Bank Application

Technical Story Card

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Date | Prepared by / Modified by | Significant Changes |
| 1 | 15 June 2022 | Sudharsan V | Introduction added |
| 2 | 15 June 2022 | Aiswarya K | Use Case model added |
| 3 | 15 June 2022 | Chithra Priya G | Description added for User Registration |
| 4 | 16 June 2022 | Mounika Puttaparthi | Description added for Login Module |
| 5 | 20 June 2022 | Ashwini Sudhakar Shinde | Architecture diagram added |
| 6 | 22 June 2022 | Chithra Priya G | Class diagram added |
| 7 | 24 June 2022 | Sudharsan V | UI screenshots added |
| 8 | 1 July 2022 | Aiswarya K | Sequence diagram added |
| 9 | 3 July 2022 | Chithra Priya G | Description for UI screenshots added |
| 10 | 6 July 2022 | Ashwini Sudhakar Shinde | UI Screenshots modified |
| 11 | 8 July 2022 | Mounika Puttaparthi | Class Diagram modified |
| 12 | 15 July 2022 | Aiswarya K | Sequence Diagram modified |
| 13 | 20 July 2022 | Mounika Puttaparthi | ER Diagram added |
| 14 | 23 July 2022 | Ashwini Sudhakar Shinde | References Added |
| 15 | 25 July 2022 | Sudharsan V | Description for UI module added |

Glossary

|  |  |
| --- | --- |
| Abbreviation | Description |
| BSC | Business Story Card |
| TSC | Technical Story Card |
|  |  |
|  |  |

Table of Contents

[1 Introduction 5](#_Toc379364881)

[2 Scope of Change 5](#_Toc379364882)

[3 List of impacted modules 5](#_Toc379364883)

[4 Design and Detailed technical updates 6](#_Toc379364884)

[4.1 Process model 6](#_Toc379364885)

[4.1.1 Use case Model 6](#_Toc379364886)

[4.1.2 Sequence diagram](#_Toc379364887) 8

[4.1.3 Architecture diagram](#_Toc379364887) 9

[4.1.3 Class diagram](#_Toc379364887) 11

[4.2 Proposed user Interface design 12](#_Toc379364888)

[4.2.1 Index page](#_Toc379364887) 12

[4.2.2 User registration page](#_Toc379364887) 13

[4.2.3 Login page](#_Toc379364887) 15

[4.2.5 Forgot password](#_Toc379364887) 16

[4.2.6 Home page](#_Toc379364887) 17

[4.2.7 Deposit page](#_Toc379364887) 18

[4.2.8 Withdrawal page](#_Toc379364887) 19

[4.2.9 Transfer page](#_Toc379364887) 20

[4.2.10 Balance enquiry](#_Toc379364887) 21

[4.2.11 Bank statement](#_Toc379364887) 22

[4.1.12 User profile](#_Toc379364887) 23

[4.1.13 Update mobile number](#_Toc379364887) 24

[4.3 Database design changes 25](#_Toc379364889)

[4.3.1 ER diagram](#_Toc379364887) 25

5 [References 26](#_Toc379364900)

# Introduction

This Software Application is used to access the functionality regarding banking. It is an application developed in Java. This project mainly focuses on basic operations in banking like create new bank account, make deposits, withdrawal and transfer and view transaction details.

It is an easy-to-use application and has a user-friendly interface. This bank application is used to access the account and make transfer easily without going to bank personally. The main aim of this project is develop secured and user-friendly application.

In this application, user will create their own account using their information and they will get login credentials to access banking facility. In this application we provided bank transfer, withdraw amount, balance enquiry, bank statement and view transaction details and forgot password mechanism for login credentials.

# Scope of Change

We can add different account facilities. We can add loan facilities in future. We can also include some other options like bank statement report in form of pdf.

# List of impacted modules

All the functional modules will be created from scratch.

