BillGuards-The Banking Manager

ADITYA JAIN

https://github.com/adityajain10/billguards-The-Banking-Manager

Table of Contents

1.0 Introduction

1.1 Purpose of this document

The purpose of this document is to present a detailed description of the **B**illGuards -The Banking Manager. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system

1.2 Statement of scope

In present times, all major economic transactions have started taking place digitally. The major trends of modern digital transactions is substantiated by use of database management. These databases can be accessed by anyone with specific rights, and perform certain actions on it. The data update is done almost automatically and is much faster. Users can, in present days can access their accounts directly without going to a bank, making transfers, transactions and accessing cash directly without standing in long queues as was prevalent earlier using ATM machines. On employee-side the data is much more organized, and accessing and performing actions on user accounts is easier for them. Due to this the bank has better work efficiency and customer experience improves as well.

1.3 Intended Audience

This document is mainly written for the bank employees, developers, project manager, and testers of BillGuards -The Banking Manager since it focuses on the required functionality, analysis, and design of the system. It is suggested that the SRS structure overview section is read first before proceeding through the sections that are most pertinent to each reader type. Any information needed for marketing staff will be communicated by the development team.

This SRS would be used by the following persons:

a) Bank Employees and End-Users: They would be used to tell them about various banking functionalities.

b) Research Students: Research students are advised to read all the sections of this document to get an overall idea of the work-flow and technicalities of the software.

c) Testers: It can be used as a documentation to know the interfaces.

2.0 System Overview

2.1 Product Perspective

BillGuards -The Banking Manager can be used by Bank Employees and/or Customers depending on bank policy. It can be used by several employees of the bank at the same time with required rights. It can be accessed using the BillGuards -The Banking Manager Application with graphical interface. Our Product consists mainly of two parts i.e. the Employee Work-Space (EWS) and the ATM Banking.

The system would deal with the internal banking functions like new account registration, withdrawal, deposit, money transfer etc. The ATM banking would be for direct access of customers, who could use it for Cash-Withdrawal, Transfers and Account-Summary. Both of them connect to a main database server for storing and retrieving the data of the Customers.

2.2 Product Functionality

Our app provides little functionality as of now. As the idea is original, there are many more modules that can be developed and marketed according to the needs of the users. As of now the modules that are developed are:

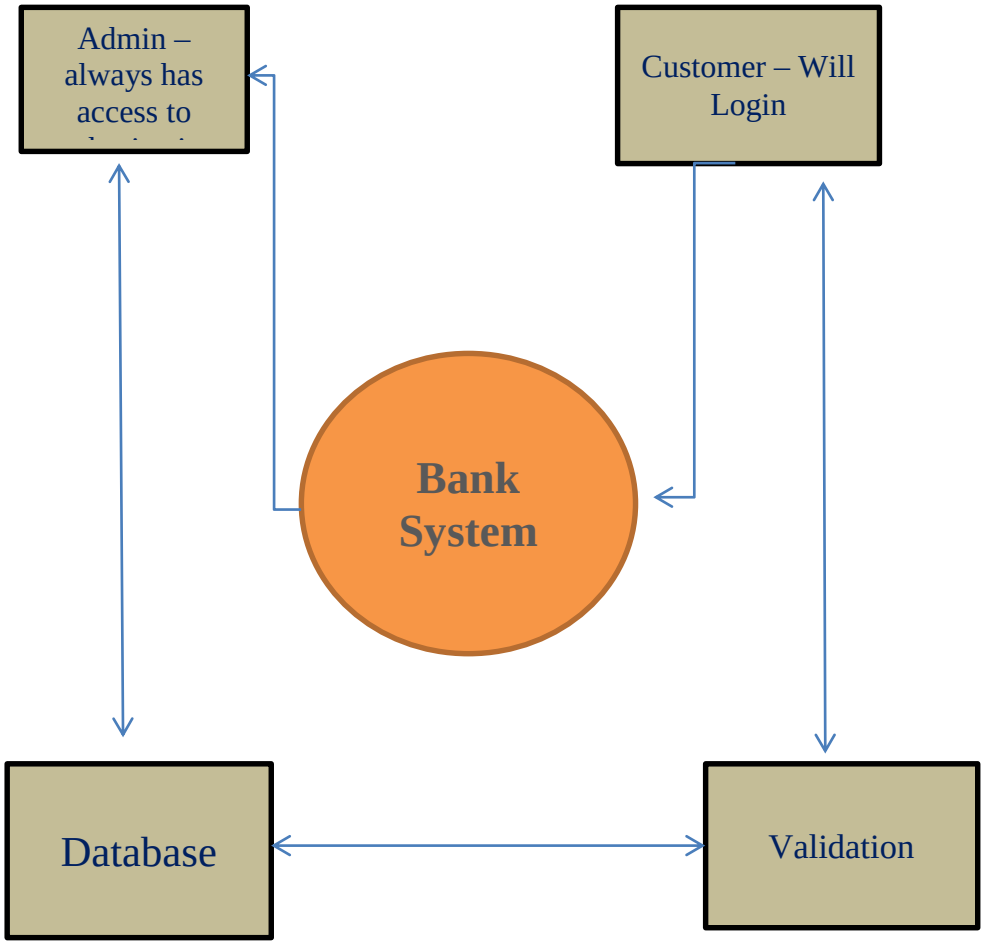
1. Employee login section for existing user
2. Facility to add new branch, department
3. Adding new recruited employer with auto generated employee id
4. Adding new customer
5. Enquiry for account, employee, current balance, transaction
6. Different tables for employee and customer details such as name, address, bank balance.
7. Separate section for transaction details of facility to issue credit card to customers
8. Transactions: credit or debit money from account
9. Deleting existing account and employee
10. New employee ID, customer ID, transaction ID will be generated automatically

3.0 Architectural design

3.1 Program Structure

BillGuards-The Banking Manager will utilize a call-and-return architecture since the program operates through a sequence of hierarchical menus. At the bottom of the hierarchical tree the program will return dynamic responses to the users strategic menu choices.

3.1.1 Architecture diagram



3.2 Communication Interfaces Structure

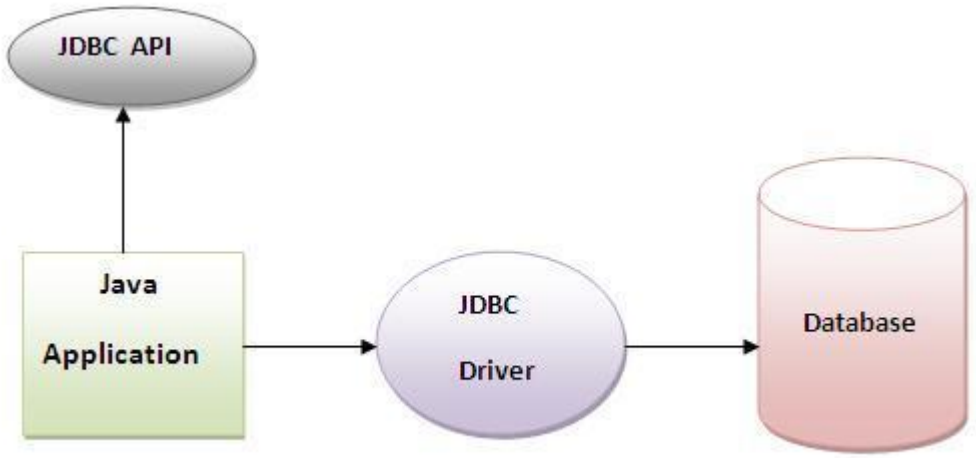
System will use JDBC Bridge for database connectivity with JAVA using NetBeans as front end.
The programming involved to establish a JDBC connection is fairly simple. Here are four steps:

Import JDBC Packages: Add import statements to your Java program to import required classes in your Java code.

Register JDBC Driver: This step causes the JVM to load the desired driver implementation into memory so it can fulfill your JDBC requests.

Database URL Formulation: This is to create a properly formatted address that points to the database to which you wish to connect.

Create Connection Object: Finally, code a call to the DriverManager object's getConnection( ) method to establish actual database connection.

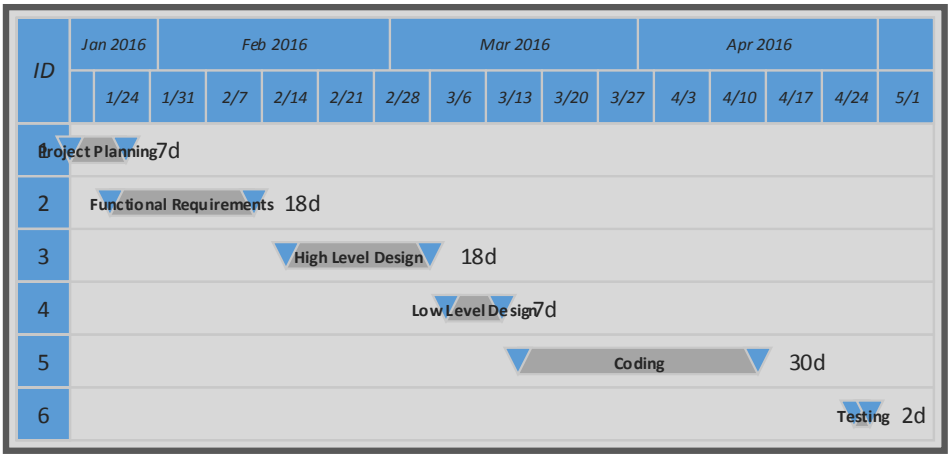


Main Communication Interface

4.0 Schedule

This program development will undergo three major cycles. The first cycle is the Documentation Phase where the Requirements and Software Design documents are built. These documents will provide the group with the layout for each module and diagrams on how everything works together. The second cycle will be the actual coding of each module (and sub-programs). This pertains to members completing their assigned sub-project. This includes their individual sub-menu, module design, and module testing. By the end of this cycle they are expect to have a stand-alone working application ready for system integration. The third and final cycle entails overall system testing and integration. During this phase all mini-games and sub-menus are integrated and tested. If time permits, interface improvements and documentation updates may be performed during this time.

4.1 Scheduling diagram



5.0 Design and Implementation Constraints

a) GUI is only in English.
b) Login and password is used for identification of customer and there is no facility for guest.
c) BillGuards -The Banking Manager software version can be updated via the Internet in case of important security patches, new compatible devices, or for adding more functionality.
d) A user cannot use the subsystem if he has not registered for an account.

6.0 Assumption and Dependencies

a) All users are assumed to be equipped with GUI terminals.
b) User has basic knowledge of computers.
c) User should have sufficient knowledge of English since the system interface will be in English.

d) The system will have simple and easy to use interfaces.

e) Provides accurate data.

7.0 Product Functionalities

The Product Functionalities are as follows:

For Admin/Employee

a) Login Issue Credit Card
b) Balance enquiry
c) Fund transfer
d) View/ Edit EmployeeDetails
e) Add Department
f) View statement
g) View transaction

For **Customer**

a) Login
b) Balance Enquiry
c) View Statement
d) View Transaction

8.0 Interface

8.1 User Interfaces

The first interface is the log-in screen. This is where the user has a specific Username and Password so that they can gain access to the database. If the login is verified with the database, then user will get access to various functions like Transactions, Balance, Credit Cards, and Departments etc. However, if the login is not verified, then user will be asked to try again with correct credentials.

8.2 Hardware Interfaces

For BillGuards -The Banking Manager, an encrypted pen drive will be used which will be provided to the client that will be used in his/her login. At the login time, the pen drive has to be inserted in order to facilitate the login. BillGuards -The Banking Manager requires an entry-level PC for smaller number of bank accounts (like, when data is being stored locally). For larger no. of bank accounts, a server class machine is recommended.

Microsoft Windows XP Professional SP3/Vista SP1/Windows 7 Professional:
Processor: 800MHz Intel Pentium III or equivalent
Memory: 512 MB
Disk space: 750 MB of free disk space

8.3 Software Interfaces

BillGuards -The Banking Manager can run on any recent version of Windows, Linux etc. System will use JDBC Bridge for database connectivity with JAVA using NetBeans as front end. It requires following development tools:

a) JDBC
b) Java SQL connector
c) Java development Tool Kits for working with java server pages.

8.4 Communications Interfaces

System will use JDBC Bridge for database connectivity with JAVA using NetBeans as front end.
The programming involved to establish a JDBC connection is fairly simple. Here are four steps:

Import JDBC Packages: Add import statements to your Java program to import required classes in your Java code.

Register JDBC Driver: This step causes the JVM to load the desired driver implementation into memory so it can fulfill your JDBC requests.

