Name: Munjal Thakkar

Email: munjal@umd.edu

Software Design Specification

Net Banking Web Application

Version: 2.0

TABLE OF CONTENTS

1.0. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Net Banking System. This document will describe all the standards for design and implementation for the system. It contains multiple figures and designs which will not only be helpful for the development of the system, but also for the Testing of the System. This document is intended for both the ENPM809W authorities (Professor, TA, Grader) and me (developer of the system).

1.2. Scope of the Project

This software is a Net Banking Web Application which will enable the Bank's customers to utilize the Bank's features without having to go through the entire hassle of going to the bank. The Net Banking webapp will be having human interventions where needed, because of the sensitivity of the information involved. The web application will help the bank's customers to check their account balance, transfer funds to another account, withdraw funds or deposit funds in the bank.

1.3. Glossary

Term	Definition
ER Diagram	An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system.
Sequence Diagram	A sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering.

Normal Employee/s or Employee/s	A person who works for the bank can view any customer's details or manage their funds. They are useful, when the customer is performing withdraw/deposit (in physical activities)
Customer	Anyone other than the employees who has an account with the bank.
Database/DB	Collection of all the information in the system.
Software Design Specifications	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Account Balance or Funds	Money in the customer's account.

1.4. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998

https://www.lucidchart.com/pages/er-diagrams

https://en.wikipedia.org/wiki/Sequence_diagram

1.5. Overview of Document

This section briefly describes all the other sections in the document. The remaining document is as follow:

The next section describes the application, its architecture (with diagram) and the security concerns in detail.

The third section gives database overview, ER Diagram.

The last section tells us how the application is going to work from a design perspective. It contains multiple diagrams and also Use Cases.

2.0 Architecture

2.1 Overview

As this is a Net Banking Web Application meant for the customers of a Bank, the server has to be Internet facing and, on the Internet, upon which, the entire web application will be loaded. Thus, we have kind of a client server architecture.

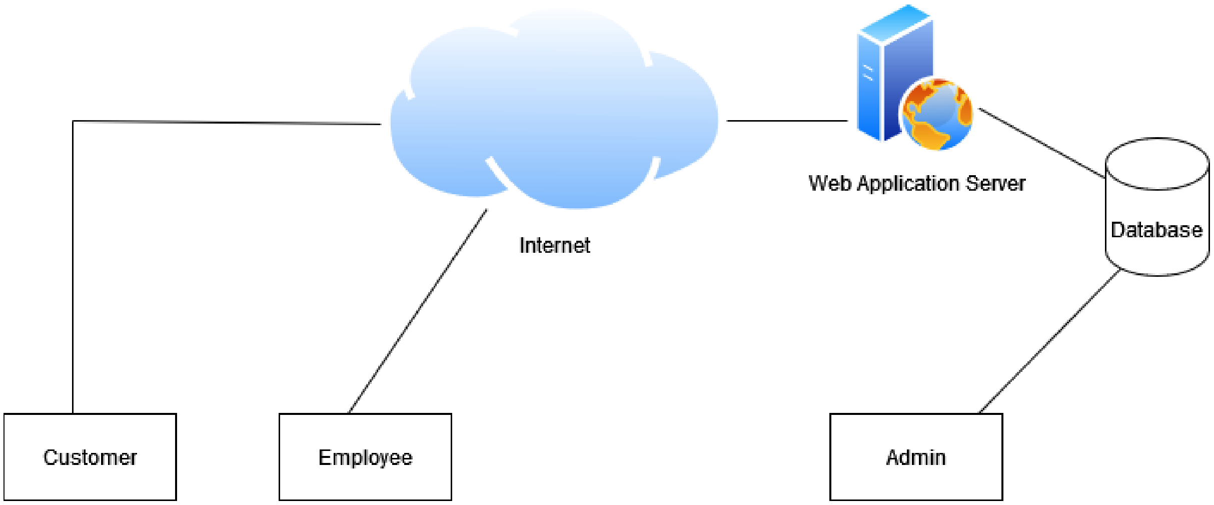
Here, as we can see, in the Architecture Diagram, customers connect to the website via the internet. Also, the Employees of the Bank are connected to the webapp via internet thus, allowing remote capabilities for all the banking work as well.