# Design and Detailed technical updates

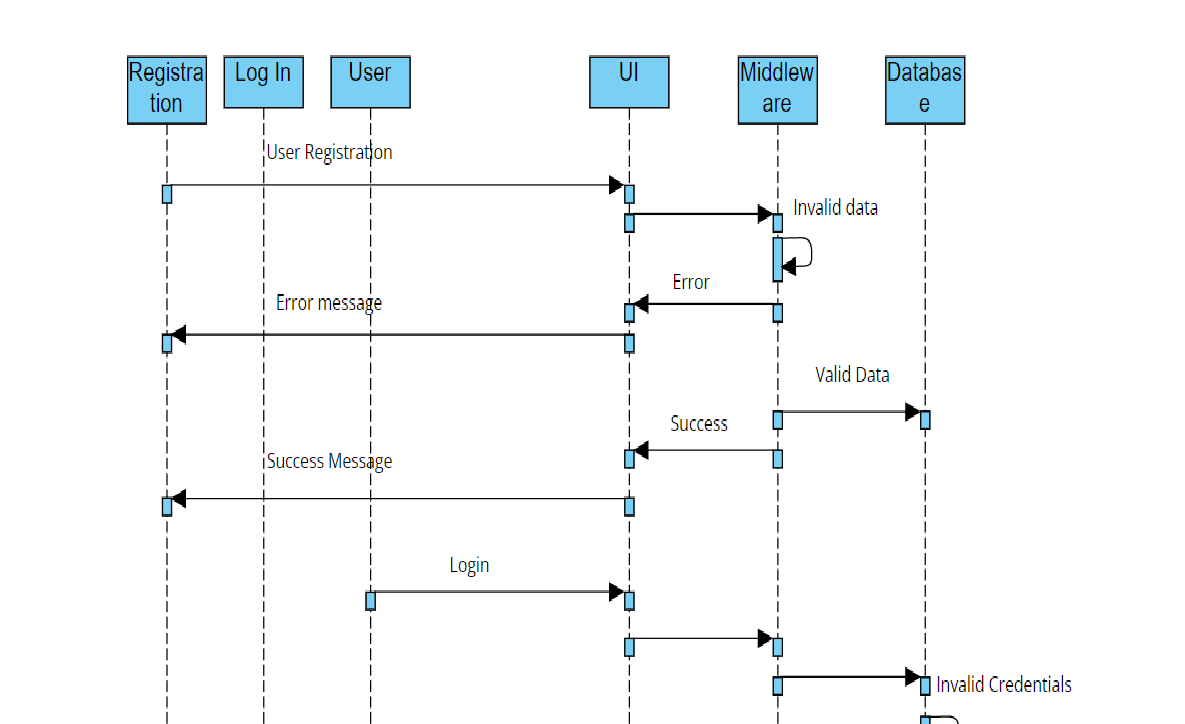
## Process model

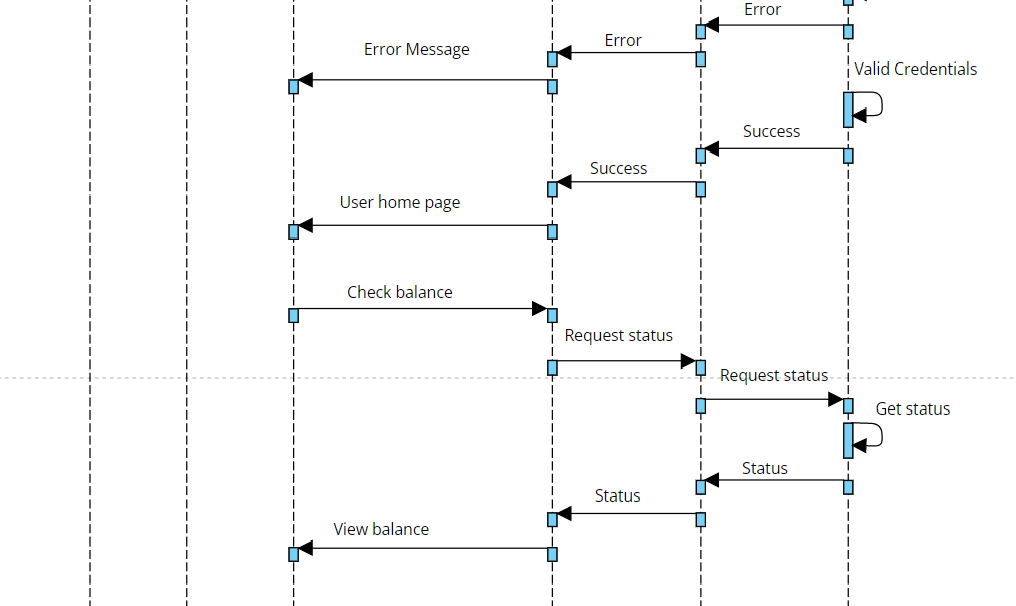
### Use case Model

### 

|  |  |
| --- | --- |
| Brief Description | User operations in Bank Application |
| Basic Flow | * New user can register into the system. After successful registration, user have to login to the system. * Existing user can login to the system and can perform: * Deposit amount by account No * Withdraw amount by account No * Transfer amount by target account No * View transaction details by userid * Check balance by giving login password |
| Alternate Flow | 1. The system will validate the information provided. If any invalid data is found, the input form will be redirected with error message. |
| Validation | 1. All the field are required 2. The Email id , Mobile number should be unique. |
| Pre-Conditions | User should have network access and Browser with latest updates. |
| Post-Conditions | Success message should be shown. |

### Sequence diagram





### Architecture Diagram

**Logical View**

**Technology/ Framework**

**Layer**

UI Components

HTML 5/CSS/

Thymeleaf

Presentation Layer

Transaction Details

Deposit

Login

Withdrawal

Web-Server Classes

Java 1.8/ Spring Boot/Spring Security

Application/ Business Layer

Controllers

Configuration

Rest Controllers

Service

Entity/ Model Classes

Spring data JPA

Data Access Layer

Repository classes

Oracle 11g Database

**Presentation Layer:**

This layer consists of all UI components.

* Login page: User can login to the system with valid username and password.
* Add new account page: User can create or register a new bank account to system with valid details.
* Deposit amount: User can deposit amount with valid account number and in the balance will be added to the user.
* Withdrawal amount: User can withdraw amount with valid account number and in the balance will be reduced to the user.
* View transaction details by ID: User can view the transaction details by user ID.

**Application Layer:**

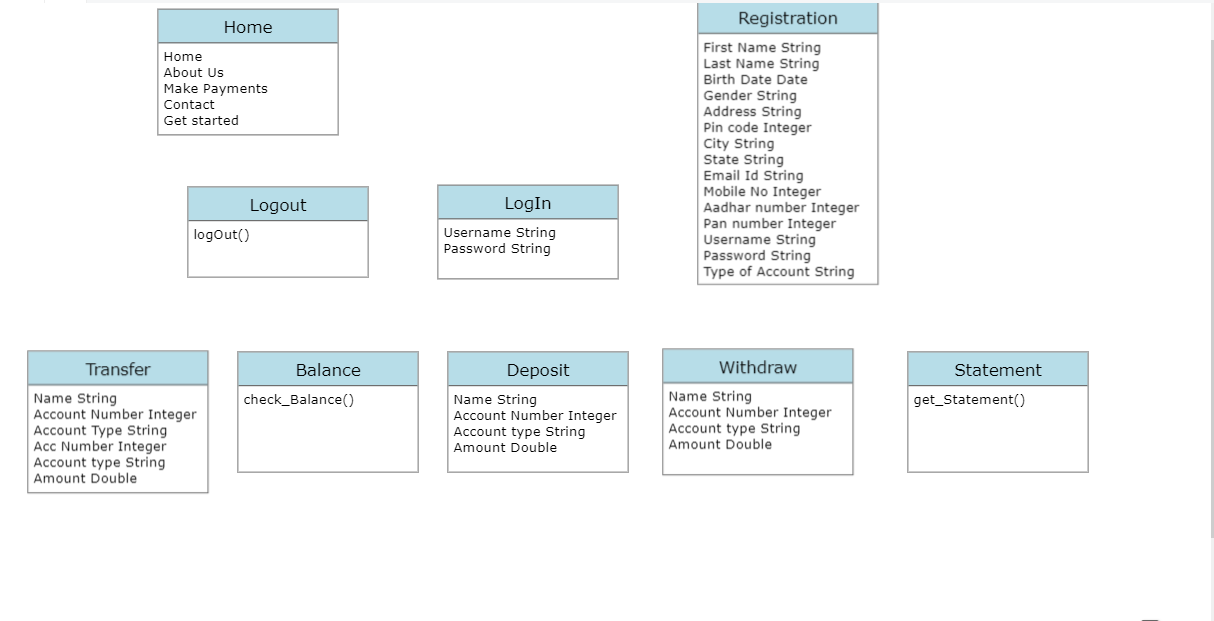
This layer comprises of all server and business classes.

* Configuration settings (XML or Class based) will define the application and server configuration.
* Spring Controllers defines the server classes, which are required to process the incoming http requests.
* Service classes are required to perform the required business services.
* REST Controllers to process the HTTP AJAX requests.
* Model classes are used the define the functions of the entities present in the system.

**Data Layer:**

Data layer is implemented through Spring data JPA. It will contain the repository classes, which provides interface to the table.