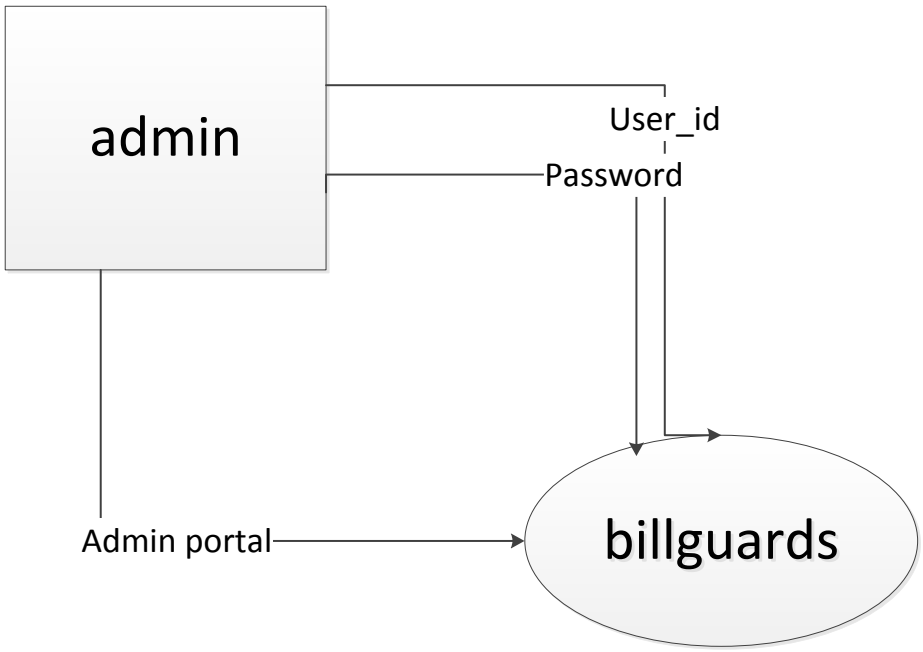
Database URL Formulation: This is to create a properly formatted address that points to the database to which you wish to connect.

Create Connection Object: Finally, code a call to the DriverManager object's getConnection( ) method to establish actual database connection.

