1. **Introduction**
   1. **Purpose**

This document is outlines the specifications of “Unicorn Banking System” which will be referred to as UBS or “the system” throughout the document. This document is intended to be referenced by developers who will be building the system. In addition, it will be used by software quality assurance team to verify the requirements and will be reviewed by the customer of the system for requirements approval.

* 1. **Scope**

This document contains a high level description of UBS in addition to a list of detailed requirements. Requirements include both high level guidelines and specific functional requirements. The document also provides an overview of the intended users of UBS. The document aims to describe the requirements in sufficient detail for developers to create the initial version of the system.

* 1. **Definitions, Acronyms and Abbreviations**
* GIC: Guaranteed Investment Certificate. A type of investment account where the interest is fixed and predetermined and the investment duration is typically in 10 years.
* HII: High Interest Investment. A type of investment account where the interest rate increases with more cash invested. This account type does not model any specific real world account and was made only for this system.
* Business day: a day other than Saturday, Sunday, a statutory holiday and any day normally observed as a holiday by the bank [1].
* Billing cycle: the time interval between the dates on which regular periodic statements are issued [2].
* GUI: Graphical User Interface
* UI: User Interface
  1. **References**

[1] Canada Post, January 13, 2014,  
 <http://www.canadapost.ca/tools/pg/customerguides/CGterms-e.asp#1012185>

[2] Mark McCracken, 2007,   
<http://www.helpwithmybank.org/dictionary/index.html#b>

[3] Dmitry Fadeyev, April 15, 2009,  
[www.usabilitypost.com/2009/04/15/8-characteristics-of-successful-user-interfaces/](http://www.usabilitypost.com/2009/04/15/8-characteristics-of-successful-user-interfaces/)

[4] Microsoft, March 2006,   
http://msdn.microsoft.com/en-us/library/aa480484.aspx#regcompliance\_demystified\_topic4

[6] Wikipedia, January 29, 2014, <http://en.wikipedia.org/wiki/Bank_statement>

* 1. **Overview**

The rest of the document describes in detail the specifications of the system. Section 2 describes the users of the system and its general functionality and lists the business rules to be followed. Section 3 outlines the functionality of the system in terms of detailed requirements organized by user class.

# Overall Description

This system is a *prototype* of an online banking application. It aims to simulate some of the functionality of a real online banking system.

## Product Perspective

The focus will be on managing Investment accounts. In particular, two kinds of investment accounts will be supported: Guaranteed Investment and Non-Guaranteed Investment. The system will have three separate user interfaces with distinct functionalities to support three categories of users: the administrator, the teller and the customer.

### System Interfaces

This system does not interface with any external systems.

### User Interfaces

All users of the system in the three categories listed above will be accessing the system through a web interface. The web interface will be available through a web browser. For this prototype only desktop browsers will be supported. Future versions may add support for mobile browsers or applications native to certain mobile devices.

The system’s user interface will be designed to have the certain usability characteristics. In particular the user interface needs to be Clear, Familiar, Consistent and Forgiving [3].

### Hardware Interfaces

This system does not interface with any hardware.

### Software Interfaces

The system is standalone and it does not interface with any other software.

### Communication Interfaces

This system does not communicate with any other systems.

### Memory Constraints

There is no constraint on memory usage at this phase. After the system is tested on a local machine, its memory footprint will be profiled. Since the system will be deployed to a web server, the web server can be chosen to have sufficient memory for the system’s operation.

### Operations

The system will support three types of operations: administrator operations, teller operations and bank customer operations. A separate user interface will be implemented for each.

### Site Adaptation Requirements

Prior to initial operation of the system, the database must be seeded with at least one administrator account. From that account, teller and customer accounts can be created.

## Product Functions

The system will serve three categories (classes) of users: The administrator, the teller and the customer.

The following day to day banking operations will be available for the customer:

* Opening a bank account (with restrictions on account types)
* Transferring of funds between own accounts (with restrictions on account types)
* Viewing the statement for an account
* Editing of personal information

The teller will be performing banking operations on behalf of the customer in addition to administrative operations:

* Opening bank accounts
* Opening investment accounts
* Activating/deactivating a bank account
* Adding a new customer to the system
* Transferring funds between a customer’s accounts
* Recording deposit and withdrawal transactions
* Viewing a customer’s account statement

The administrator will be able to perform the following operations on user accounts.