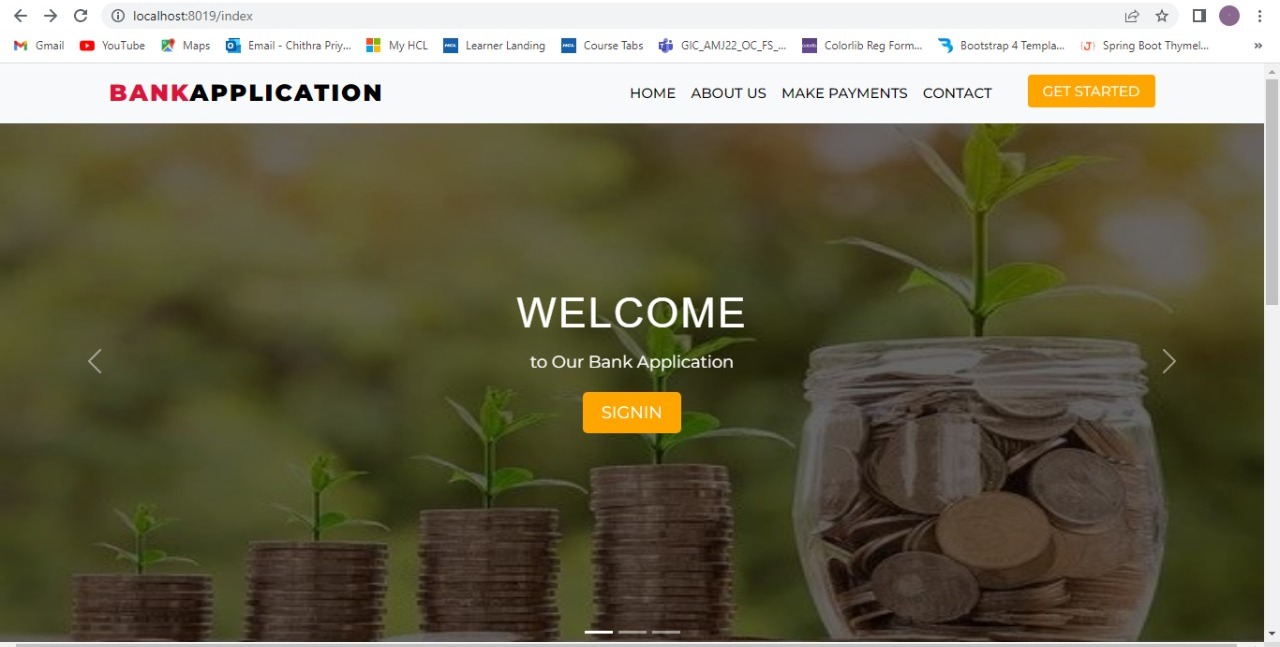
### Class Diagram



## Proposed user Interface design

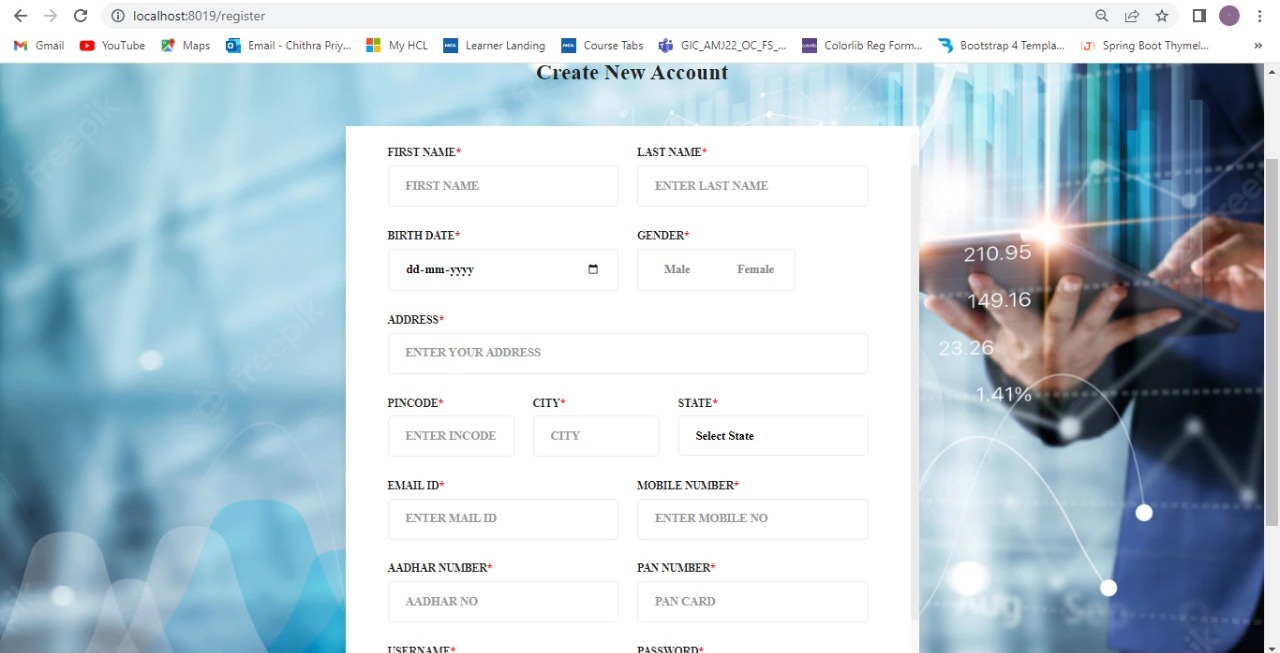
### Index Page

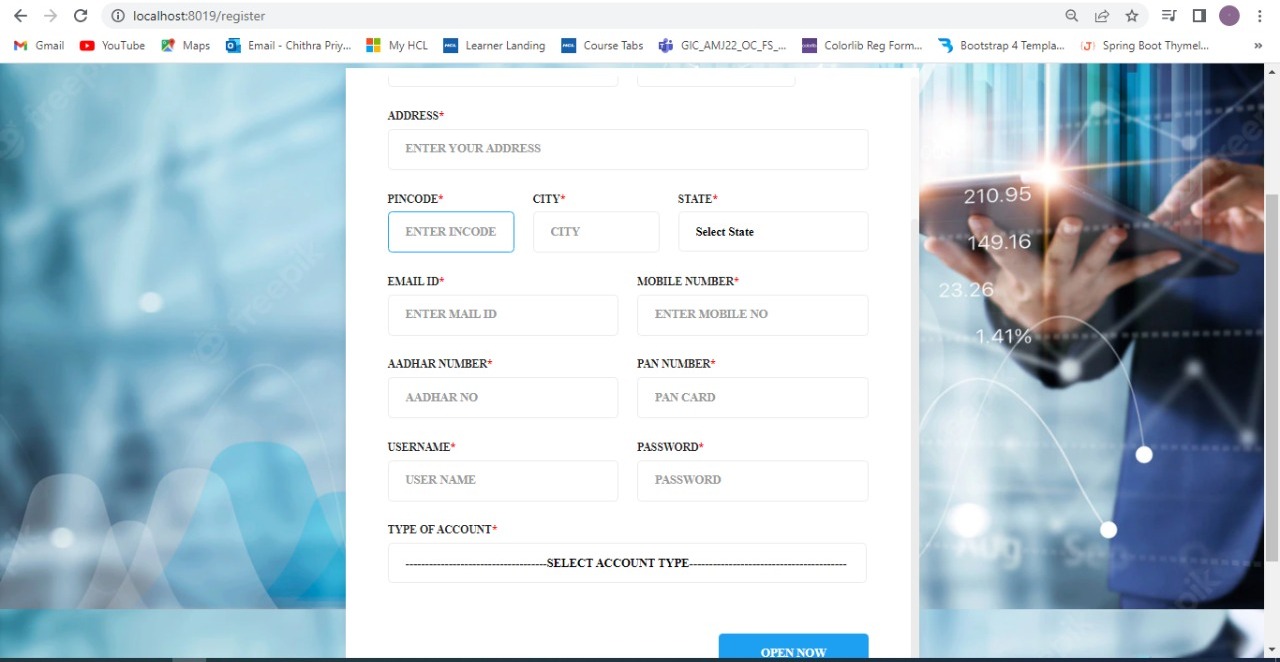
|  |  |
| --- | --- |
| Brief Description | Index Page |
| Basic Flow | This use case about the Home Page of application.   1. The following Modules are available on Home Page.  * Sign In * About Us * Make Payments * Contact |
| Pre-Conditions | User should have network access and Browser with latest updates. |
| Post-Conditions | Corresponding page has to be displayed. |



### User Registration

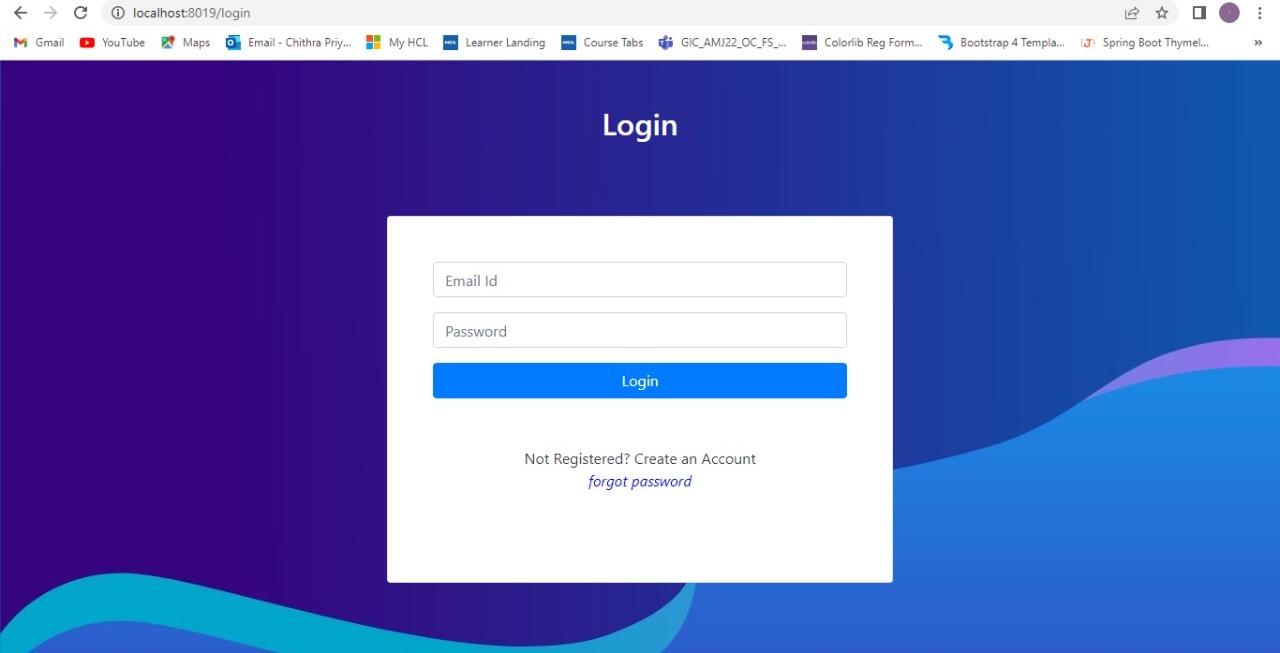
|  |  |
| --- | --- |
| Brief Description | User Registration |
| Basic Flow | This use case describes how a user register in to the system   1. The user has to register into the system. 2. After the successful registration, user will get a success message and account number will be generated automatically. 3. The following information is required during registration.  * First Name * Last Name * Birth Date * Gender * Address * Pin code * City * State * Email Id * Mobile Number * Aadhar Number * Pan Number * Username * Password * Type of Account |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the input form will be redirected with error message. |
| Validation | 1. All the field are required to fill. 2. Same Email Id can’t be used twice. 3. The Email Id, Mobile number should be unique. |
| Pre-Conditions | User should have network access and Browser with latest updates. |
| Post-Conditions | Success message should be shown. |