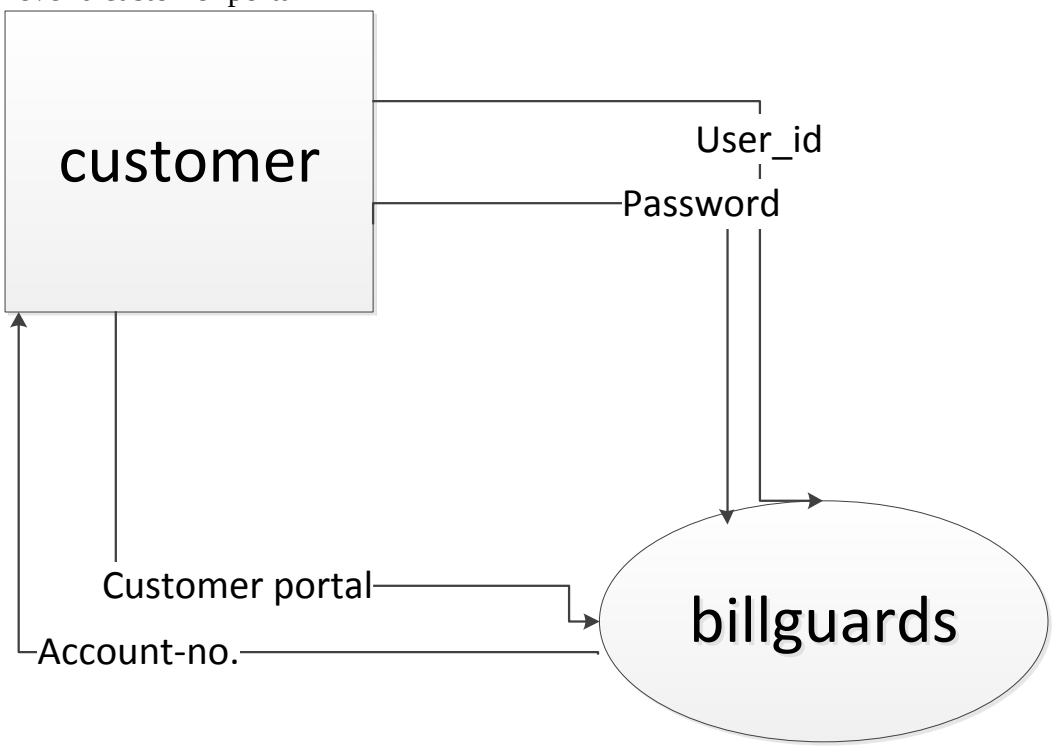
8.0 Detailed system design

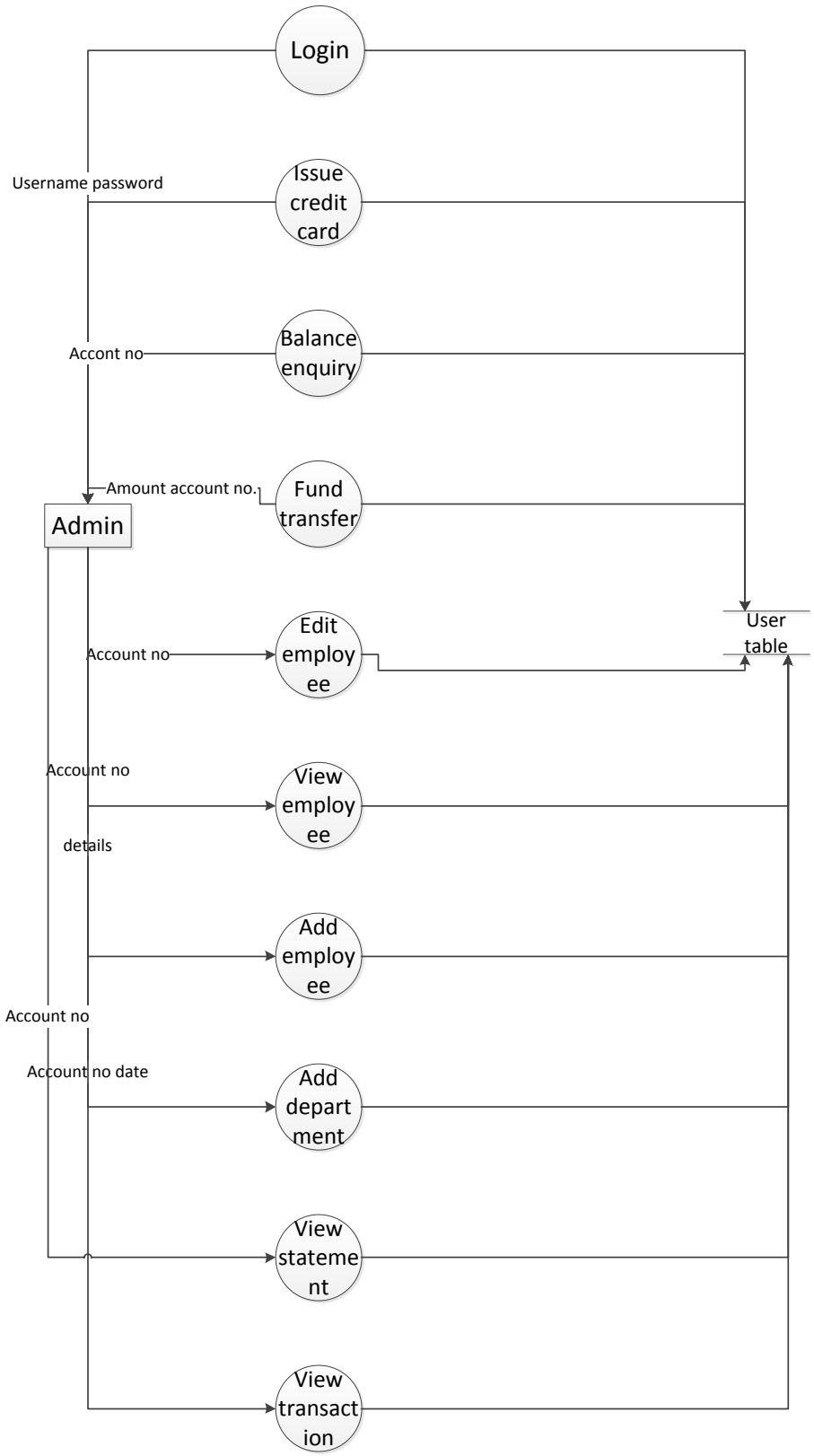
8.1 Data Flow Diagram

Level 0 admin



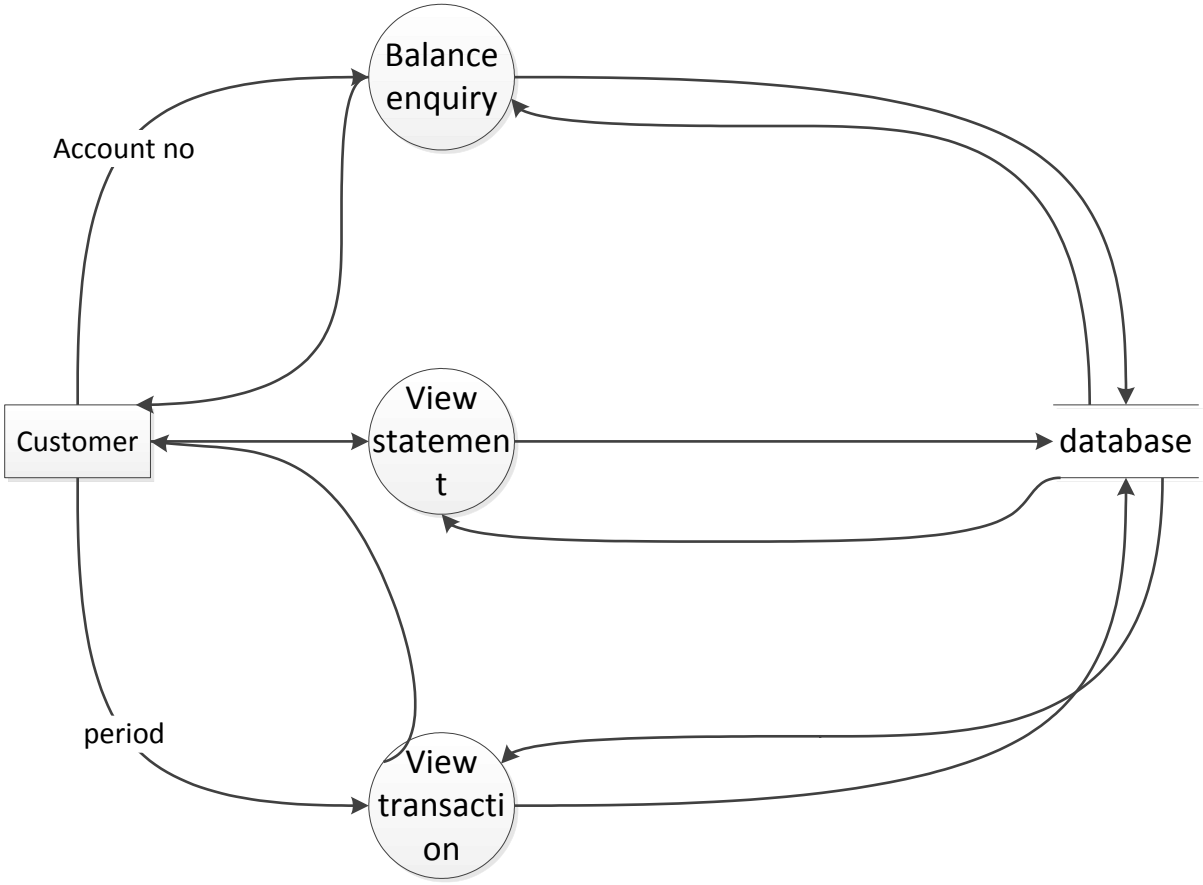
Level 0 customer portal





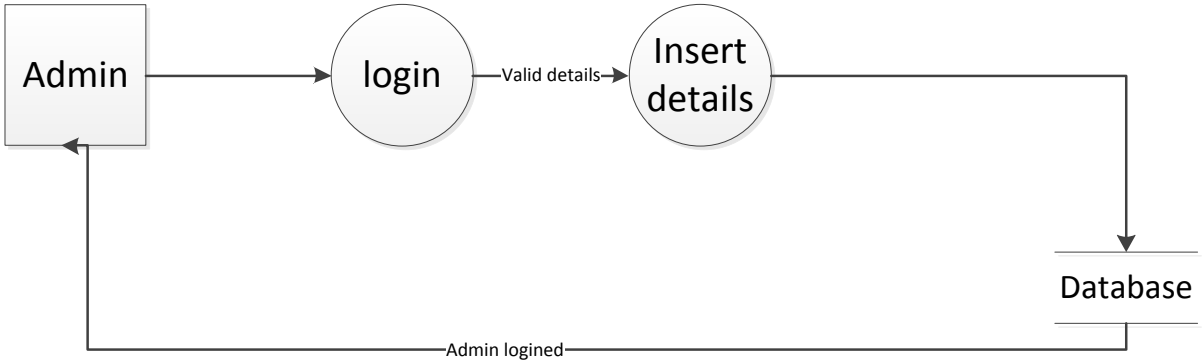
Level 1 admin

Level 1DFD for customer

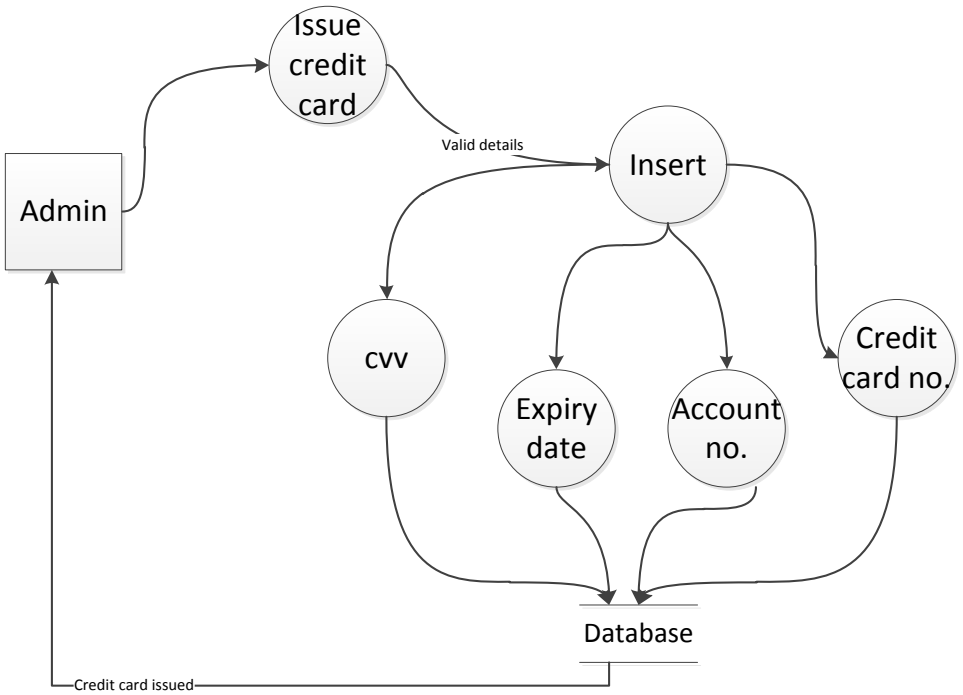


For admin

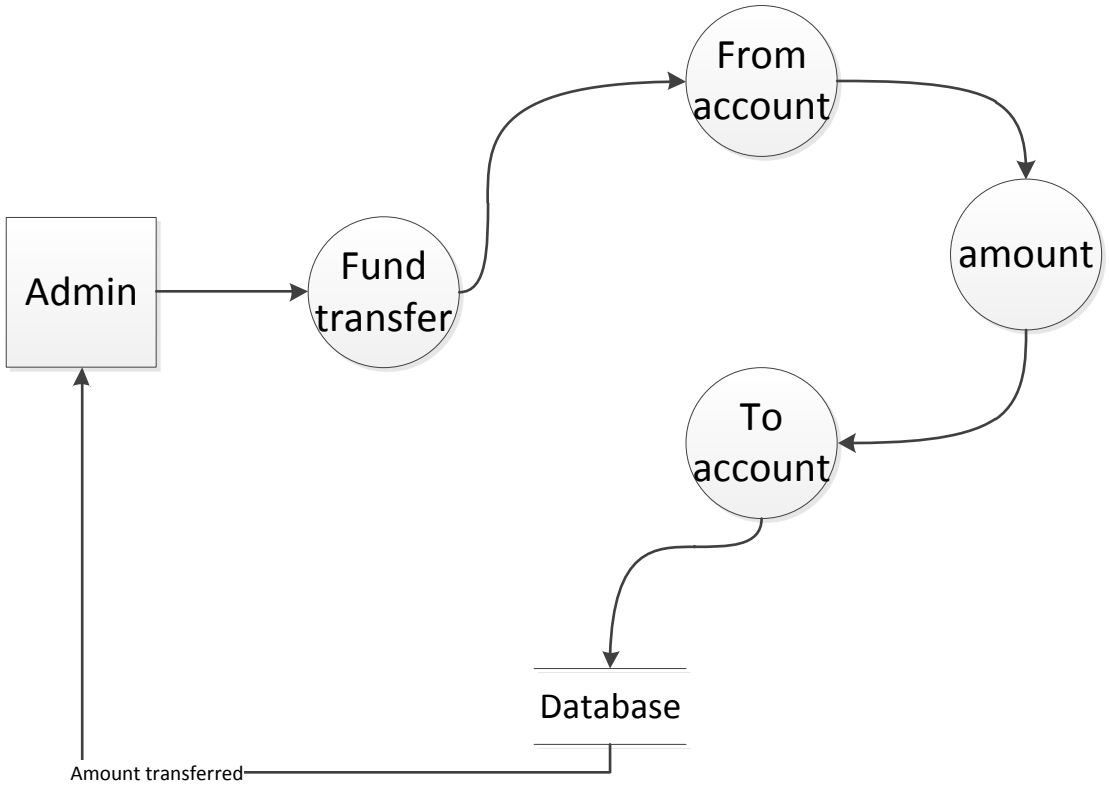
Level 2 dfd for admin login



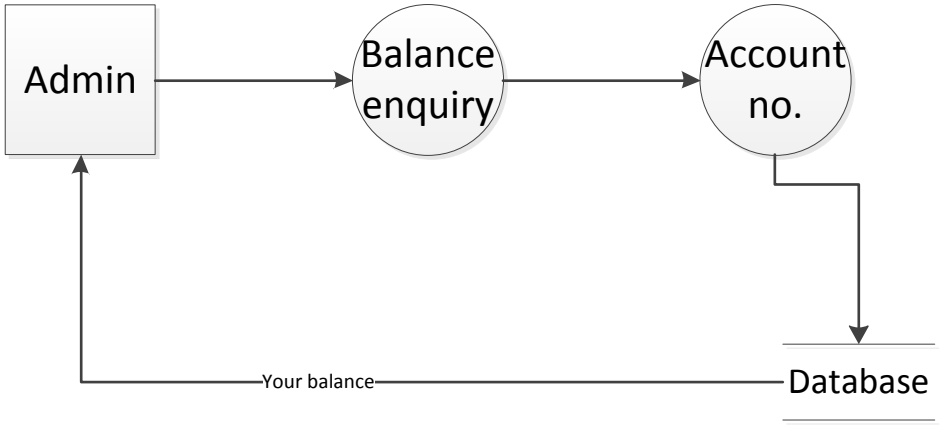