Although a lot of it is online, there are however some offline interactions which we have to consider. One of these applications is that, an employee can register for the creation of their account and then on their account being created, they will have to get a userID from the DBA of the bank, to access the Net Banking Web Application.

Also, for picking up or giving cash (withdraw or deposit), the customer has to be there on site to collect/deposit the money. But, to collect cash, the customer has to be on-site. Thus, this too is not remote functionality.

The main remote use case is that the customer can check his/her own funds present in the account. Or they can also transfer funds to some other person's account in the same bank. The admin does have a direct connection to the webserver and the database. If anything occurs, then the admin is the main person to be contacted.

2.2 Architecture Diagram



2.3 Security Concerns for this Architecture

- The Server:

  The entire Banking System lives on the server and its security is of utmost importance. Thus, the server needs to be safeguarded physically from intruders, as well as it has to be kept at a location where it is safe from natural calamities. The server is where the website will be hosted thus, a backup server should be there, and a Disaster Recovery Plan should also be made.

- The Database:

  The database contains the data of all the bank's employees and customers. It has all the monetary details and details of the transactions of the customers. Thus, its protection is of utmost importance. The database should be kept safe from the attackers, only the webservers IP should be whitelisted, all other IP's need to be blacklisted. The data can be fetched via a WPI or an API call that too made only by the server.

- The Employee’s system:

  If the attacker gets into an employee’s system and gathers the credentials, then it is very easy for the attacker to transfer the money into their account. One way to negate this threat considerably is to enable MFA from the employees’ phone whenever they want to login.

- Weak Passwords:

  Since everything and all accounts are linked with passwords, any kind of weak or common passwords cannot be tolerated. There should be strong password policies that are implemented. Moreover, the hashes of the passwords stored in the database have to be using secure algorithms like SHA256. Before storing the passwords, they need to be provided with some salt as well.

- Session Management:

  Proper Session Management also needs to be implemented so that an attacker cannot take another user’s session data wrongly.

3. Database Design

3.1 Database Overview

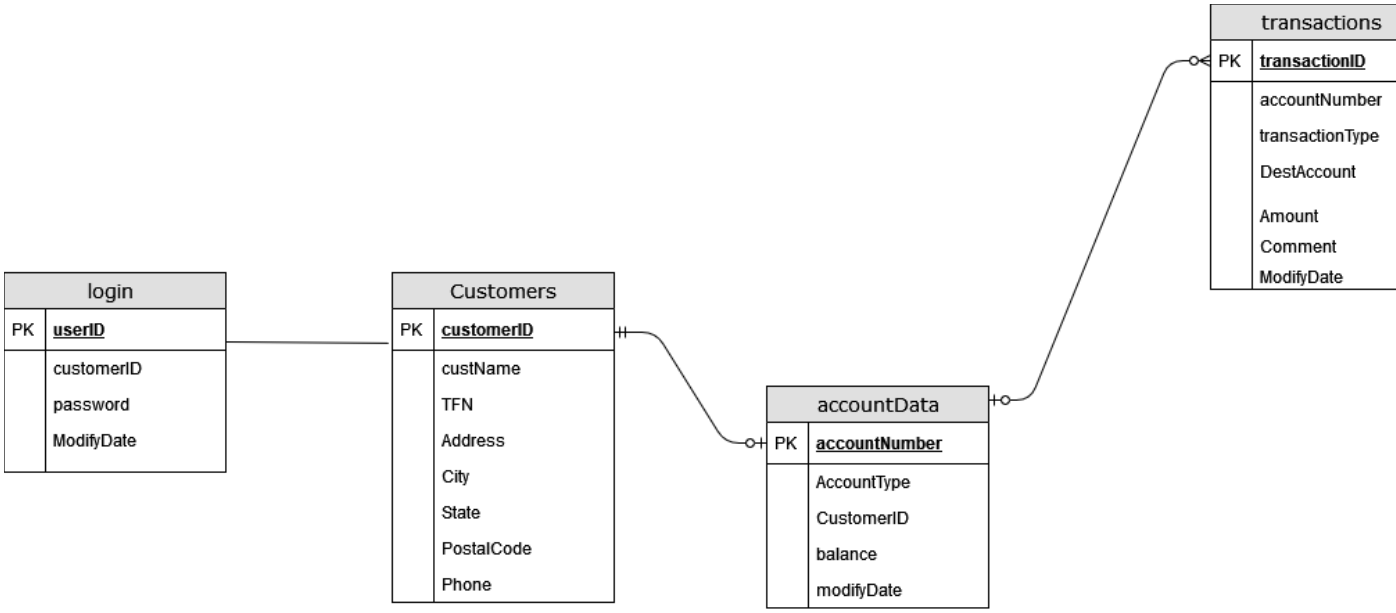
The ‘login’ database contains all details that are required when logging in. These details will be userID, CustomerID and Password (which is salted and hashed). The userID is the main requirement to login, which is given to a customer by the administrator (physically). We are not using Account Number for the same due to the fact that one user could have two accounts, and so, whenever that user logs in, we will need to display details for both the accounts.