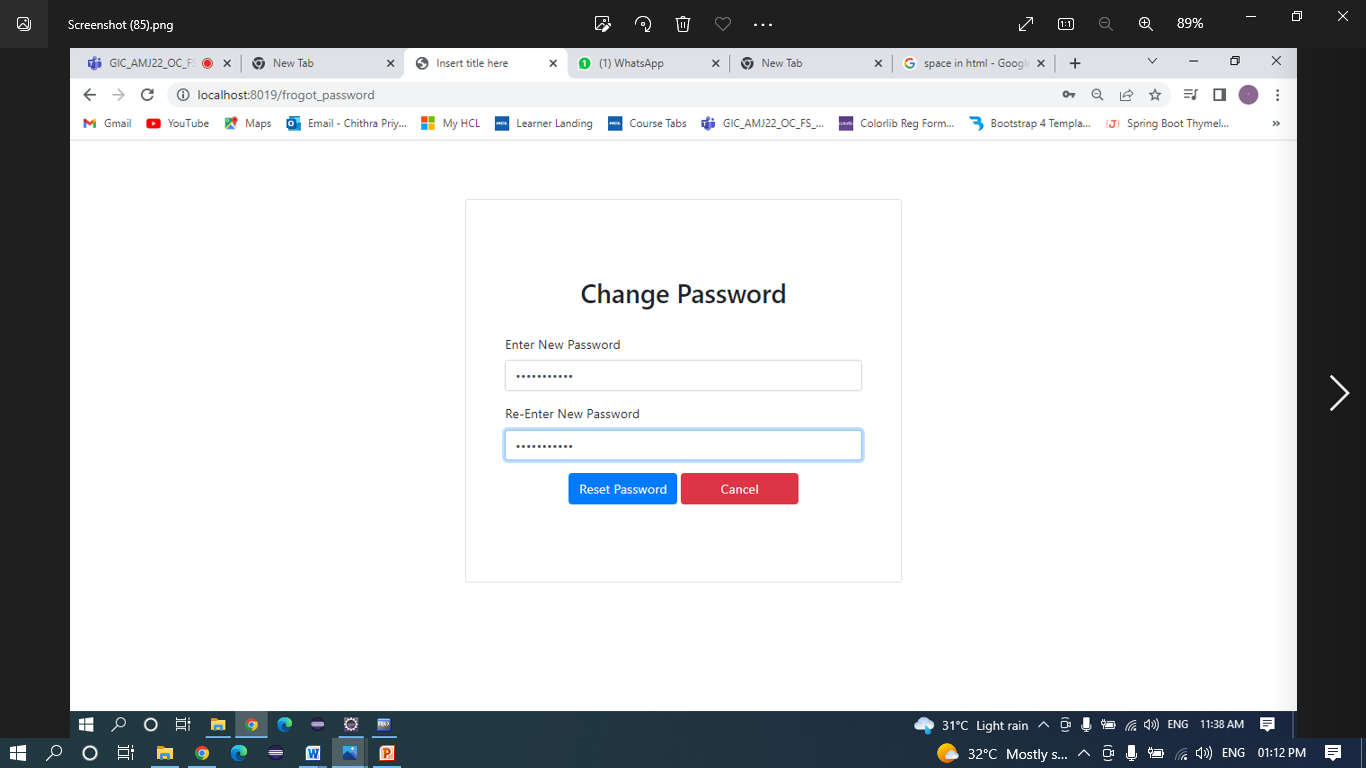
### Login page

|  |  |
| --- | --- |
| Brief Description | Login page |
| Basic Flow | This use case describes how a user log-in in to the system   1. The User has to register himself into the system. 2. After the successful login, user will be taken to the appropriate home page. 3. The following information is required to login.  * Username(username is a Mail id of user) * password |
| Alternate Flow | 1. The system will validate the credentials provided. If credentials are invalid, login form will be redirected again with error message. |
| Validation | 1. Valid username 2. Valid password |
| Pre-Conditions | User should have already registered into system. |
| Post-Conditions | home page has to be displayed. |



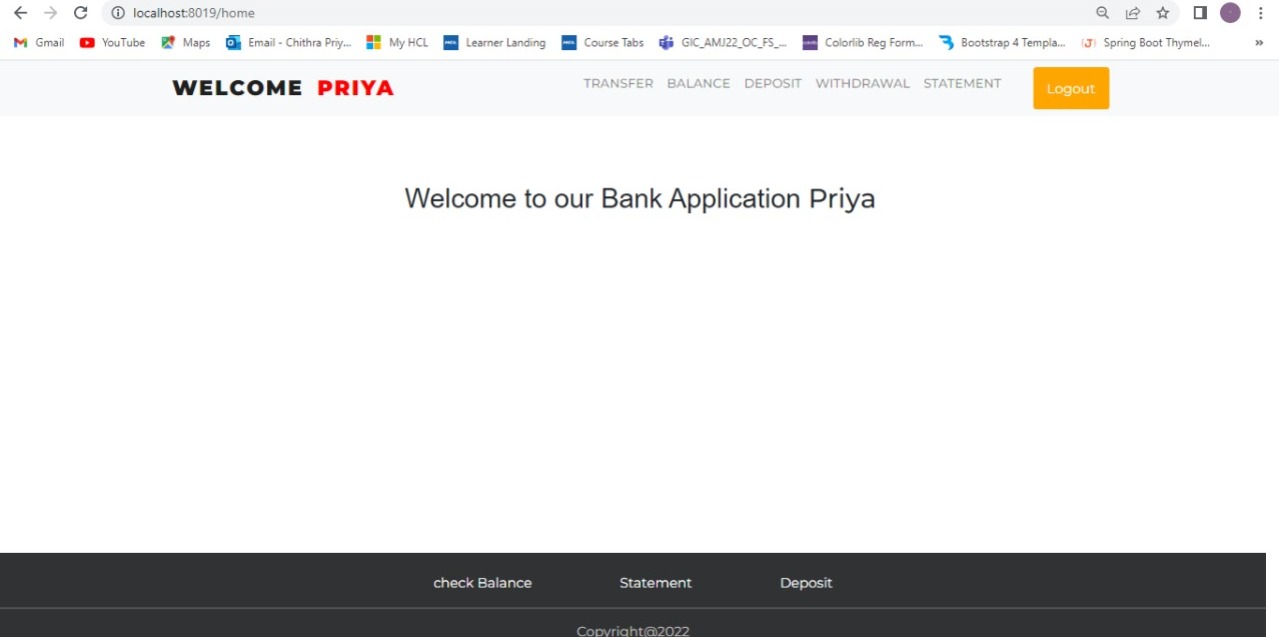
### Forgot Password page

|  |  |
| --- | --- |
| Brief Description | Login page |
| Basic Flow | This use case describes how a user can change password in the system.   1. The following information is required to login.  * New password * repeat password |
| Alternate Flow | 1. The system will validate the credentials provided. If credentials are invalid, login form will be redirected again with error message. |
| Validation | Validates both password is same. |
| Pre-Conditions | User should have already registered into system. |
| Post-Conditions | Success message will be shown. |