Level 2 dfd for admin issue credit card



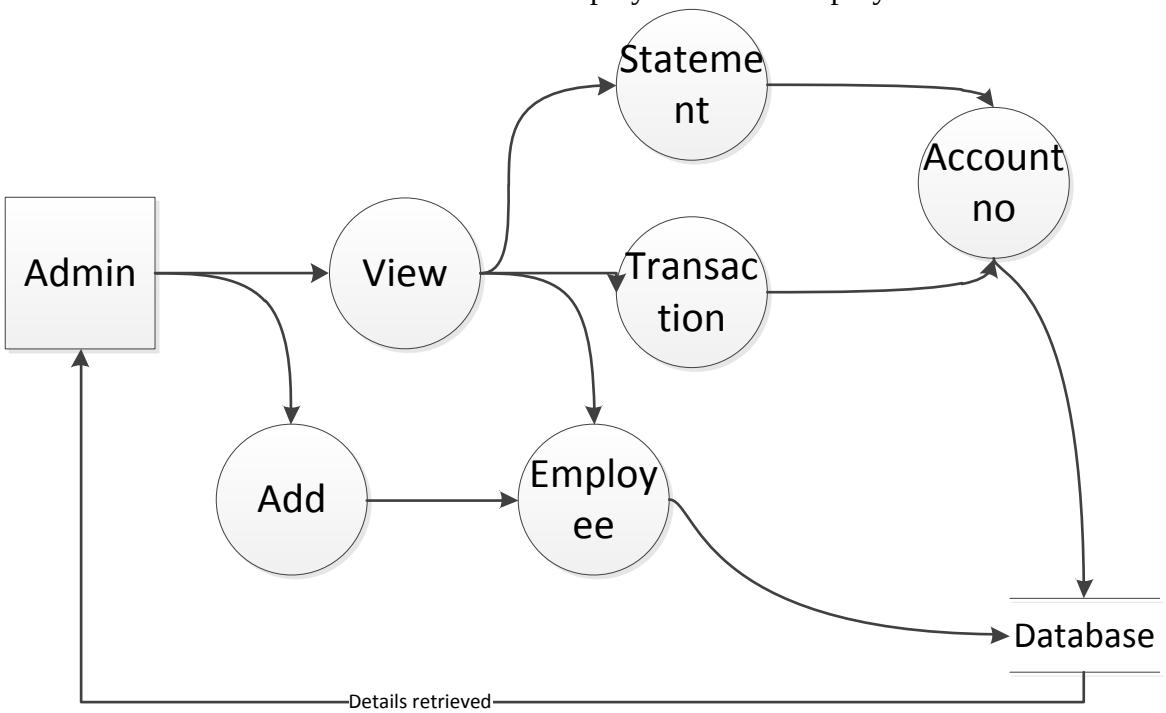
Level 2 dfd for fund transfer



Level 2 dfd for balance enquiry

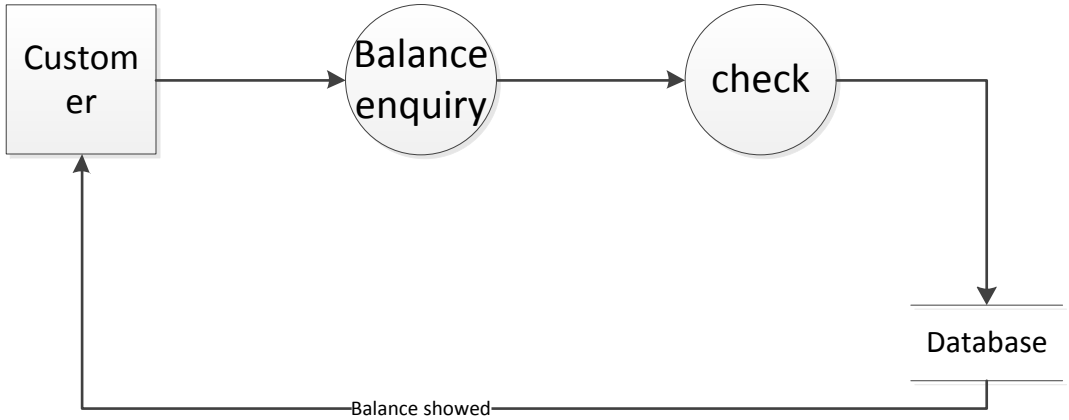


Level 2 dfd for view statement transaction employee and add employee

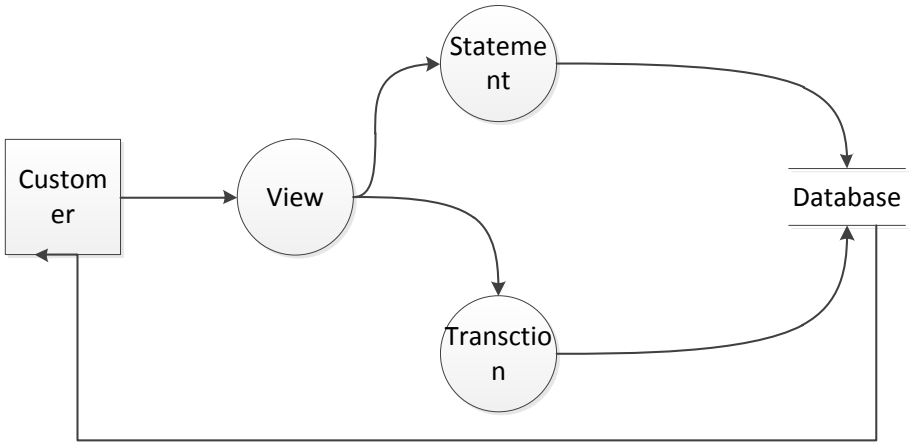


For customer

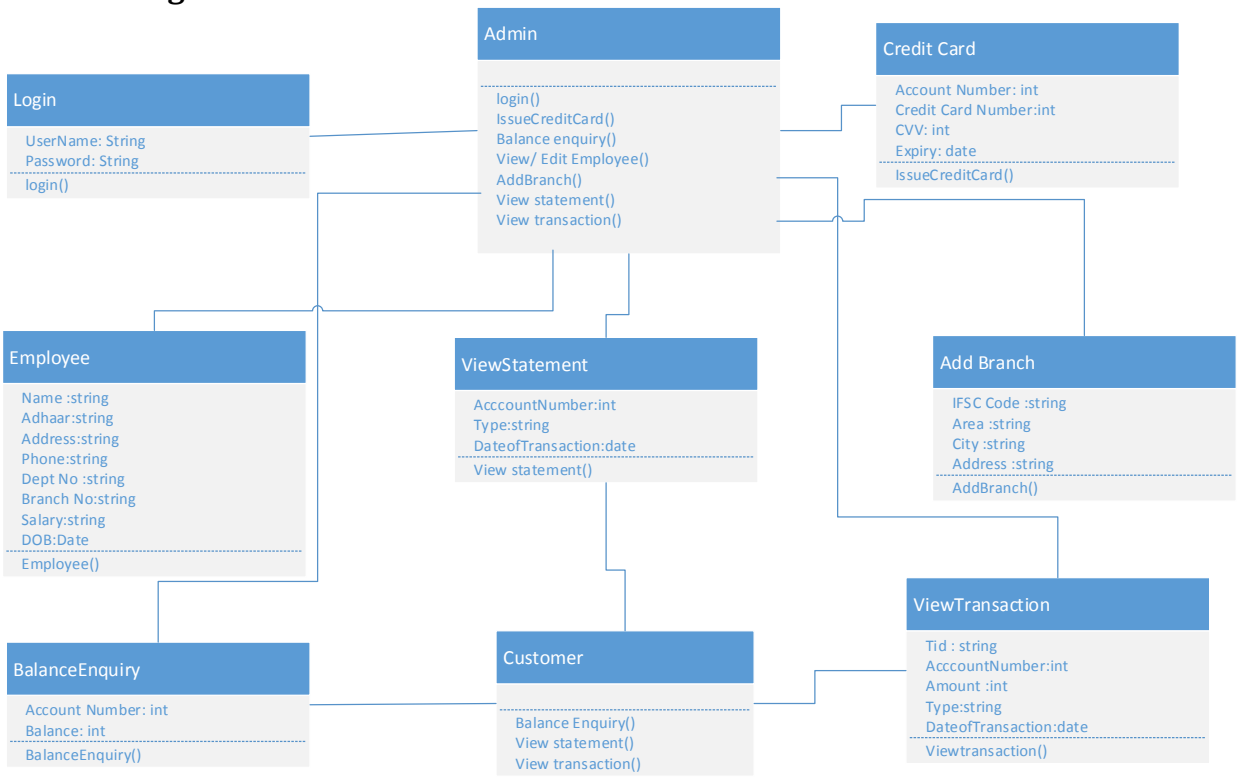
Level 2 dfd for balance enquiry



Level 2 dfd for view statement and transaction

