amt = Convert.ToInt32(reader.GetValue(3));
94              string cdate = reader.GetString(4);
95              string ctime = reader.GetString(5);
96              string accname = "";
97
98
99
100             if (accfrom==Convert.ToInt32(tdaccno))
101             {
102                 label1[i].Text = accto.ToString();
103                 pb[i].Image = Image.FromFile(@"F:\DesignLab\Final Project\images\red arrow.png");
104
105                 string ntext = "select accname from bank where accno=" + accto.ToString();
106                 OleDbCommand commandname = new OleDbCommand(ntext, connection);
107                 OleDbDataReader rd = commandname.ExecuteReader();
108                 while (rd.Read())
109                     accname = rd.GetString(0);
110
111             }
112             else
```

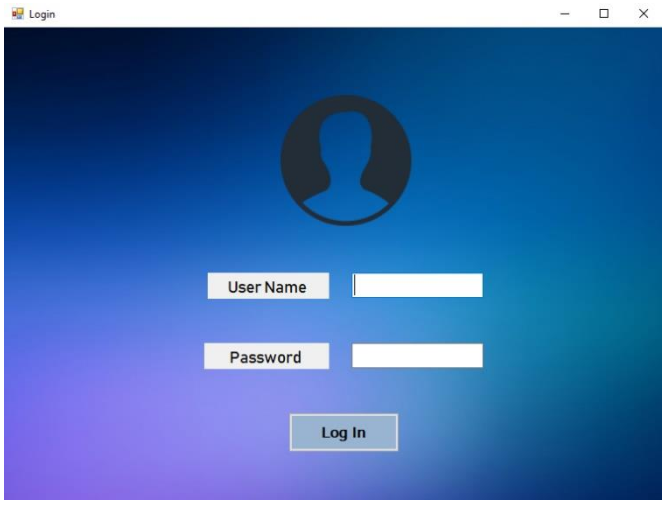
RESULT

12. Software Screens

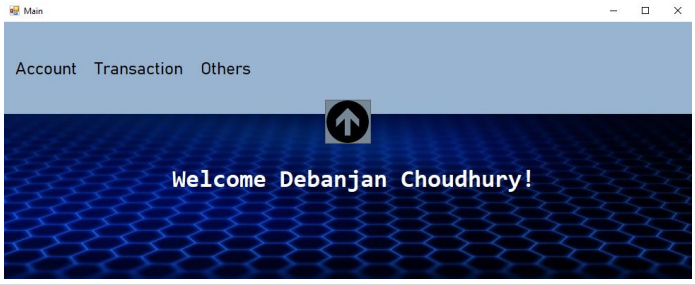
12.1 Startup page:



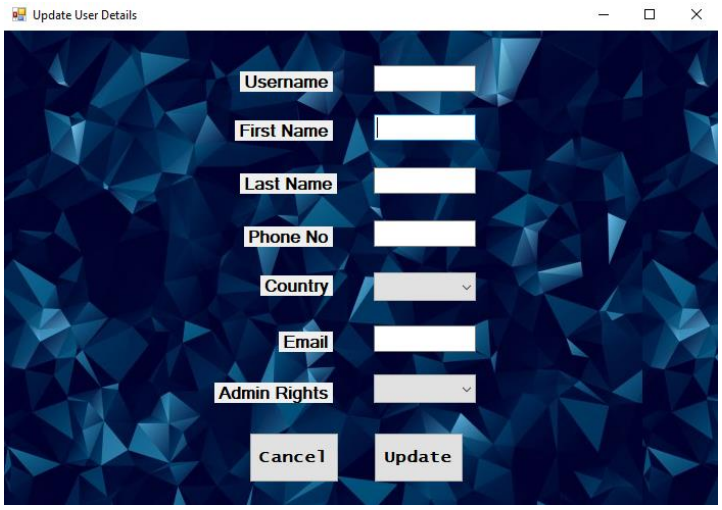
12.2 Login page:



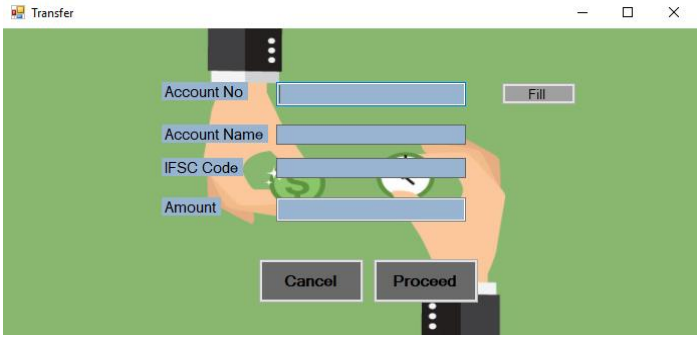
12.3 Dashboard:



12.4 New user creation:



12.5 Transfer page:



13. Conclusion

This Project on Speed Cash System has been made with respect to multipurpose transaction for multiple users. A transaction can occur any time with accordance to the user's choice. It is the duty of the admin to update the user account status and to update user account details. The user on the other hand can check for the available account balance and transaction details. Transaction is an integral part of any individual. Thus, we made an effort to create an online speed cash system to simplify the transaction process.

14. Future Work

The work done here consists of only managing of transactions. Future improvements on the work might include dealing with the user validation process with the UAN. Since this project deals with only one bank account of a single user, future scope might as well include multiple bank accounts for a single user. There is wide scope in the approach.