‘login’ table is linked with ‘customer’ table via CustomerID field, and the customer table contains other important user information.

The ‘customer’ table is also linked with ‘accounts table which contains information of the accounts that a customer will hold.

The ‘transactions’ table is the final table that holds all of the transaction data inside itself. It has the transactionID field that separates all the transactions from each other and also adding comments is possible for a transaction.

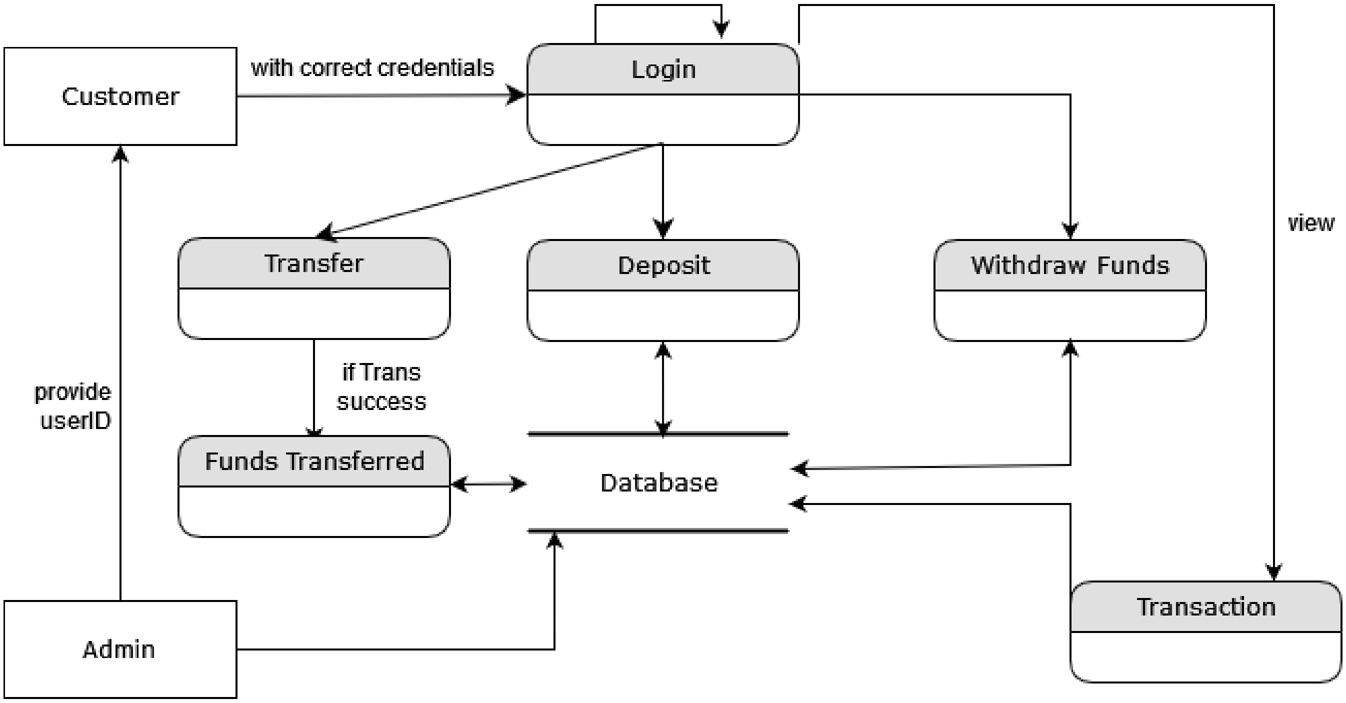
3.2 ER Diagram



4. System Interface Diagrams

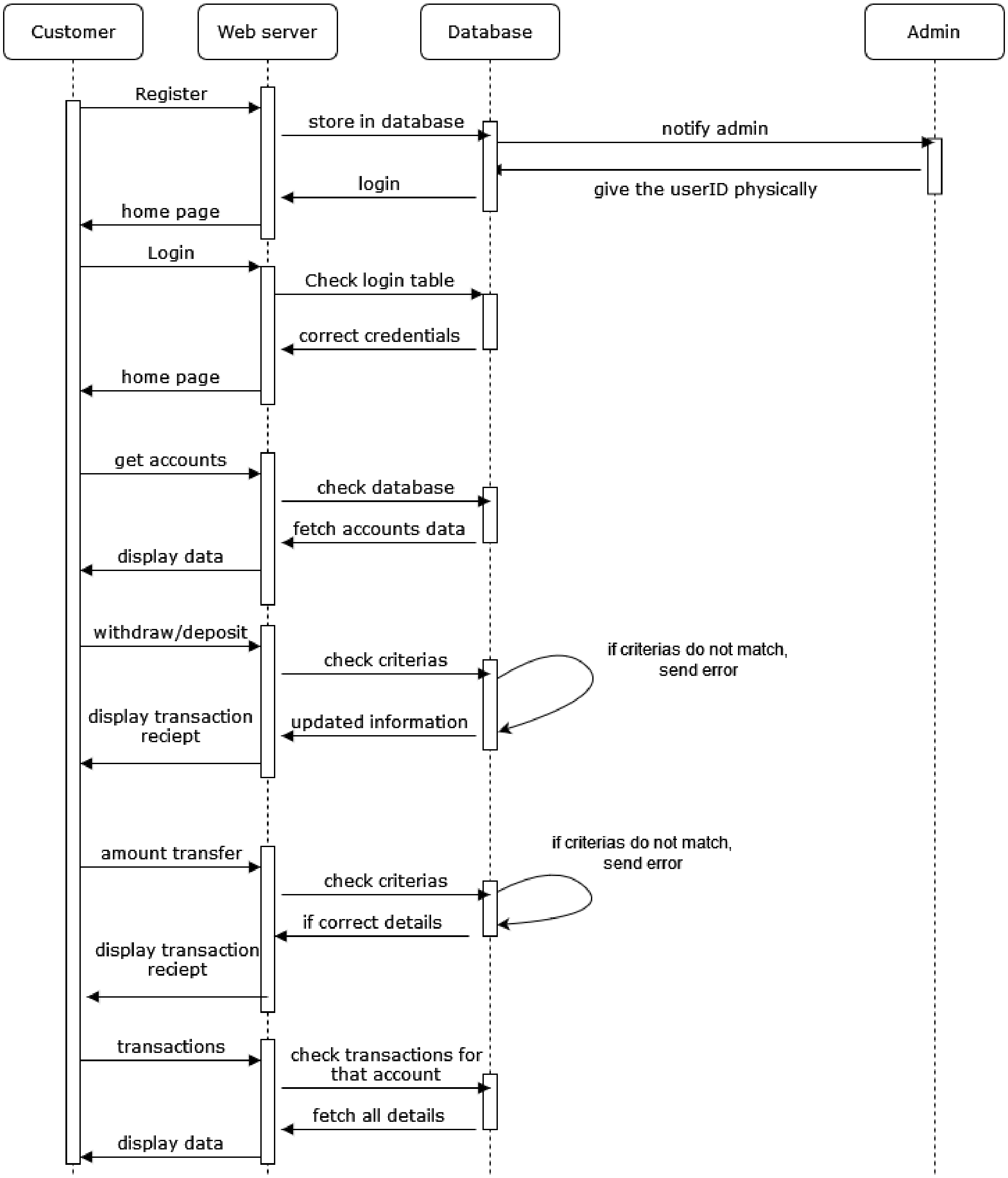
4.1 Data Flow Diagram

The data flow diagram shows us the data flow logic in the entire system.