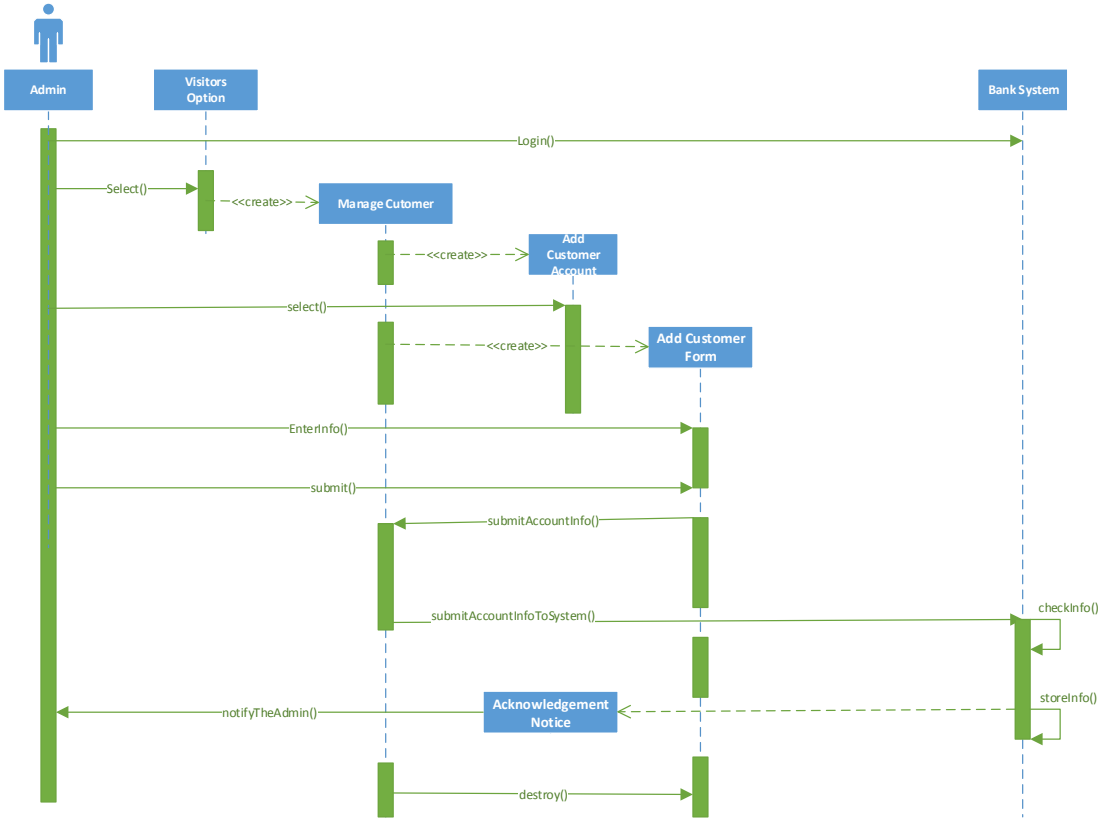


8.2 Class Diagram

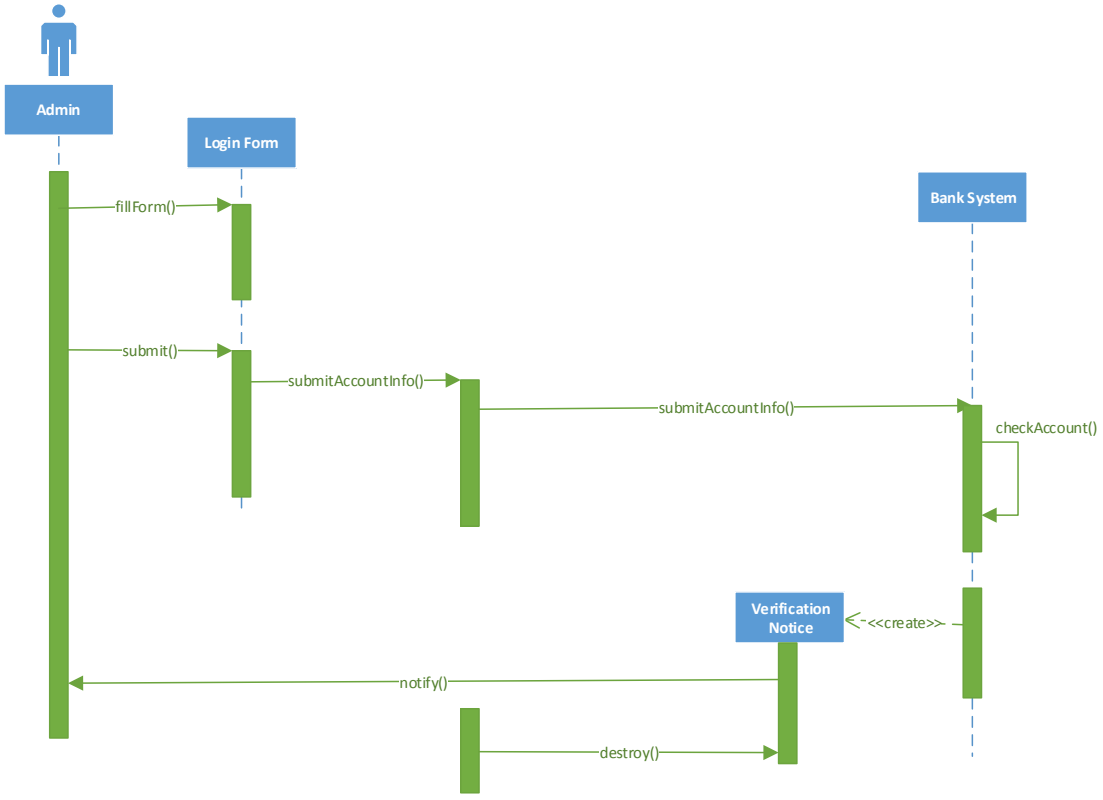


8.3 Scheduling Diagram

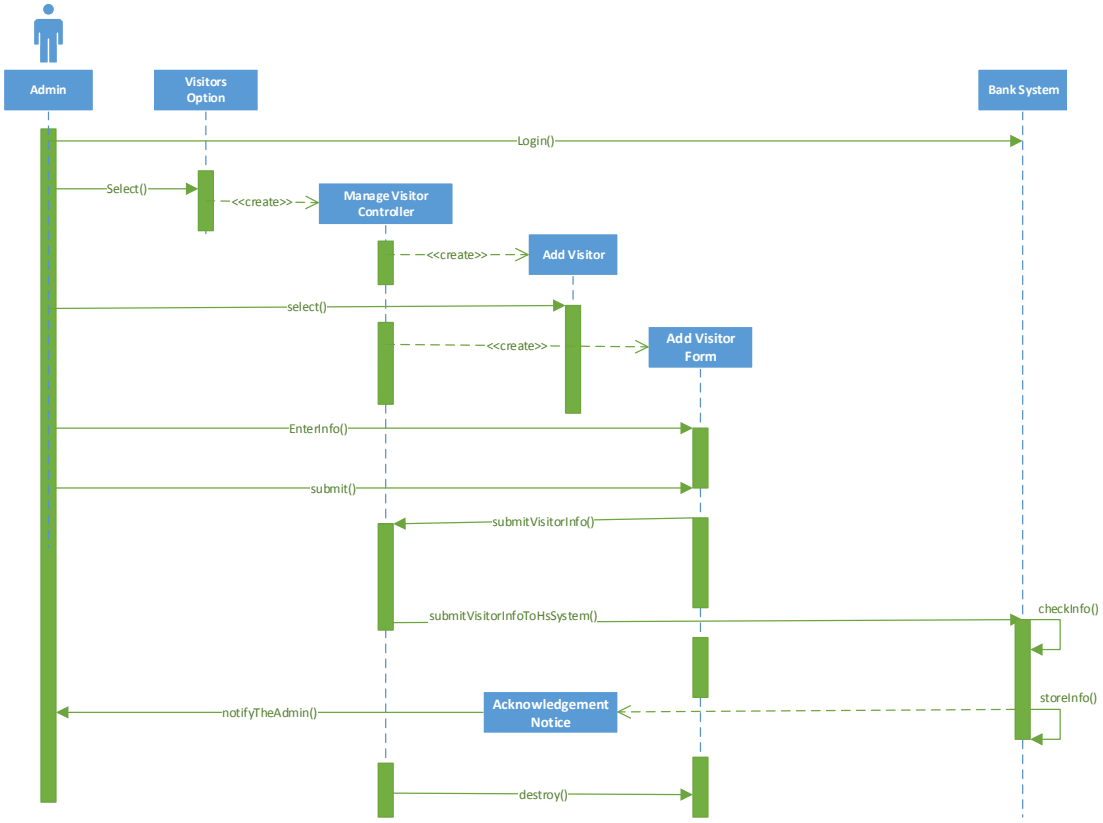
Edit Account



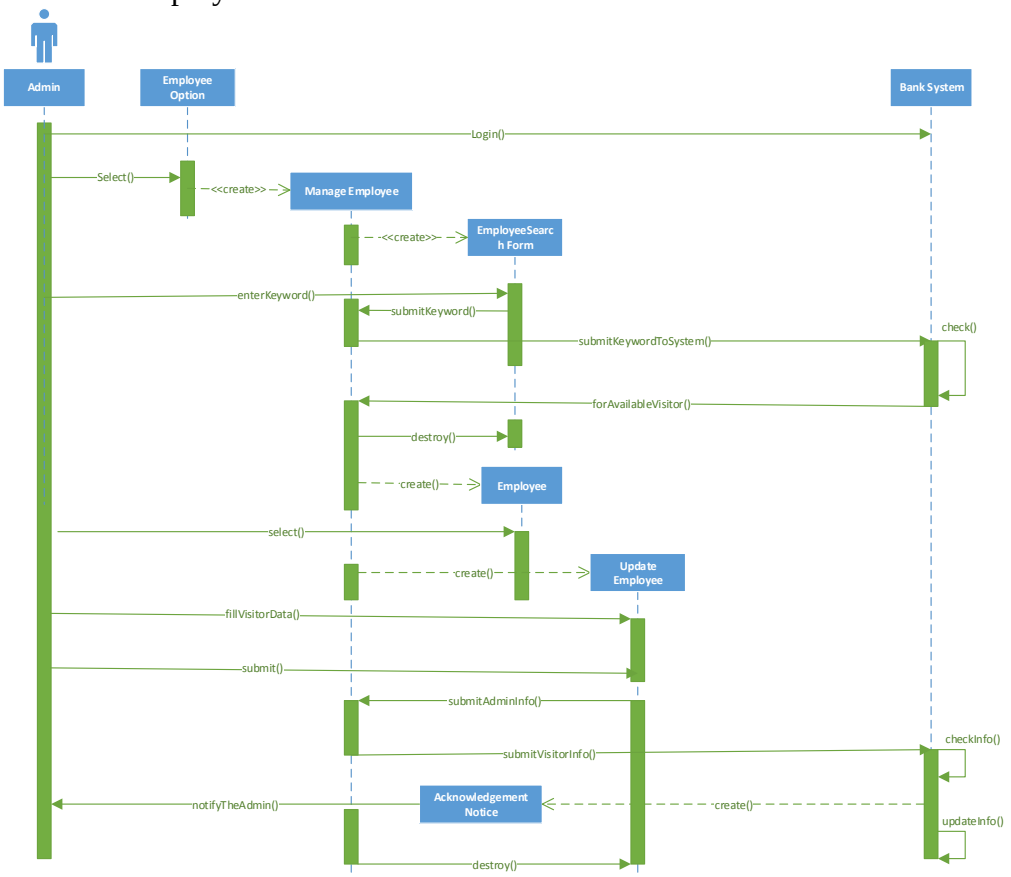
Login



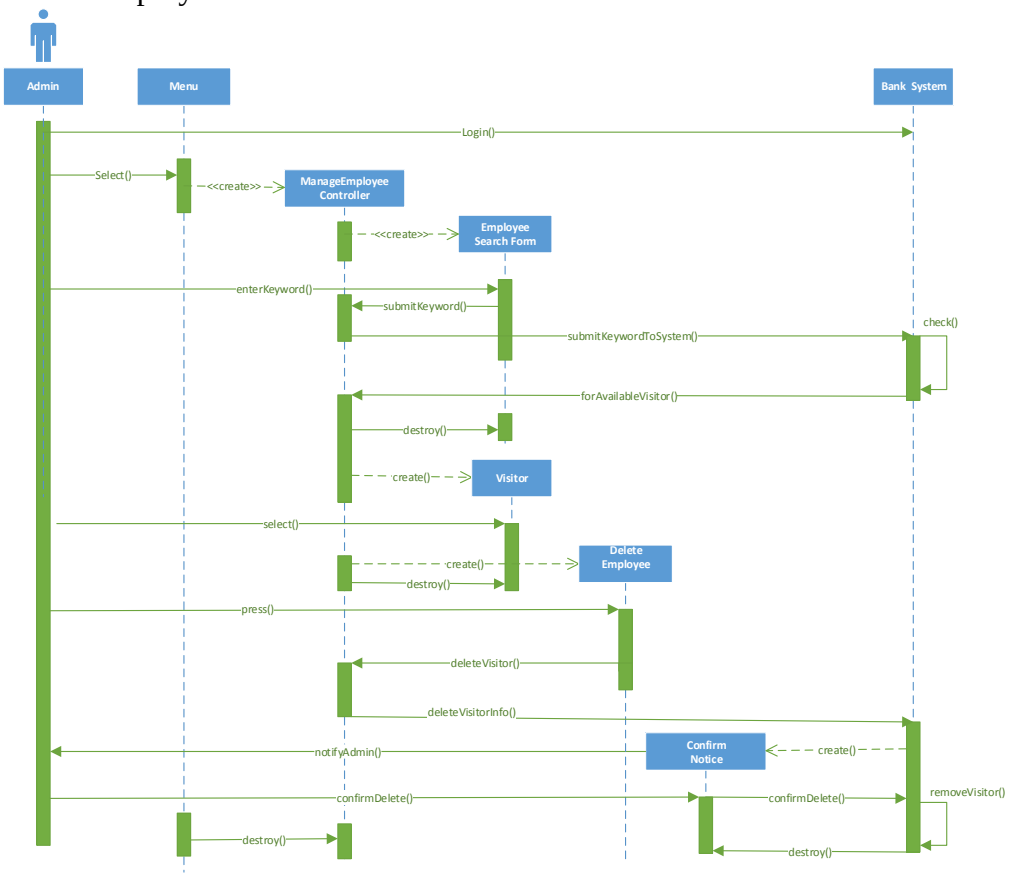
Customer



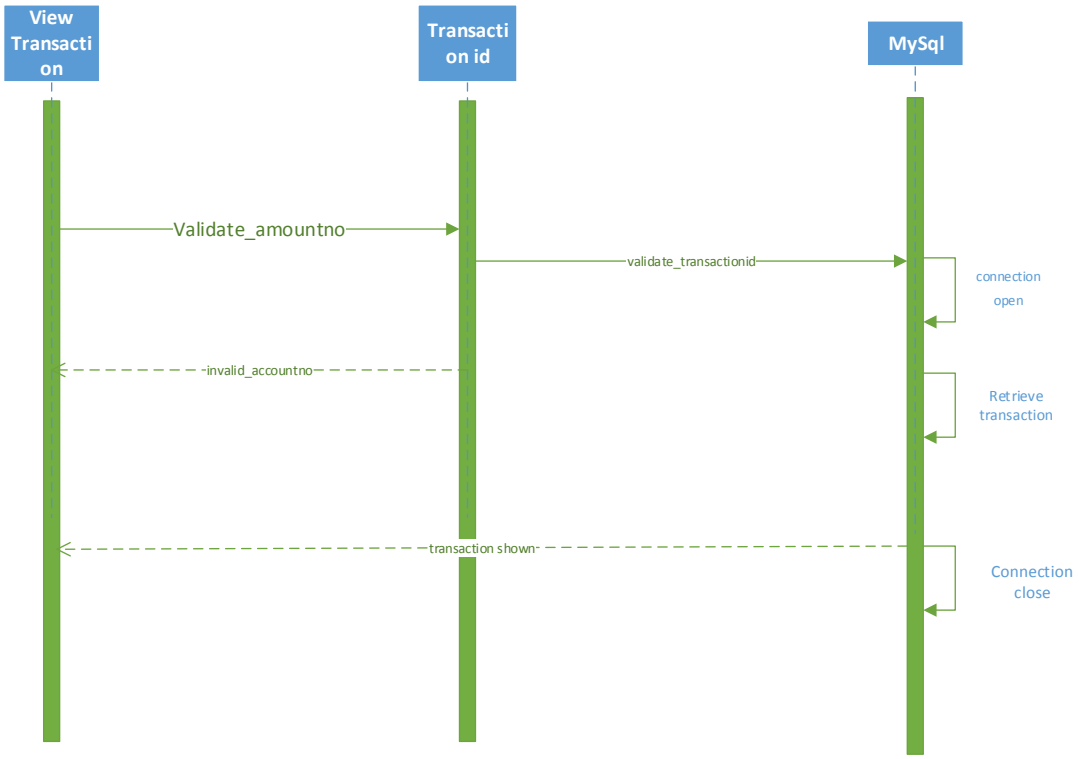
Add/Edit Employee



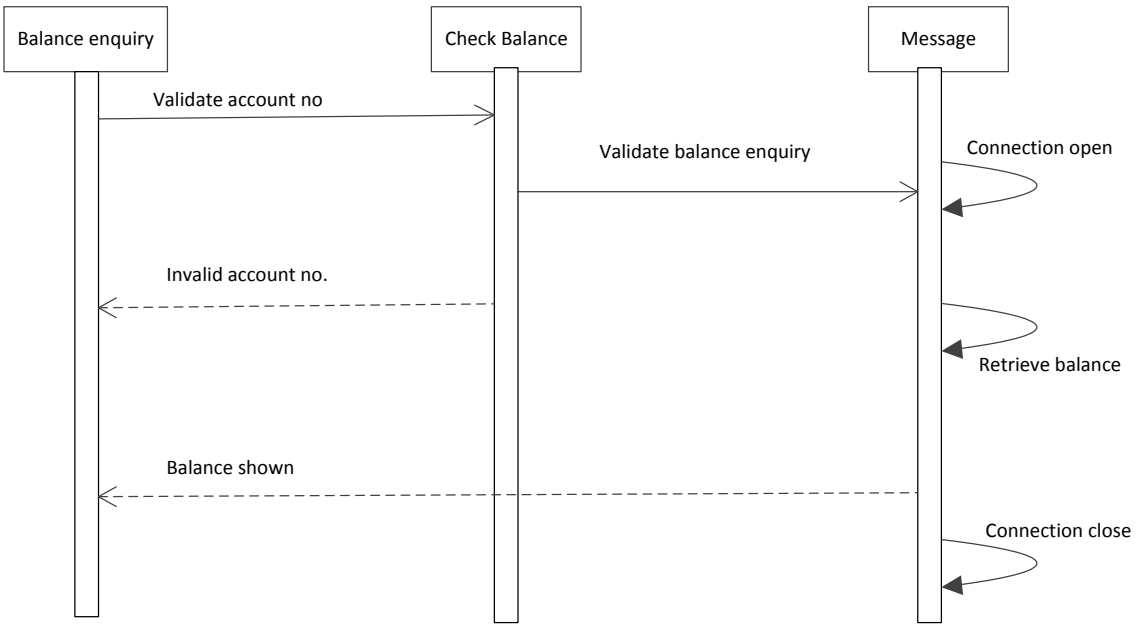
Delete Employee



Transaction

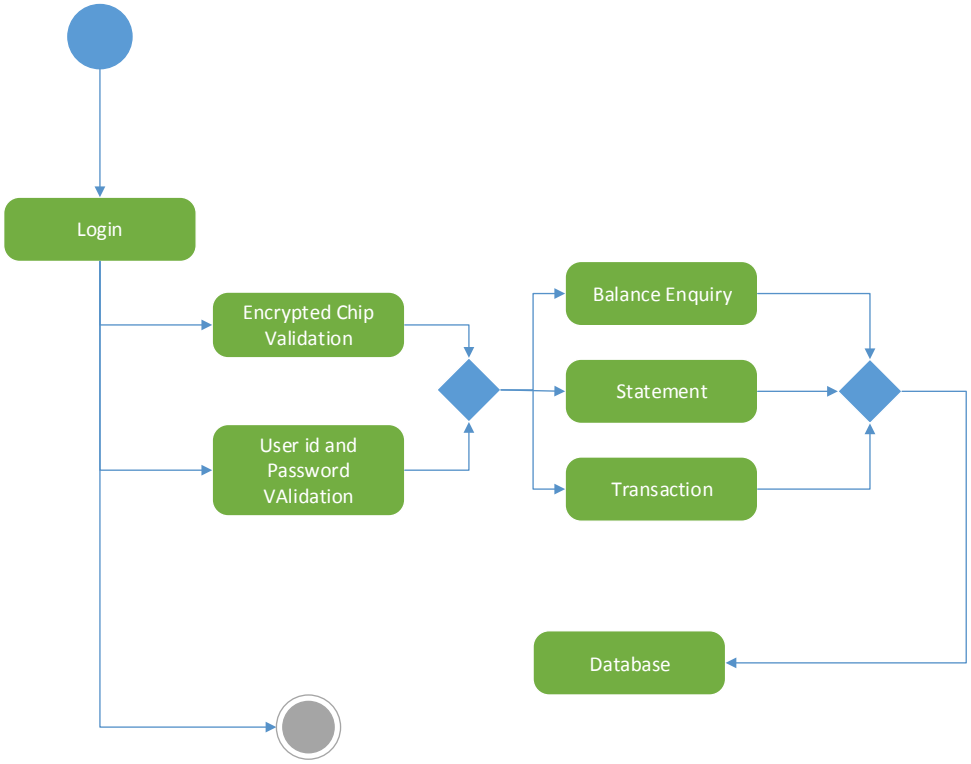