* Creating a new user account (teller, customer or administrator)
* Disabling a user account (teller, customer or administrator)
* Resetting a blocked user account
* Editing a user account

The following section describes the rules that govern account management in UBS.

**GIC account:**   
This is a long term guaranteed investment account. Interest rate is a factor of the investment period where the maximum return is 25%.The interest rate is set at the time of purchase and the investment term is between 5 years and 10 years in 1 year increment. Withdrawal from this account is allowed but will incur a financial penalty of 0.5% on the withdrawn amount. The withdrawn amount will be deposited in the target account the next business day.

**HII account:**   
This is an investment account where the initial interest is a factor of the amount invested and the investment term. The investment term is between 1 and 5 years in 6 months increment. The invested amount can be withdrawn at any time. Interest rate is calculated by the formula: interest = initial interest rate + (0.1% for every extra 10000CAD invested). Interest rate is recalculated after withdrawals using the same formula but subtracting the withdrawn amount instead. The withdrawn amount will be deposited in the target account the next business day.

**Savings Account:**

The first version of the system will support one type of savings account. The interest rate will be 3% and will be deposited in the account monthly. The daily withdrawal limit will be fixed at 1000CAD.

**Checking Account:**

The system will support one type of checking account. The daily withdrawal limit will be set to 2000CAD.

**Credit card account:**

The first version of the system will support one type of credit card account. The annual interest of this account will be 20% and will be charged at the end of the billing cycle. The formula for calculating the interest is XXX. Credit card transactions will appear on the account statement after 3 business days.

**Check deposit delay:**

The system will allow depositing checks into either the checking account or a savings account. The amount will become available in the target account after 5 business days.

## User Characteristics

### User classes

This system will be designed to support the same user base as a real online banking system. Users of the system fall in three classes with different functionality available for each user class:

#### The Customer

* Definition:   
  The person who is using the bank’s services. Holds one or more banking account and may use the system for day to day banking operations in addition to investments.
* Characteristics:   
  A typical bank customer will be of age 18 and up with an equal gender distribution between male and female. The customer may use his/her online banking couple of times a week and his/her computer experience can vary from novice to advanced. The average customer might not feel comfortable with a complex system so he/she should be provided with a simple user interface. The customer should be able to perform banking operations reliably so the user interface should provide feedback in form of confirmations for successful operation and error messages for failed operations.

#### The Teller

* Definition:  
  The bank employee who is facing the customer. Can perform banking operations on behalf of the customer.
* Characteristics:  
  Typically between the age of 20 and 50. Tellers are required to have a certain level of computer proficiency and are trained to use the system. Tellers will have a more advanced user interface and more permissions than customers.

#### The Administrator

* Definition:   
  The bank employee who is responsible for supporting the system’s functionality. Has access to user accounts and the privileges to create and edit user accounts.
* Characteristics  
  Administrators are the most technically knowledgeable users of the systems. They will be trained to use the system and to provide support for tellers.

## Constraints

### Multiple login

Simultaneous login with the same user account from multiple devices will not be allowed.

### Localization

The user interface must be available in both English and French languages. The system must therefore support localization.

### Security

Connection to the server must be over SSL.

### Error Notifications

Any error that might impact the user will be reported to the user. In the case the error requires further actions on the customer side, instructions on how to proceed will be communicated to the user.

### Error Tolerance

The system should provide a mechanism to log failed operations. In the event of a failure that can potentially affect data integrity, the failure log can be analyzed so that data integrity can be verified and/or restored.

## Assumptions and Dependencies

* Customers will access the system through a modern browser.
* Server security will be provided by the hosting party.

# Specific Requirements

This section is organized by user classes that are defined in User Characteristics section. For every user class a set of functional requirements specific to that user class is listed.

## External Interface Requirements

### User Interfaces

The system will have a GUI accessible through a web browser. A separate user interface for each of the three user classes will be provided. Only desktop browsers will be officially supported however some features may work in some mobile browsers.

### Hardware Interfaces

The system will not interface with any hardware.

### Software Interfaces

The system will not interface with any software.