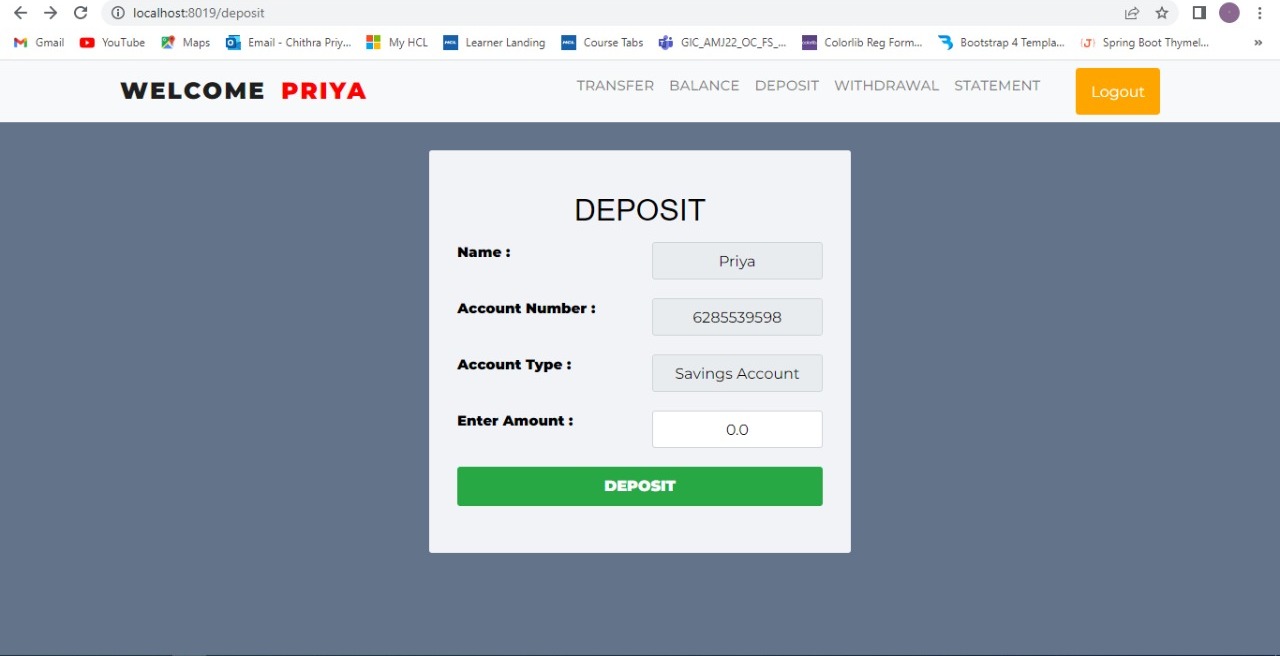
### Home Page

|  |  |
| --- | --- |
| Brief Description | Home Page |
| Basic Flow | This use case describes how User will use banking functionality.   1. The following information has to be displayed.  * Transfer * Balance * Deposit * Withdrawal * Statement |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Corresponding page has to be displayed. |



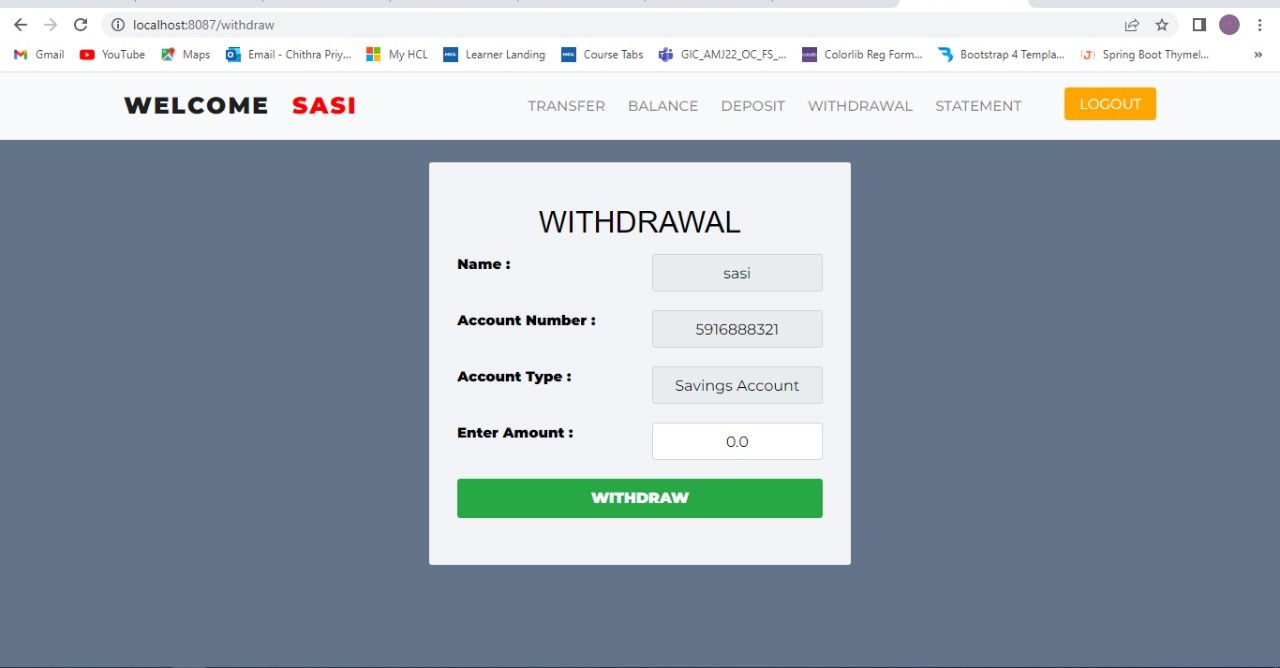
### Deposit page

|  |  |
| --- | --- |
| Brief Description | Deposit page |
| Basic Flow | This use case describes how a user deposit amount into the system   1. The following information is required during registration.  * Name * Account Number * Account Type * Enter Amount |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the input form will be redirected with error message. |
| Validation | 1. Amount should not be zero. 2. User account is saving account - amount should be <20000. 3. If user account is current account – amount should be <40000. 4. In case of saving account the maximum limit for balance should be -500000. 5. In case of current account the maximum limit for balance should be -1000000. |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Success message should be shown. |