Balance Enquiry

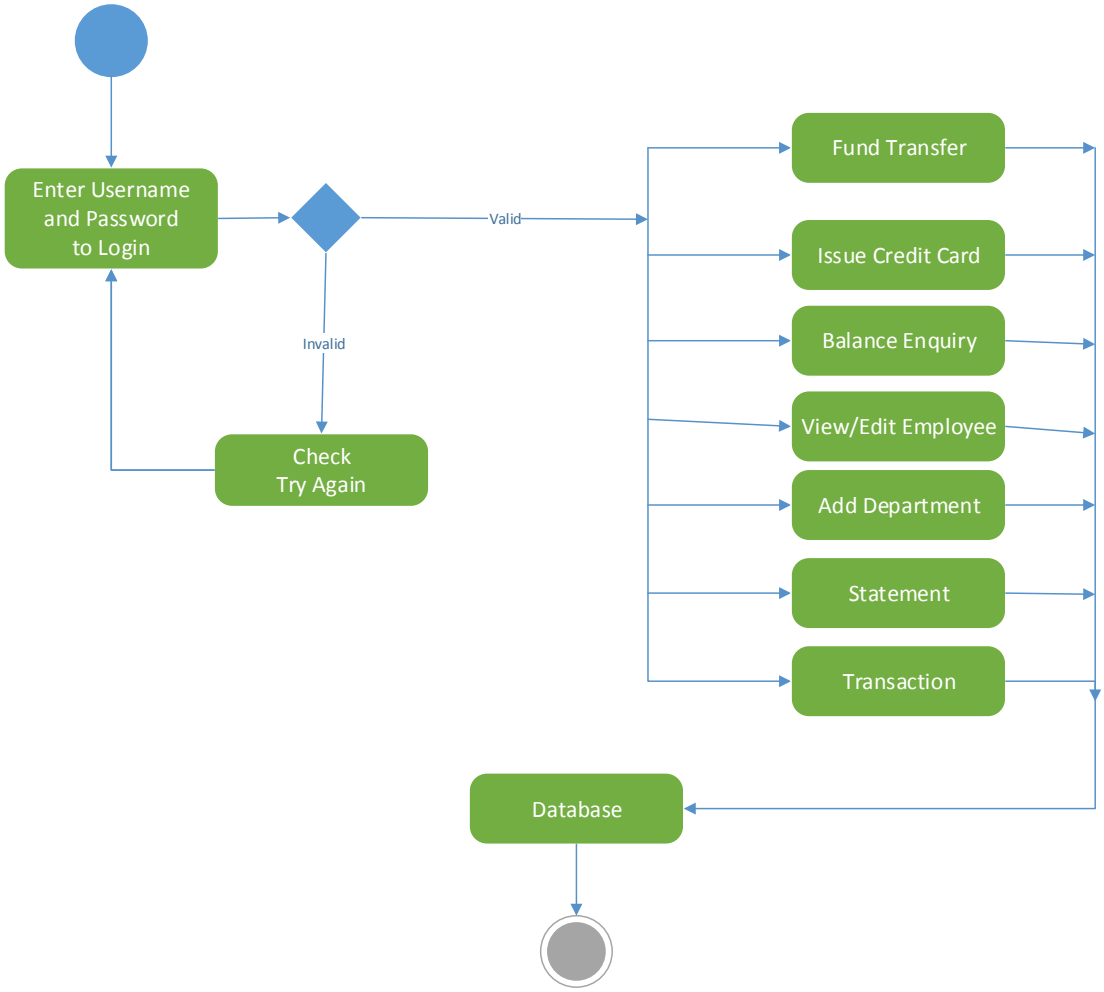


8.4 Activity Diagram

Customer

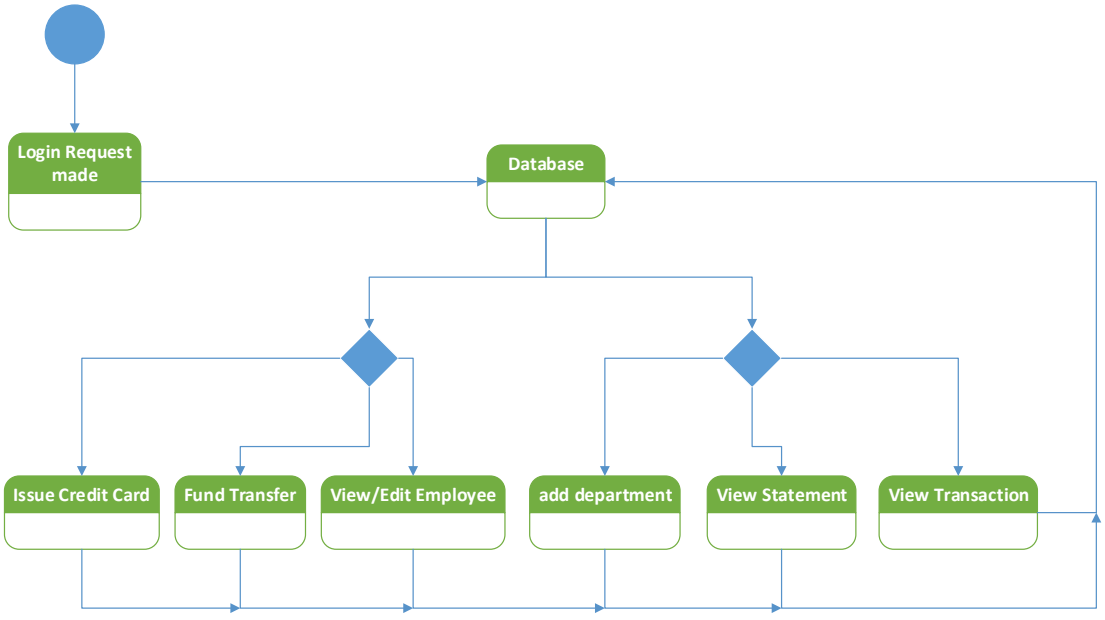


Admin

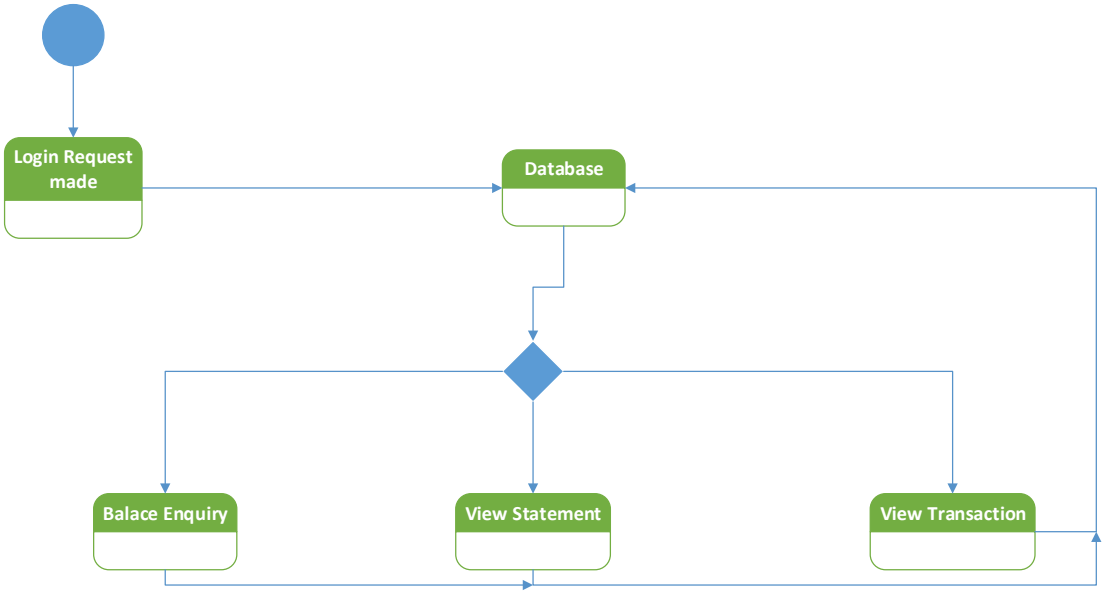


8.5 State Transition Diagram

Admin



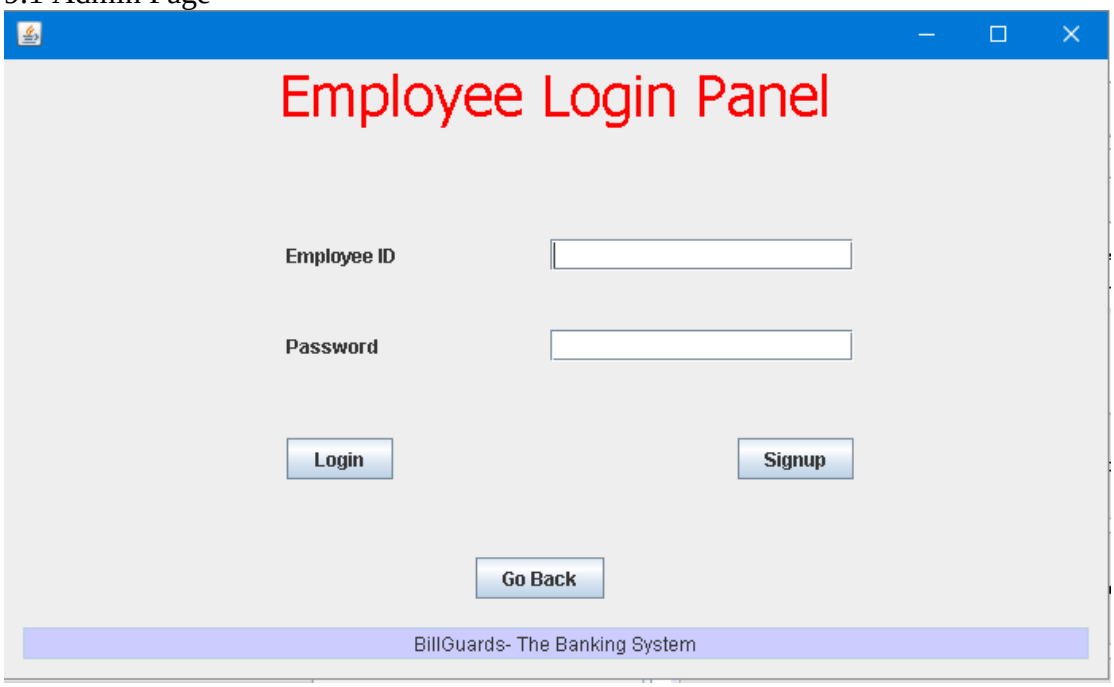
Customer


Login Request made
Database
Balace Enquiry
View Statement
View Transaction

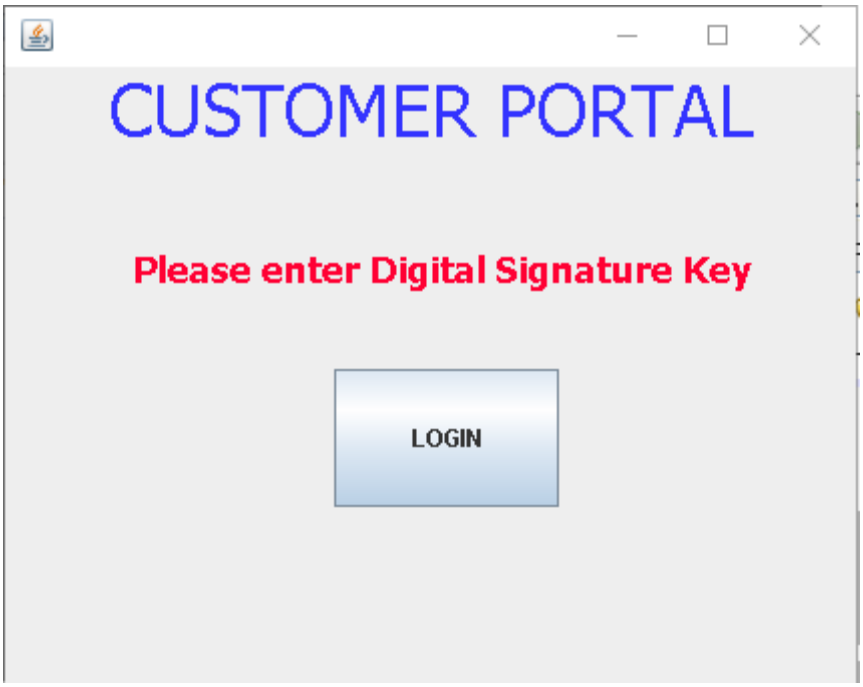
9.0 User interface design

Below are prototype designs of the interface.