### Communication Interface

The system does not have any communication interfaces.

## Functional Requirements

### User class 1 – Customer

#### Functional requirement 1.1

**ID:** FR1.1  
**Name:** Customer login into the system  
**Details:** The customer shall be able to authenticate himself/herself with the system and obtain access to the customer section of the system.

#### Functional Requirement 1.2

**ID:** FR1.2  
**Name:** View own accounts  
**Details:** The system shall display a list of all of the customer’s accounts. For each account the following information is displayed: account name, number, balance and status.

#### Functional Requirement 1.3

**ID:** FR1.3  
**Name:** View account details  
**Details:** The customer can select an account to view in detail. The system shall display the following details of a customer account: Account number, balance, a list of transactions in the account within a specified date range. For each transaction the following information will be displayed: transaction description, recipient/source, type (debit/credit) and amount.

#### Functional Requirement 1.4

**ID:** FR1.4  
**Name:** Transfer funds between own accounts  
**Details:** The customer shall be able to transfer an amount from an account that he/she owns (source account) to another account that he/she owns (destination account). The amount cannot be greater than the amount available in the source account. The source account can be of type Checking or Savings and the destination can be of type Checking, Savings and Credit Card.

#### Functional Requirement 1.5

**ID:** FR1.5  
**Name:** View account statement  
**Details:** The customer shall be able to view the statement for each of his/her own accounts. The statement shall display the following information: The start and end dates of the statement, the number and the type of the account, a list of transactions that in the account and that were carried out between the start and end dates of the statements and the opening and closing balance of the statement [5]. For each transaction the system shall display the following information about the transaction: date, reference (id), description, amount, type (deposit/withdrawal) and the balance of the account after the transaction is carried out.

### User class 2 – Teller

#### Functional Requirement 2.1

**ID:** FR2.1   
**Name:** Tellerlogin into the system   
**Details:** The Teller shall be able to login into the employee section of the System.

#### Functional Requirement 2.2

**ID:** FR2.2   
**Name:** Find a customer   
**Details:** The teller shall be able to find a customer by the following search parameters: client card number, account number, first / last name. In the case of searching by name multiple search result could be returned.

#### Functional Requirement 2.3

**ID:** FR2.3   
**Name:** View the accounts of a customer.   
**Details:** The teller shall be able to view a list of all the accounts for a given customer.

#### Functional Requirement 2.4

**ID:** FR2.4  
**Name:** Transfer funds from one account to another.   
**Details:** The teller shall be able to transfer funds, on behalf of a customer, between two accounts owned by the customer.

#### Functional Requirement 2.5

**ID**: FR2.5   
**Name:** Create new bank account.   
**Details:** The teller shall be able to open a new account for a customer. The new account can be one of the following types: Checking, Savings, Credit.

#### Functional Requirement 2.6

**ID:** FR2.6   
**Name:** Change the status an account  
**Details:** The teller shall be able to change the status of an account from active to inactive and vice-versa.

#### Functional Requirement 2.7

**ID:** FR2.7  
**Name:** Open GIC account  
**Description:** The teller shall be able to open a new GIC account for a customer. The system will prompt for the following information: duration of the investment in increments of one year.

#### Functional Requirement 2.8

**ID:** FR2.8  
**Name:** Create High Interest Investment Account  
**Description:** The teller shall be able to create a HII account for a customer. The system shall prompt for the following information: duration of the investment in increments of six months.

#### Functional Requirement 2.9

**ID:** FR2.9  
**Name:** Deposit in an Investment Account  
**Description:** The teller shall be able to deposit/transfer an amount in an investment account of type GIC or HII. The amount will appear in the investment account the next business day.

#### Functional Requirement 2.10

**ID:** FR2.10  
**Name:** Withdraw from an Investment Account  
**Description:** The teller shall be able to transfer an amount from an investment account of type GIC or HII to one of the customer’s accounts. For a GIC account the bank will charge a financial penalty to the customer if the withdrawal is applied before the end of the investment term. The amount will appear in the destination account the next business day.

## Performance requirements

Client response time should be no longer than 30 seconds. To achieve this speed the system should be deployed on a server with sufficient resources. In particular the server should have sufficient memory and network bandwidth.