4.2 Sequence Diagram

4.2.1 Sequence Diagram for Customers

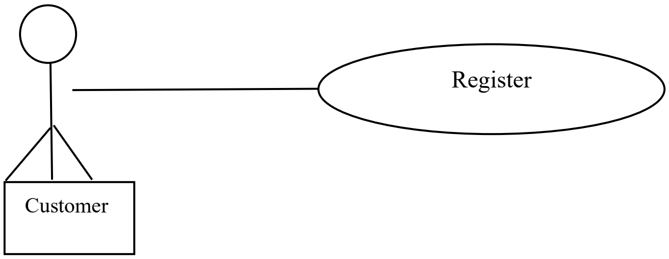


4.3 Use Cases

4.3.1 Customer Use Case

Use case: **Register**

Diagram:



Brief Description

The customers can register on the application.

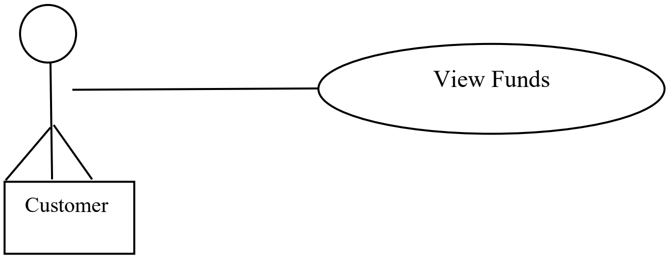
Initial Step-By-Step Description

Before this use case can be initiated, the customer has already accessed the index page and is **not** logged in as they do not have their own account.

1. The customer registers and gives their details.
2. Once the details have been submitted, the customer waits for a userID from the admin, which is given physically to the customer in this case.

Use case: **View Funds**

Diagram:



Brief Description

The customers view the funds in their bank account.

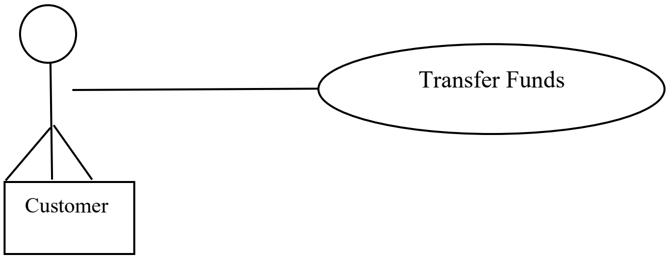
Initial Step-By-Step Description

Before this use case can be initiated, the customer has already accessed the home page and is logged in as the customer from their own account.

1. The customer selects My accounts option.
2. The amount of money in each of the bank accounts of that customer will be shown.

Use case: **Transfer Funds**

Diagram:



Brief Description

The customers transfer some money from their account to someone else holding an account in the same bank.

Initial Step-By-Step Description

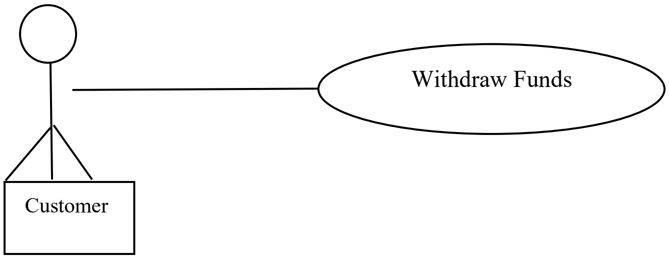
Before this use case can be initiated, the customer has already accessed the home page and is logged in as the customer from their own account.

1. The customer selects the ATM option.
2. On selecting that, the customer has to select the Transfer option.
3. Thereafter, the customer has to enter the correct amount, and the account nuumber whom they want to transfer the amount to.
4. The customer can also add a comment if they want to.
5. On submitting the transaction, the customer is shown the receipt of that transaction.

If everything is entered correctly, the transaction will be executed.