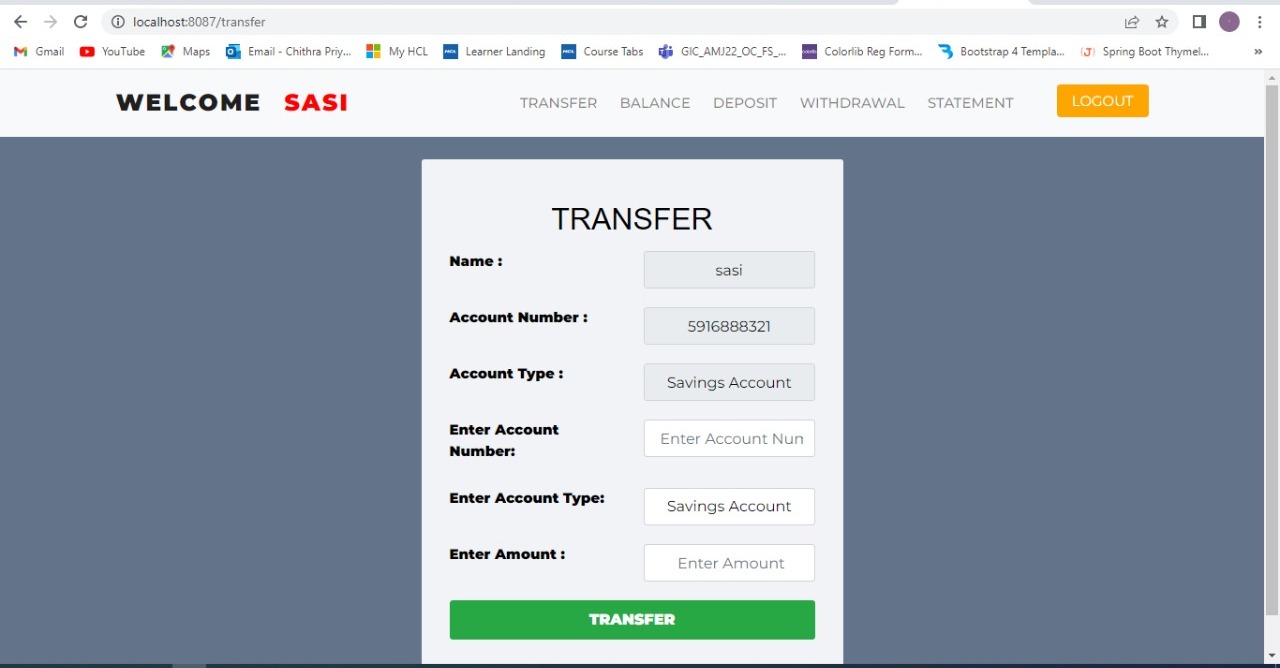
### Withdrawal page

|  |  |
| --- | --- |
| Brief Description | Withdrawal page |
| Basic Flow | This use case describes how to withdraw amount from the user account   1. The user needs to mention how much amount to be withdrawn from account. 2. After withdrawal customer will get transaction details with success message and balance of customer will be updated in database. 3. The following information is required during withdrawal.  * Name * Account Number * Account Type * Amount |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the provided information will be redirected with error message. |
| Validation | Balance should not be less than zero. |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Success message should be shown. |



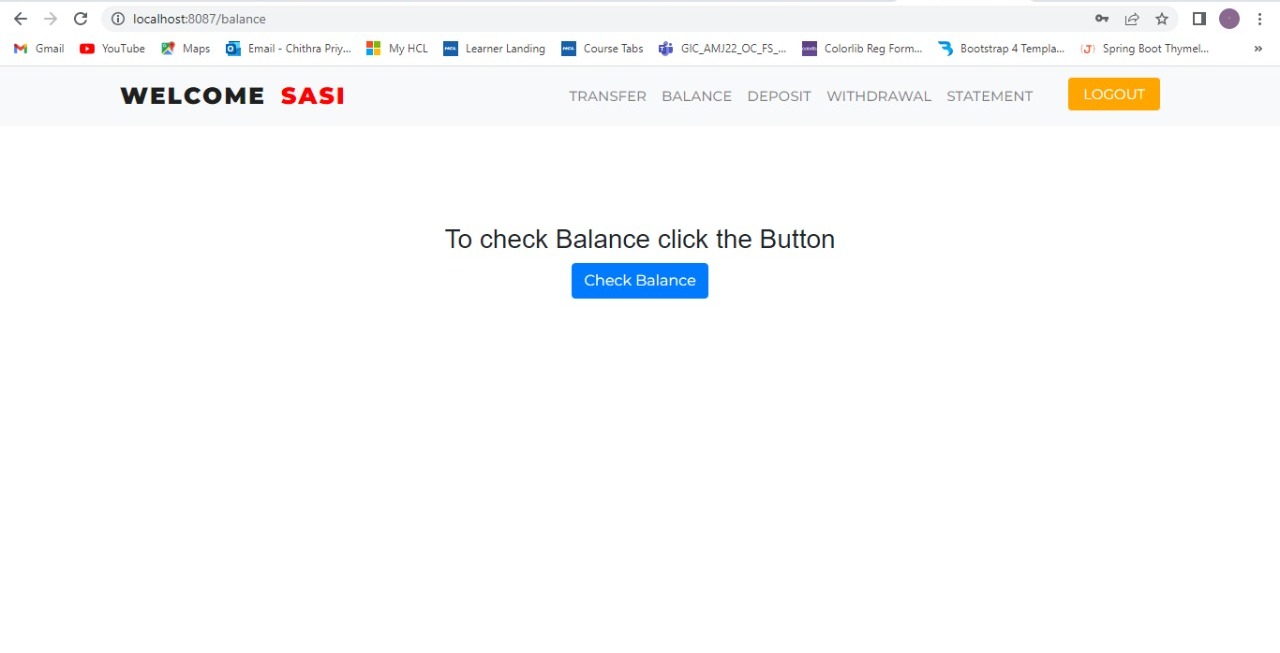
### Transfer Page

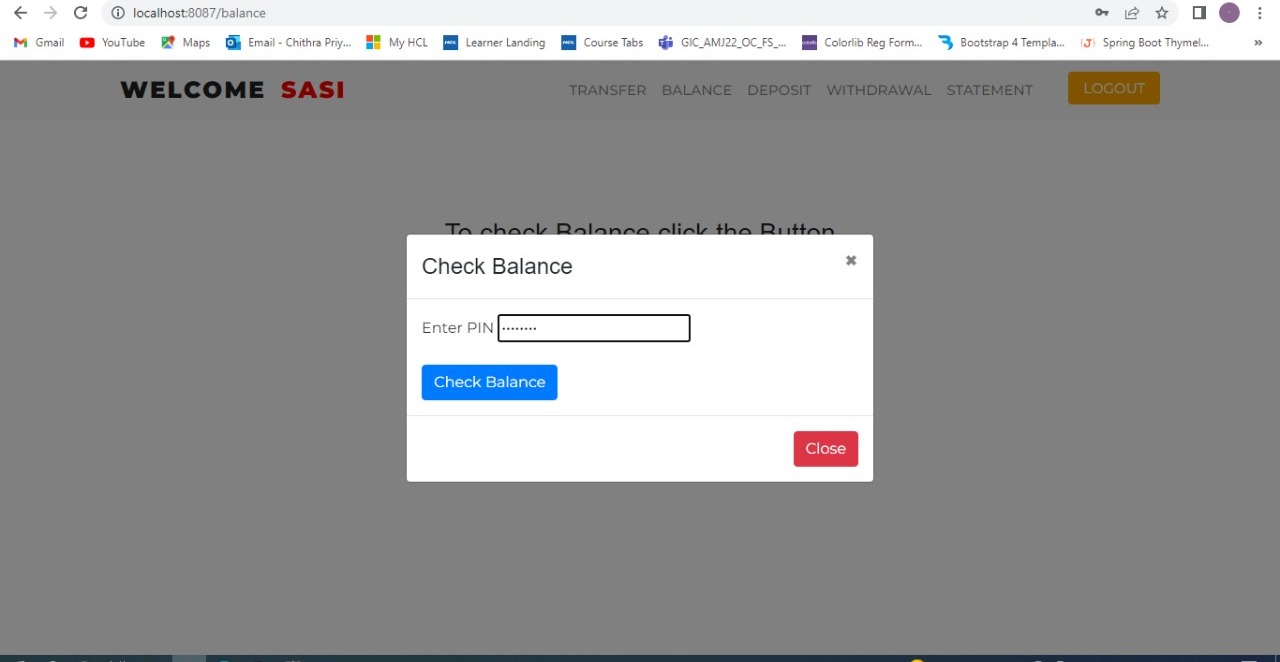
|  |  |
| --- | --- |
| Brief Description | Transfer page |
| Basic Flow | This use case describes how user can transfer amount from one account to another.  The following information is required during withdrawal.   1. Name 2. Source Account Number 3. Target Account Number 4. Source Account Type 5. Target Account Type 6. Amount   In these field only target account number, target account type and amount should be filled by user.  Other details will be fetched from the system. |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the provided information will be redirected with error message. |
| Validation | 1. Validates if target account number is already present in system. 2. Validates the amount is not equal to zero. |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Success message should be shown with transaction details in transaction\_success page. |