## Logical Database Requirements

The following diagram is a high level logical model of the main entities of the system.

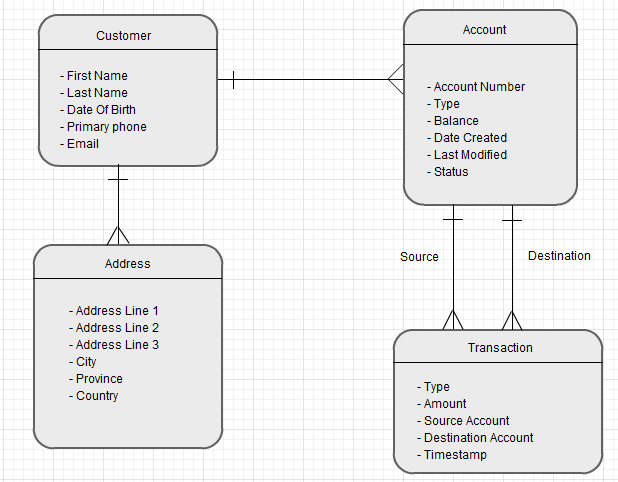


Figure 1

## Design Constraints

* Login and password is used for identification of customer’s account and there is no facility for non users to login.
* Minimum resolution to be supported is 1024x768 so the design of the web pages must accommodate that.
* The system will be officially supporting four major browsers: Firefox, Chrome, Internet Explorer and Safari.

### Standards Compliance

Since the system is a prototype, standards compliance will not be enforced. Ideally, however, the system should comply with the Payment Card Industry Data Security Standard [].

From the technology perspective the system should comply with web standards set by the World Wide Web Consortium for browser compatibility.

## Software System Attributes

### Reliability

The system must pass a series of tests before being deployed. For this prototype the two following practices will be enforced:

* User input must be scrutinized on both the UI and business logic levels and any anomalies must be logged for further analysis.
* The system must pass a stress test to test its ability to handle concurrent access without data corruption.

### Availability

Real banking systems have 99.999% (or five nines) availability. Since our system is a prototype the aim is to make it available 99% (or two nines) of the time once deployed. This means that the system will be available 24\*7 with maximum downtime of 3.65 days per year.

### Security

* The system must be designed to prevent unauthorized access to user accounts.
* All users shall require a password in order to access the system.
* Passwords will be encrypted before they are stored.
* The user should not remain logged in indefinitely. The user session should expire within when the user is not actively using the system.
* After three unsuccessful attempts to login into the system the user's account will be blocked and will have to be reset manually.
* The customer will need to present himself/herself in person at the bank to unblock his account.
* Connection to the server will be over an encrypted SSL session.
* Vulnerability Scanners and more specifically web application scanners, otherwise known as penetration testing tools will be used which help to keep the software closed to exploitation.

### Maintainability

Test Driven Development mode is recommended for the development of the system. This will enforce stronger validation of business make introducing changes significantly less risky. To facilitate regression testing an automated tool like Selenium will be used.

### Portability

The system is not required to be portable. Using platform independent technologies for its development can only be recommended if it does not compromise security and maintainability.

# Change Management Process

Additional features and changes to requirements will be reflected in the SRS. The document will be stored in source control and will be given a new version number after every major amendment. Minor amendments will be tracked by source control.

# Document Approvals

Not applicable.

# Supporting Information

Appendices:

* Appendix A: contains context, database and sequence diagrams.

In this document, appendices are considered as part of the requirements and the SRS is considered incomplete without their inclusion.