Use case: **Withdraw Funds**

Diagram:



Brief Description

The customers want to withdraw funds from their account.

Initial Step-By-Step Description

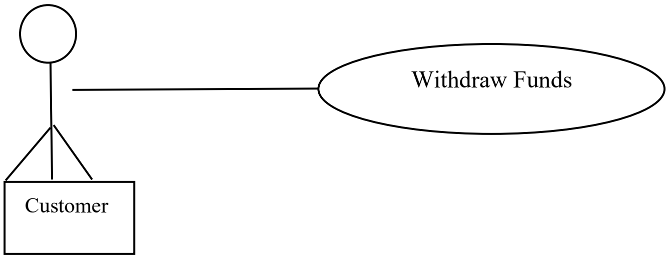
Before this use case can be initiated, the customer has already accessed the home page and is logged in as the customer from their own account. The customer is physically present at the location of the bank and is doing this in front of an authority.

1. The customer selects the ATM option.
2. On selecting that, the customer has to select the Withdraw option.
3. Thereafter, the customer has to enter the correct amount, and the account Number of the account they want to withdraw the money from.
4. The customer can also add a comment if they want to.
5. On submitting the transaction, the customer is shown the receipt of that transaction.

If everything is entered correctly, the transaction will be executed, and the banking authority will give the customer that amount.

Use case: **Deposit Funds**

Diagram:



Brief Description

The customers want to deposit funds into their account.

Initial Step-By-Step Description

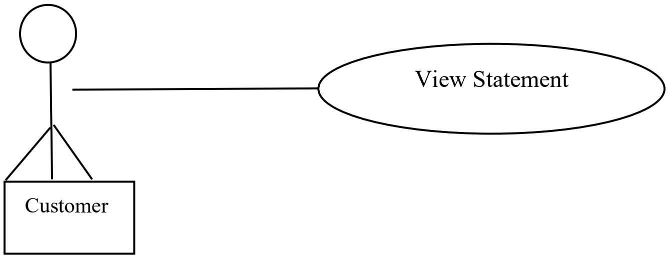
Before this use case can be initiated, the customer has already accessed the home page and is logged in as the customer from their own account. The customer is physically present at the location of the bank and is doing this in front of an authority.

1. The customer selects the ATM option.
2. On selecting that, the customer has to select the Deposit option.
3. Thereafter, the customer has to enter the correct amount, and the account Number of the account they want to deposit the money to.
4. The customer can also add a comment if they want to.
5. On submitting the transaction, the customer is shown the receipt of that transaction.

If everything is entered correctly, the transaction will be executed, and the banking authority will give the take that amount from the customer (since the customer is depositing money to their account).

Use case: **View Statement/Transaction Summary**

Diagram:



Brief Description

The customer wants to view the Statement (Transaction Summary) of their account.

Initial Step-By-Step Description

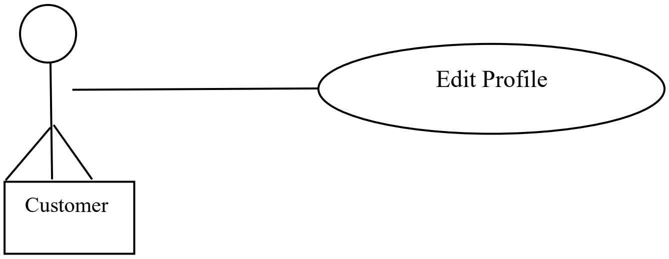
Before this use case can be initiated, the customer has already accessed the home page and is logged in as the customer from their own account.

1. The customer selects the My Accounts option.
2. On selecting that, the customer can see all the accounts in their name.
3. The customer clicks "View Statement" for the account whose transaction summary they want to view.

If the customer does everything correctly the transaction summary for the account will be shown.

Use case: **Edit Profile**

Diagram:



Brief Description

The customer wants to make changes to their Profile.

Initial Step-By-Step Description

Before this use case can be initiated, the customer has already accessed the home page and is logged in as the customer from their own account.

1. The customer selects the "My Profile" option.
2. On selecting that, the customer can see their current profile details.
3. The customer clicks on the "Edit" button to Edit their profile.

If the customer does everything correctly, the customer can change their profile data.

THE END