### Balance Enquiry

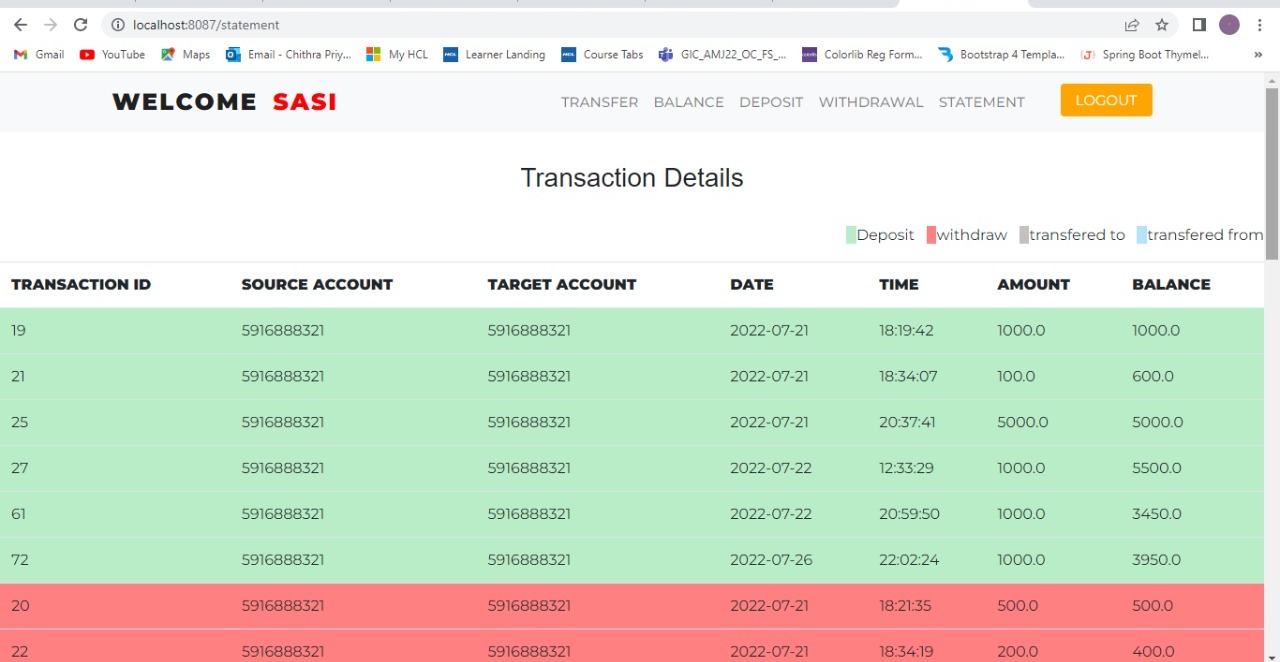
|  |  |
| --- | --- |
| Brief Description | Balance Enquiry |
| Basic Flow | This use case describes how user can check balance status.   1. The Customer has to login into the system. 2. Click on check Balance button and provide Pin (Password which is given for login credentials) to view Balance. 3. After Pin validation it will show Balance status. |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the provided information will be redirected with error message. |
| Validation | 1. Validates password is correct or not. |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Shows the Balance details status. |





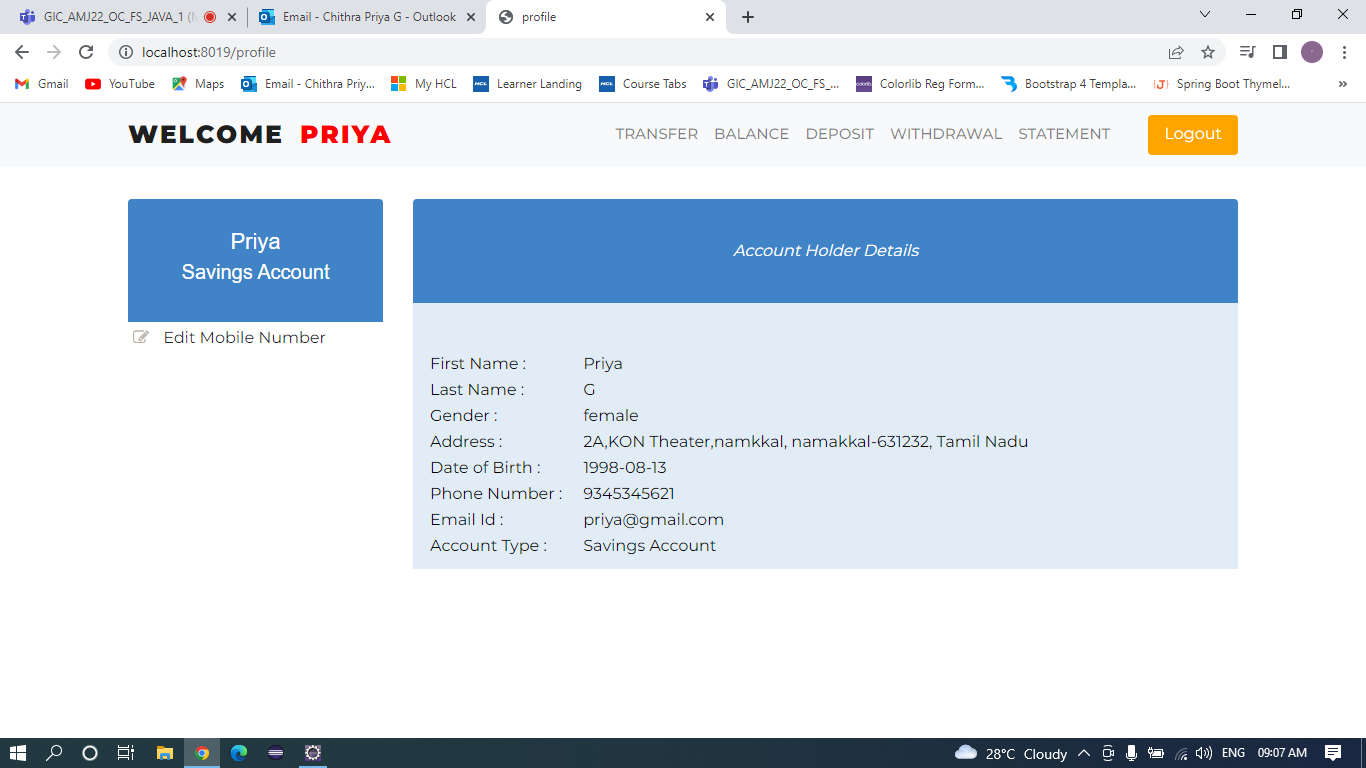
### Bank Statement

|  |  |
| --- | --- |
| Brief Description | Bank Statement |
| Basic Flow | This use case describes how a user will get bank statement of their transactions details from the system   1. Transaction details is fetched by user ID 2. Displayed in form of table, which represents the different types in different colors. They are:  * Deposit - light green * Withdraw - light red * Transferred to – grey * Transferred from – sky blue |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the provided information will be redirected with error message. |
| Validation | Fetched transaction details by user ID. |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Landing page has to be displayed. |