# Appendix A

## Context Diagram

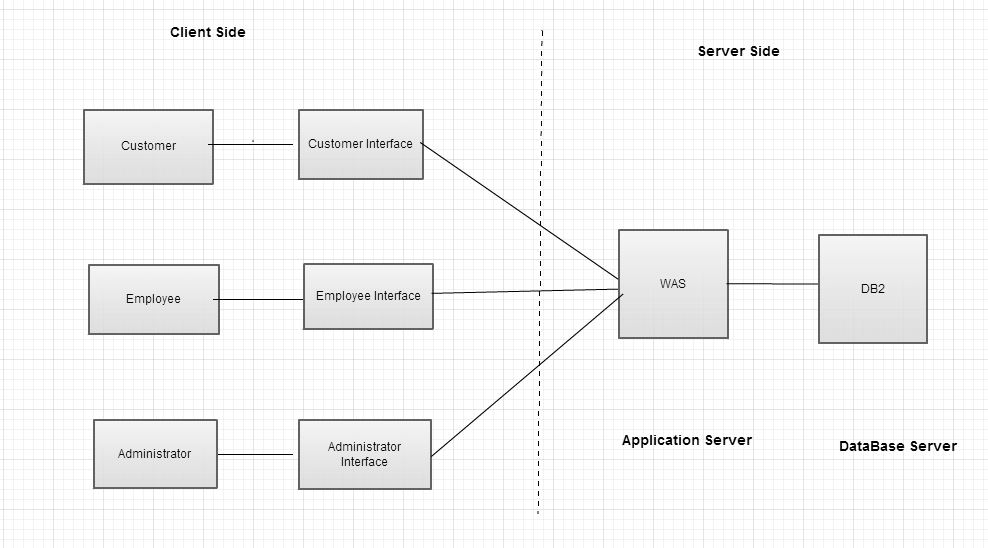


Figure 2

## Sequence Diagrams

### Customer login sequence Diagram

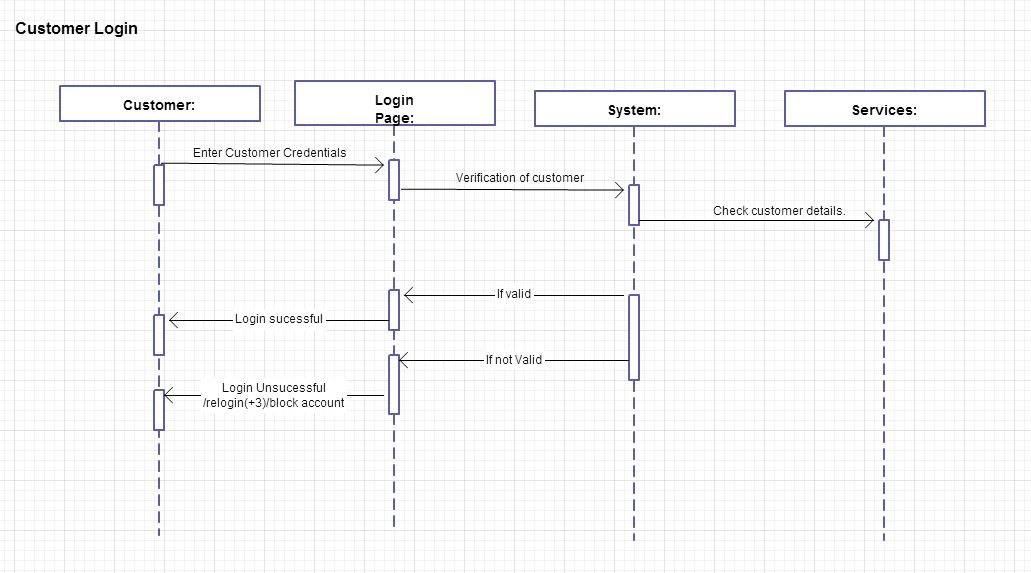


Figure 3