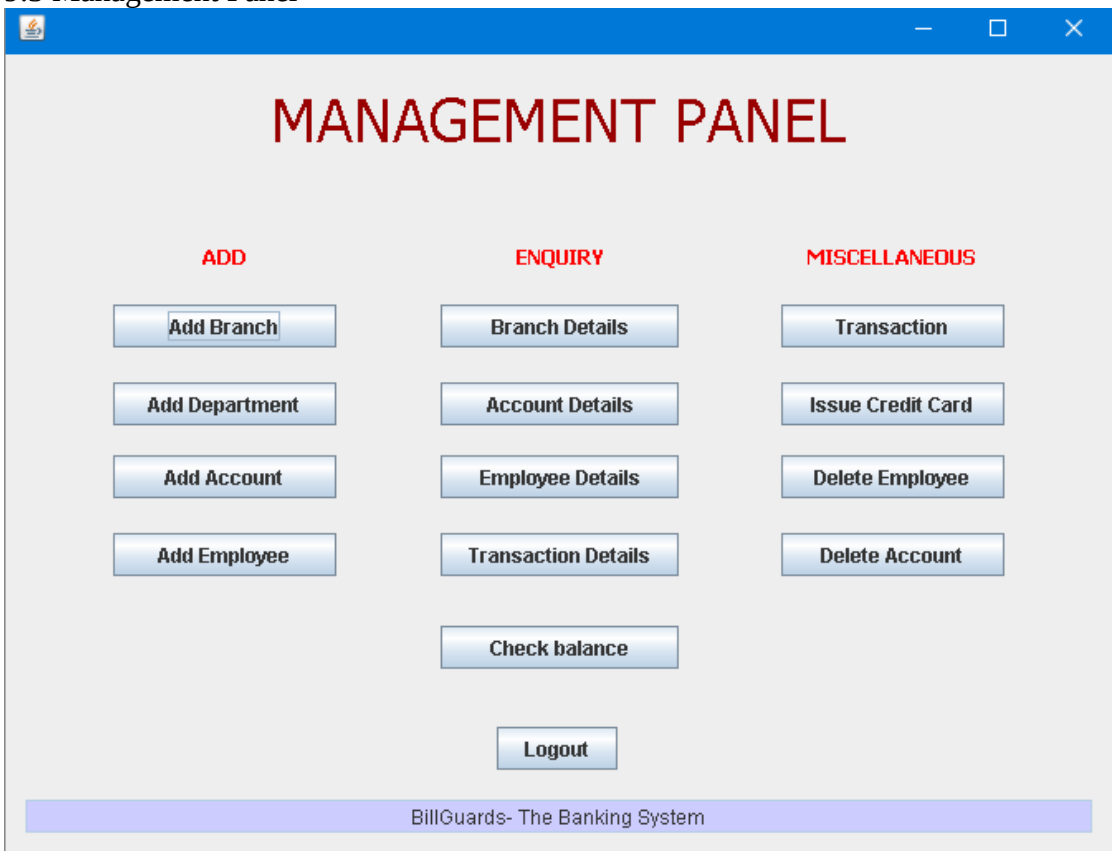
9.1 Admin Page



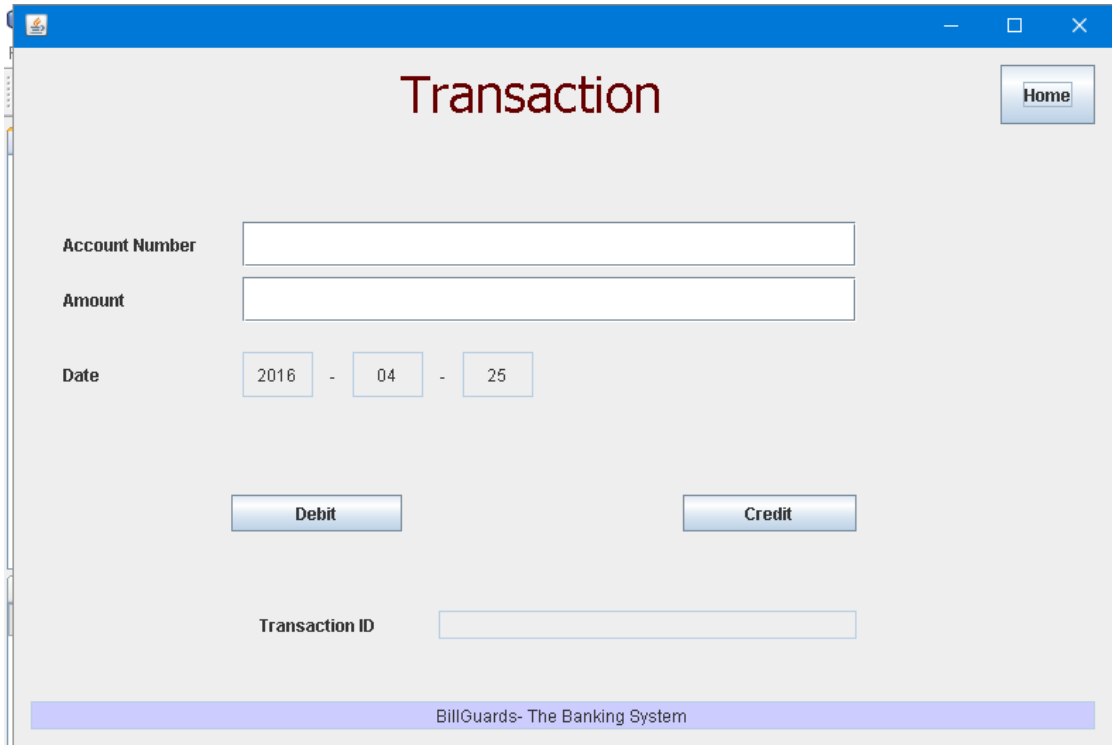
9.2 Customer Portal



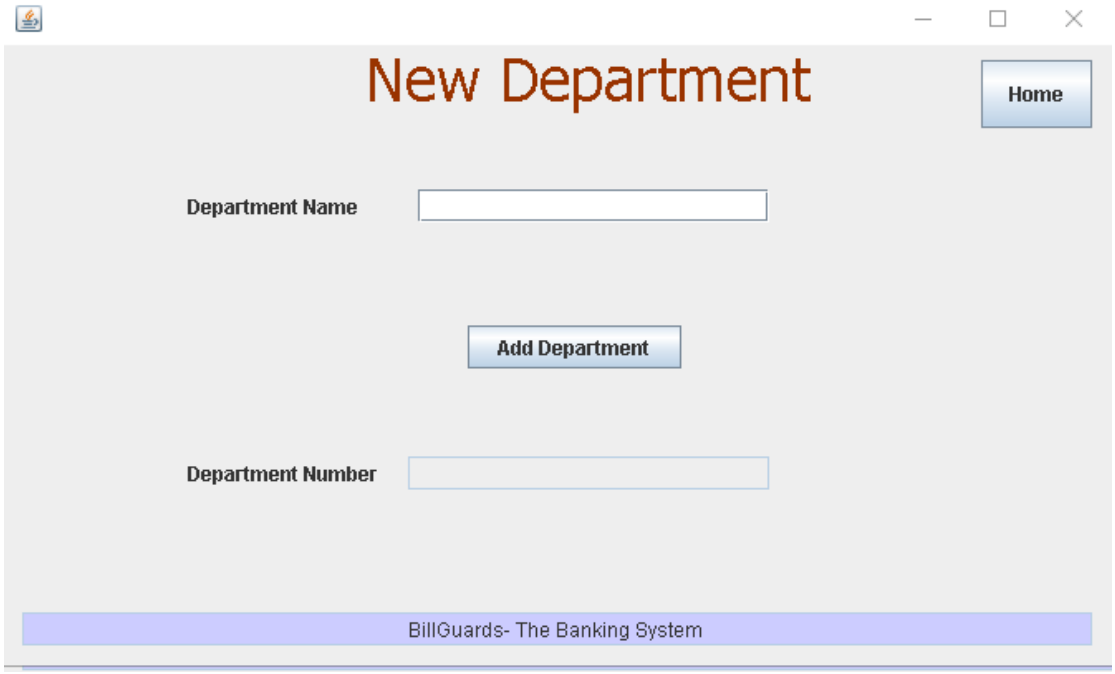
9.3 Management Panel



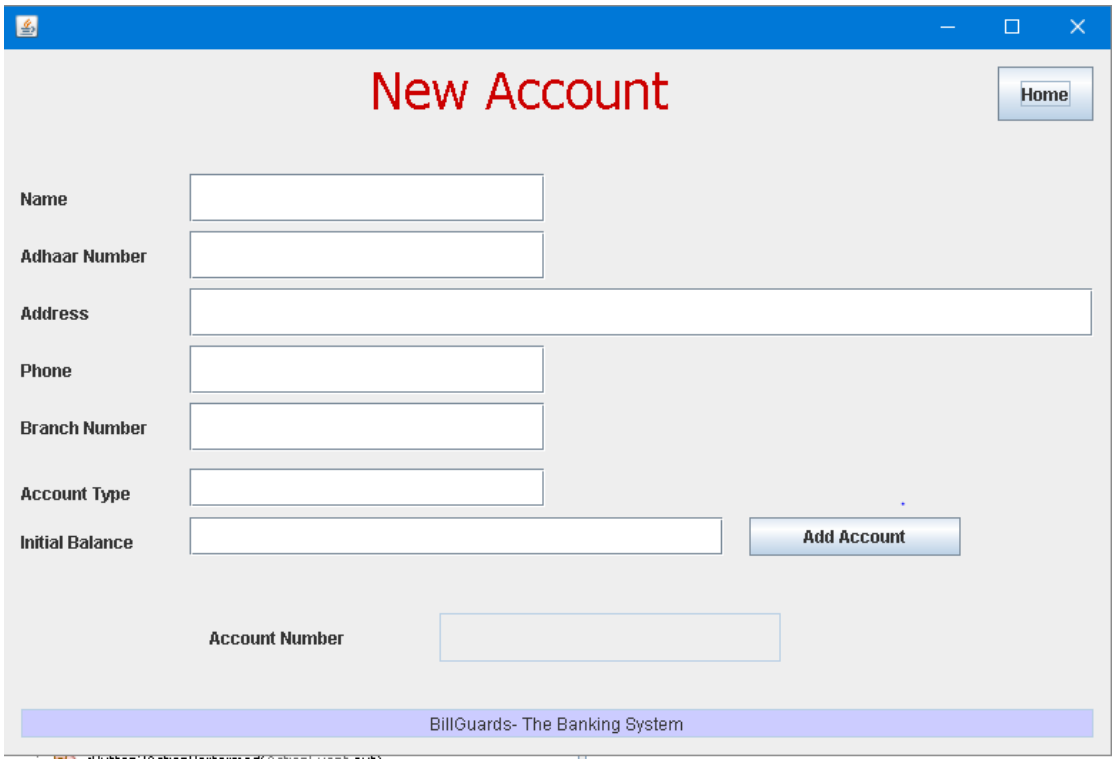
9.4 Transaction



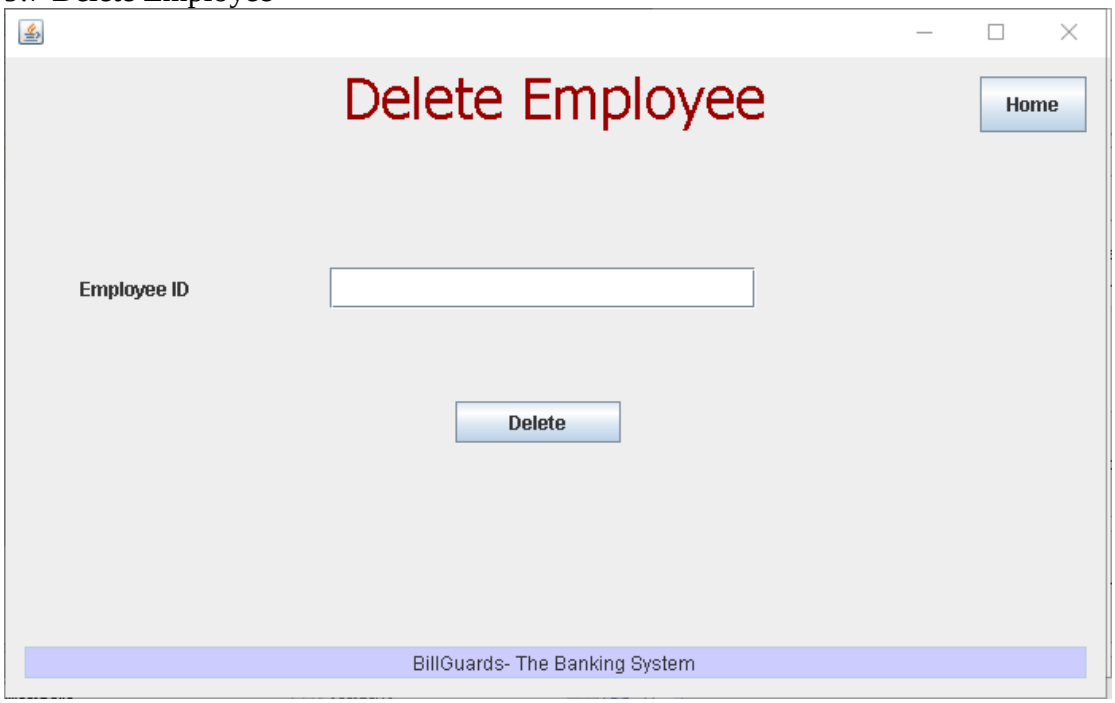
9.5 New Department



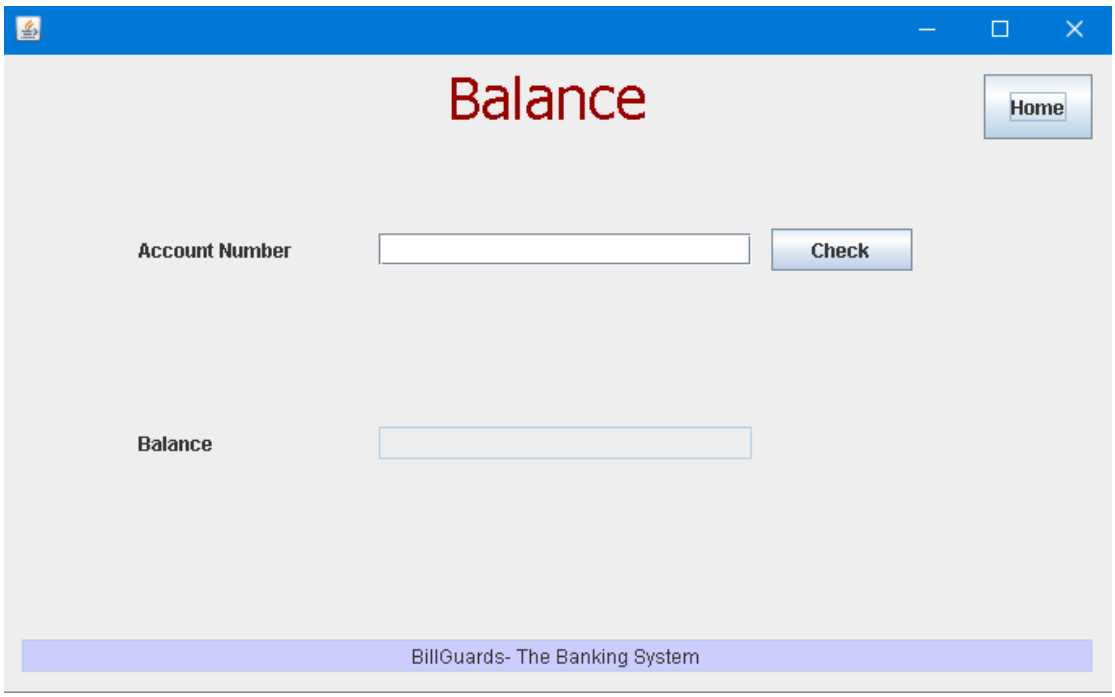
9.6 New Account



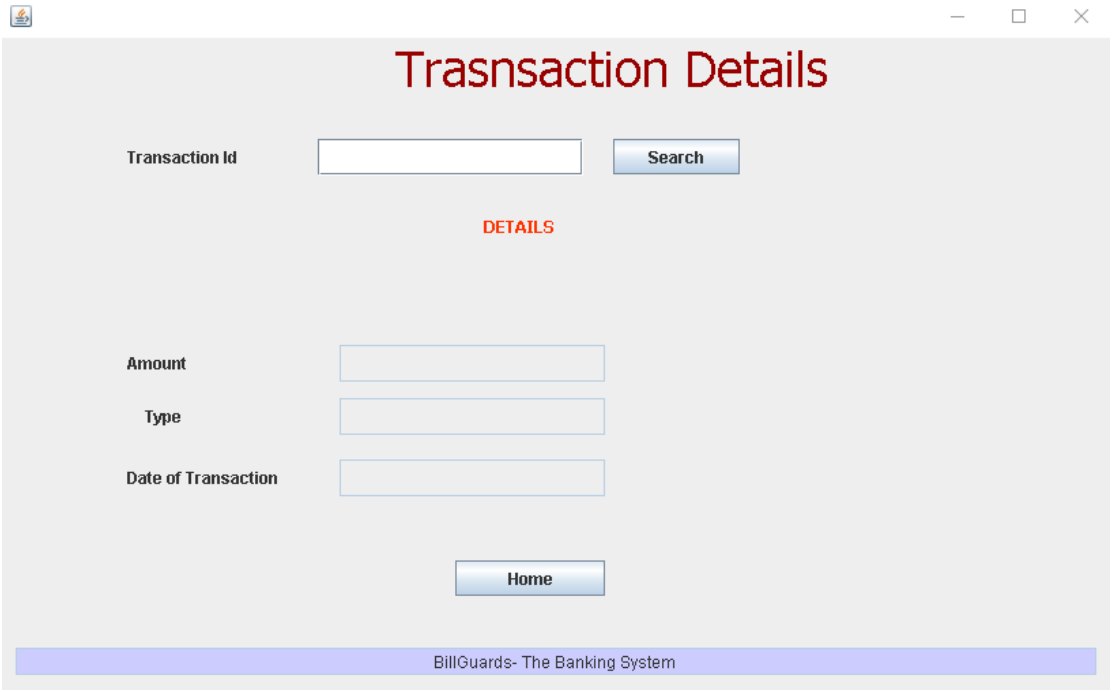
9.7 Delete Employee



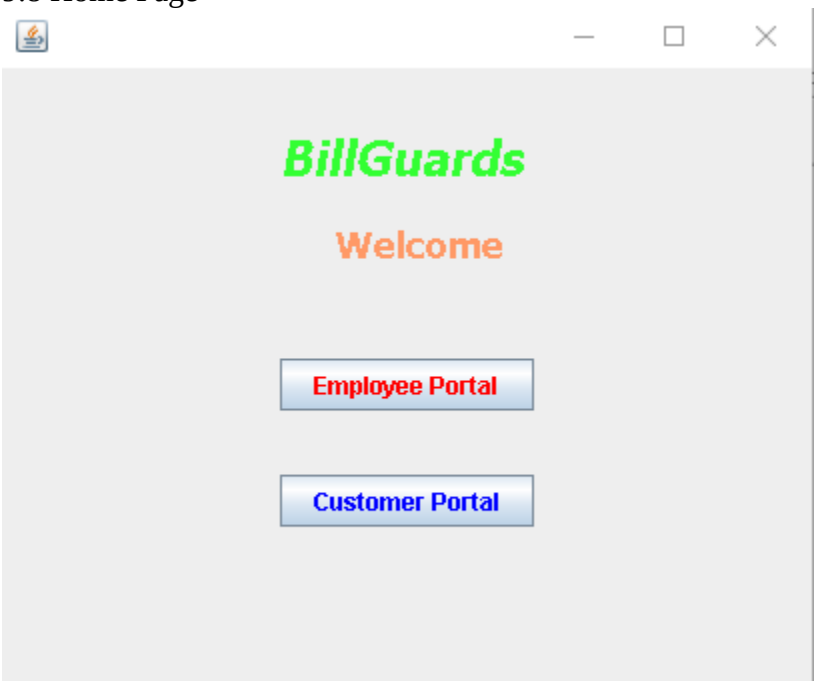
9.8 Balance



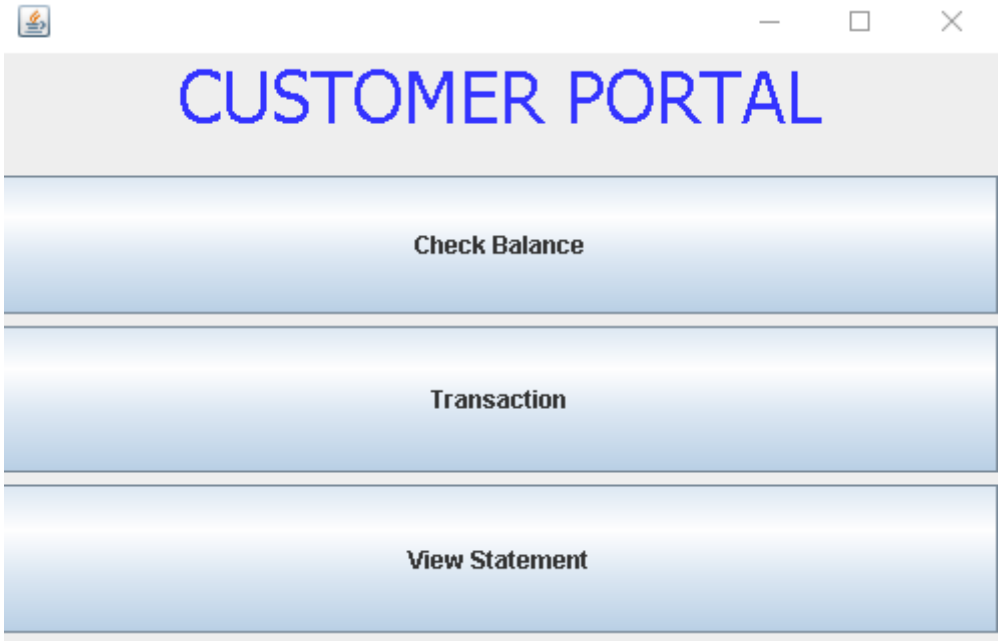
9.7 Transaction Details



9.8 Home Page



9.9 Customer Portal



11.0 Performance

The most important factor in the working of the whole project is its Connectivity with the Server.

•Speed: The system should be made as fast as possible to reduce response time.
•Throughput: The throughput should be as high as possible. We should be able to attain maximum output in minimum time.
•Capacity: We should try to make it accessible to maximum users at a time.
•Resource Utilization: Resources are modified according to user requirements.

12.0 Safety and Security

The Central Server comprises of the MySql Server 4.17. Both are open source software solutions and the best among their fields. They have their respective Security Mechanism which would prevent any unauthorised access or exploitation of the Server.

12.1 Login requirements

12.1.1 Users will be provided access to the system after they are registered by their recruitment officers
12.1.2 Users will be logging into the system using their ID and one time password emailed to them.
12.1.3 On login they will be asked to change their passwords

12.2 Password

12.2.2.1 Passwords must have a minimum length of 8 characters
12.2.2.2 Passwords must meet at least 3 out of the 4 requirements for quality:

a)at least 1 lower case letter
b)at least 1 upper case letter
c)at least 1 number
d)At least 1 special character (?, *, %)

12.2.3 Password should not contain the user's first name, middle name, last name, or username.

4.2.2.4 Passwords on sensitive IT systems must be changed, at a minimum, every 90 days.

12.3 Inactivity timeouts

12.3.1 System should timeout when there is no activity for five minutes.

10.0 Restrictions, limitations, and constraints

In order for the BillGuards-The Banking Manager to be downloaded and installed ,the user must be using a mobile device running Windows Operating System version 8 and above, as well as having access to the Windows Marketplace. All development for the app was done on Windows 10 machines. Testing of the application was done on Windows 10. As long as the Windows Marketplace is available to Microsoft's Windows Operating System users, the software will be marketable, maintainable and functional to both users and developers.