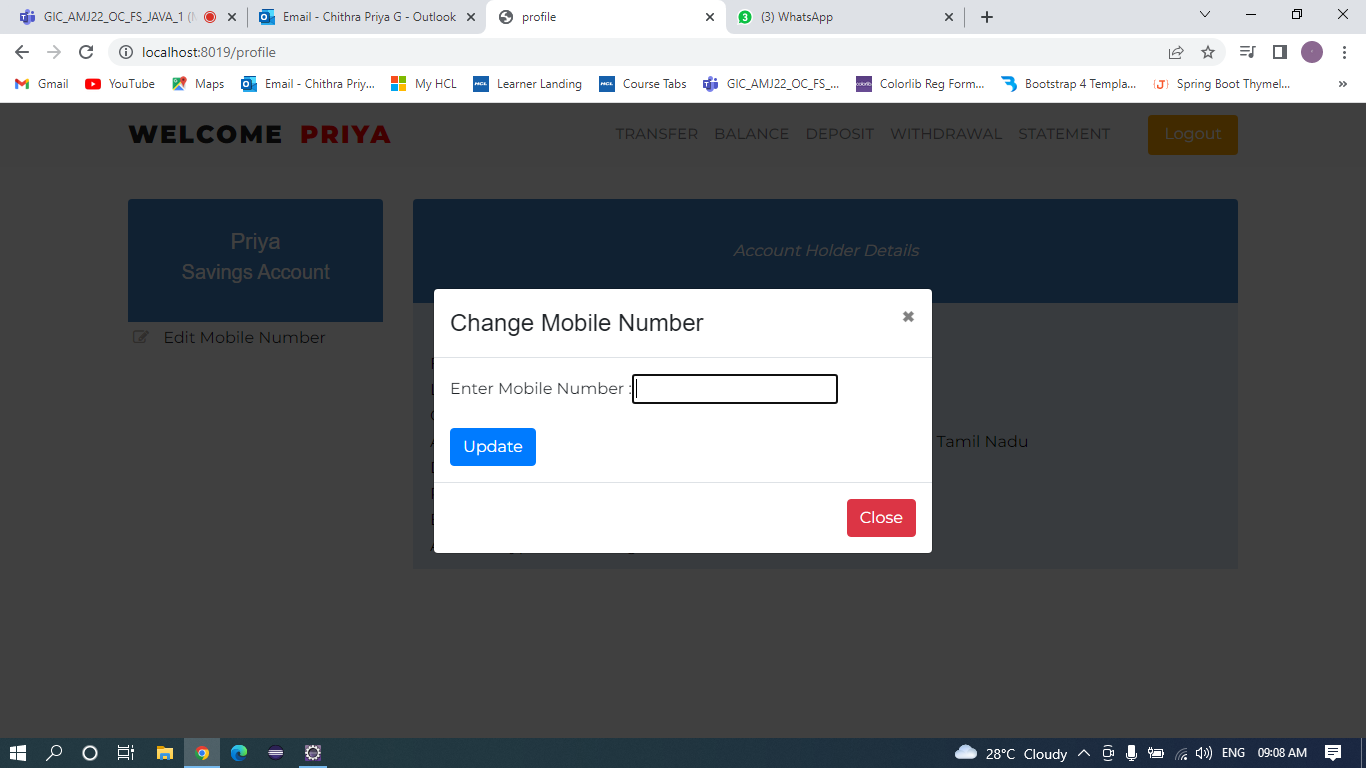
### User Profile

|  |  |
| --- | --- |
| Brief Description | User Profile |
| Basic Flow | This use case describes how a user can see User Details.  Here user can see user details like,   * First name * Last name * Mobile number * Email Id * Address * Type of account |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the provided information will be redirected with error message. |
| Validation | Fetched user details by user ID. |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Landing page has to be displayed. |



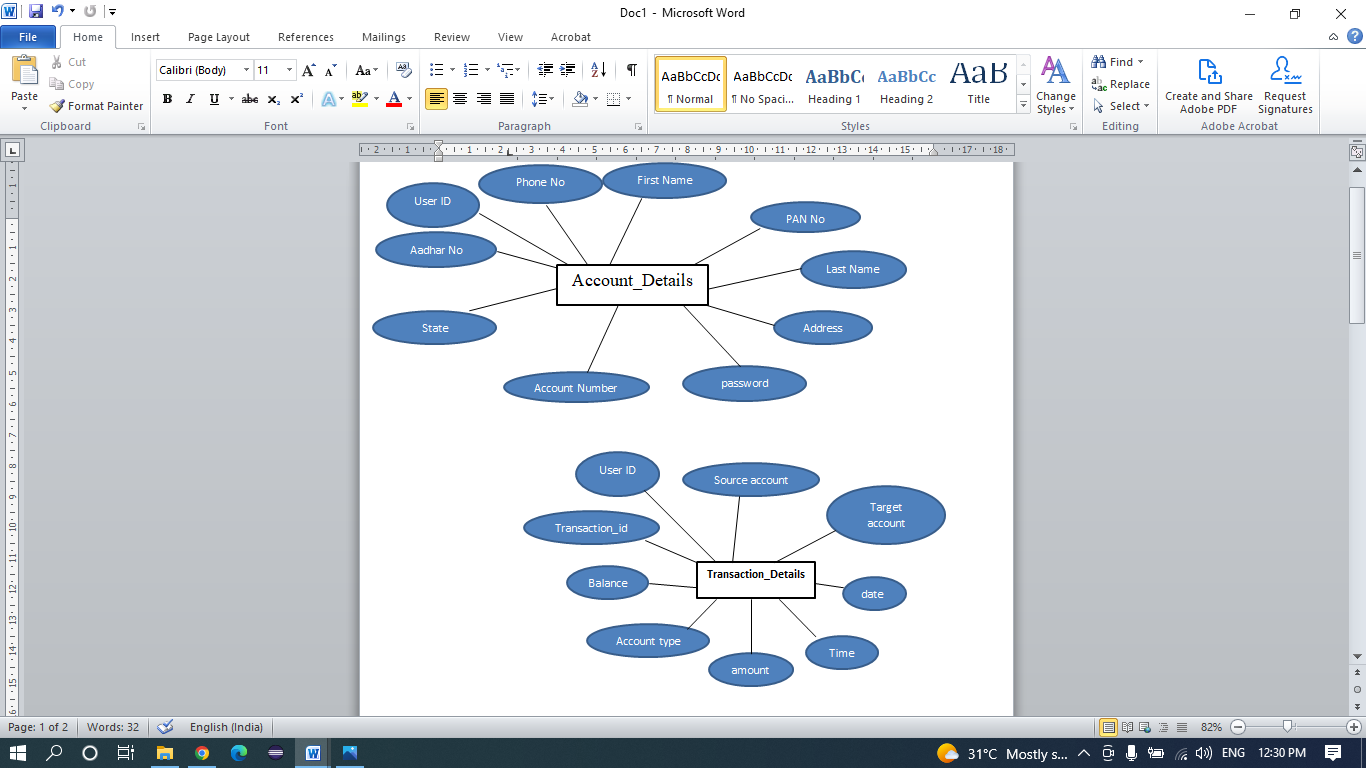
### Update Mobile Number

|  |  |
| --- | --- |
| Brief Description | Update Mobile Number |
| Basic Flow | This use case describes how a user can see change mobile number.  The following information is required during updating mobile number   * Mobile number |
| Alternate Flow | The system will validate the information provided. If any invalid data is found, the provided information will be redirected with error message. |
| Validation | mobile number should have max-length-10 |
| Pre-Conditions | User should have already registered and logged into the system. |
| Post-Conditions | Success message will be shown |



## Database Design Change

### ER Diagram



# References

We as a team have used references to LMS content to complete our project of “Bank Application”. HCL provide the three months training for us. On that particular training we learnt so many technologies like core java, HTML, CSS, spring boot. What all we learnt in the training we implemented in our project.