### View account statement sequence diagram

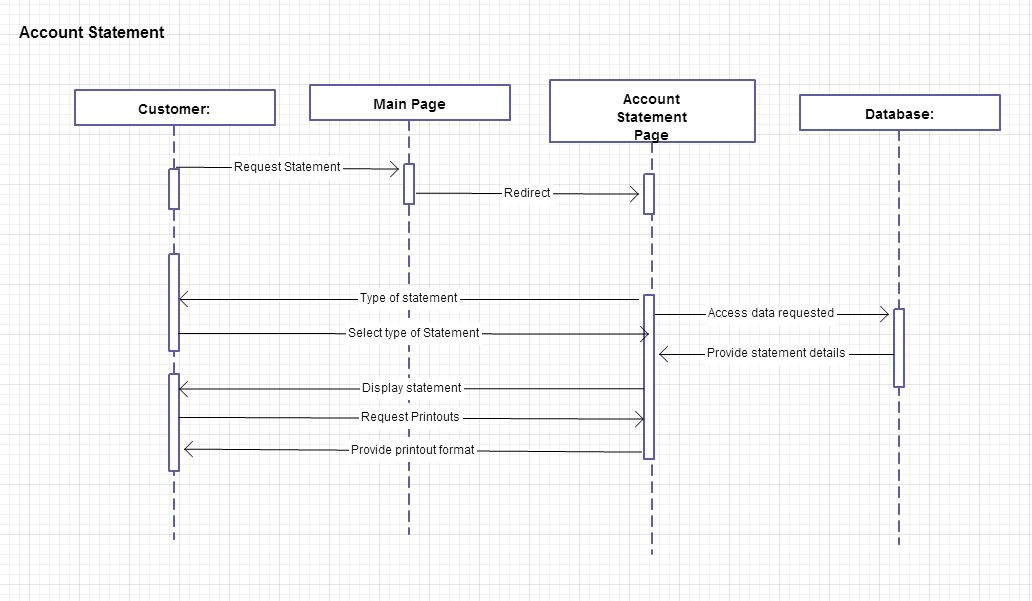


Figure 4

### Update personal info sequence diagram

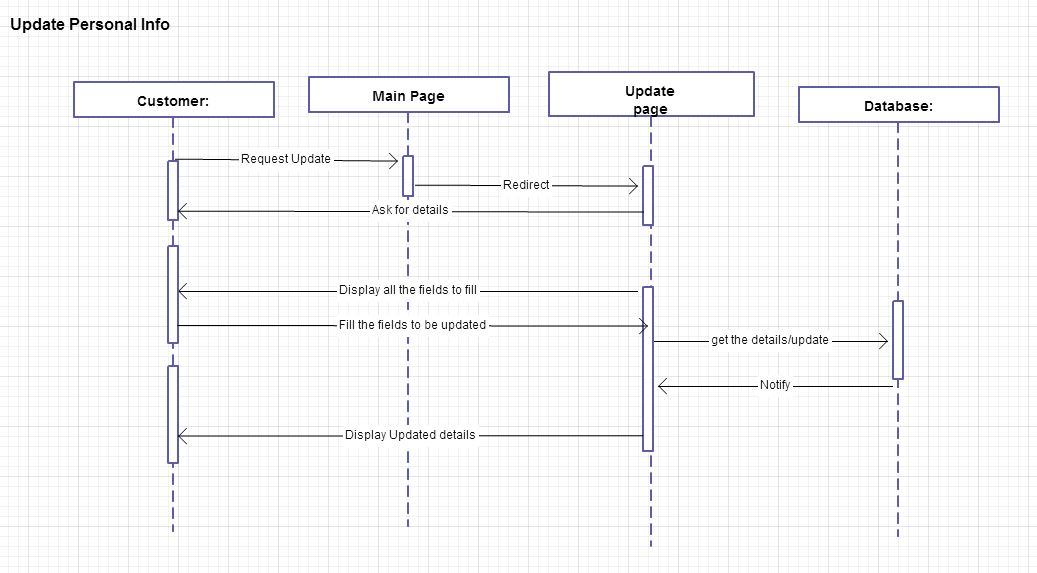


Figure 5

### Transfer funds sequence diagram