11.0 Testing Issues

11.1 Classes of tests

We will conduct first tests on each individual app within the system as separate entities using the Windows. Once each individual app is tested thoroughly, the package will the built together and tested as a whole. All known valid input will be tested as well as known invalid input. A more comprehensive overview of our testing strategies will be included in our testing specification documentation.

11.2 Expected software response

Each test performed will be clearly observed as either failing or succeeding.

11.3 Testing Report

Test Case	Check Item	Test case Objective	Steps to Execute	Test Data / Input	Expected Result
TC-001	Log-in Page	Leave all fields as blank and click Log-in button	Click Log-in		By leaving all fields as blank and on click Log-in button then mandatory symbol (*) should appear in front of Username and Password fields

TC-002	Username	Enter Invalid Username	NA	Username : Jackk	By entering invalid Username then an error message should appear as " Please Enter Valid Username "
TC-003	Username	Enter valid Username	NA	Username : Jack	It should allow the user to proceed
TC-004	Password		NA		The password field should display the encrypted format of the text typed as (****)
TC-005	Password	Enter wrong password	NA	Password : ***	By entering invalid password then an error message should appear as " Please Enter Correct Password "
TC-006	Password	Enter Correct password	NA	Password : *******	It should allow the user to proceed
TC-007	Log-in button	Correct Inputs	Click Log-in		It should lead the user to the respect page
TC-008	Forgot Password	Check hyperlink on Forgot Password label			while mouse over of the label an hand icon should display
TC-009	Forgot Password		Click Forgot Password		User can recover the password using the “Forgot Password” link page
TC-010	Registration	Check hyperlink on Registration label			while mouse over of the label an hand icon should display
TC-011	Registration		Click Registration		On click " Registration " page should redirect to the User Registration page

12.0 Appendices

12.1 Packaging and installation issues

The Software will be packaged and distributed as an applications installation package on the Website. It will be available to all users of devices running the Windows Operating System.

12.2 Legal Considerations

We will be using the NetBeans IDE in accordance to the NetBeans IDE License Agreement distributed by Microsoft (Copyright holder of the NetBeans IDE). This agreement grants us as developers “limited, worldwide, royalty-free, non-assignable, and nonexclusive rights to use the IDE solely to develop applications for the Windows platform.”