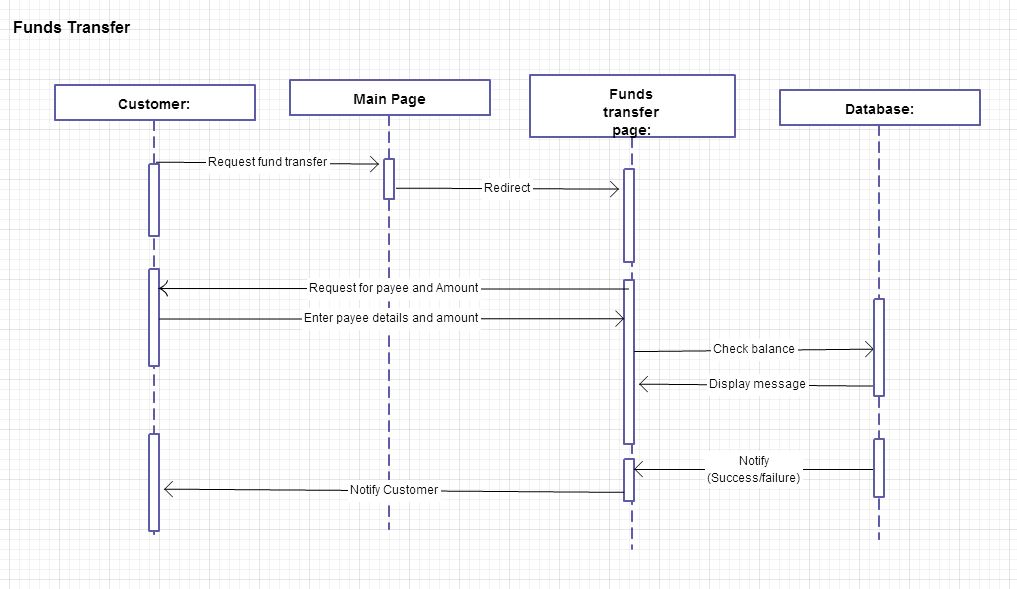


Figure 6

# State Diagrams

### Customer state diagram

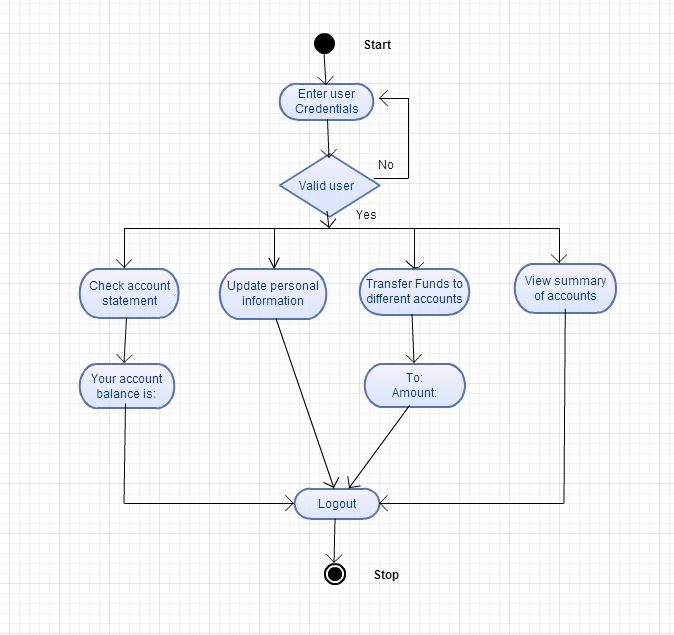


Figure 7

### Teller state diagram

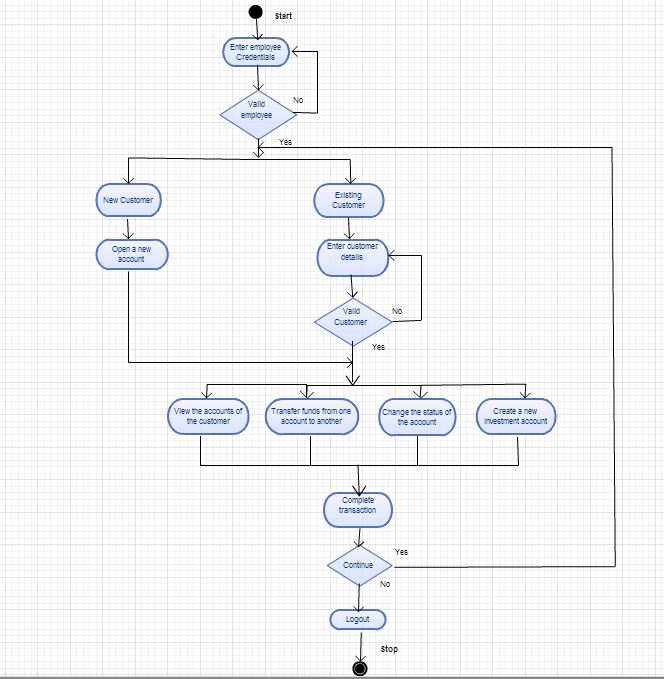


Figure 8