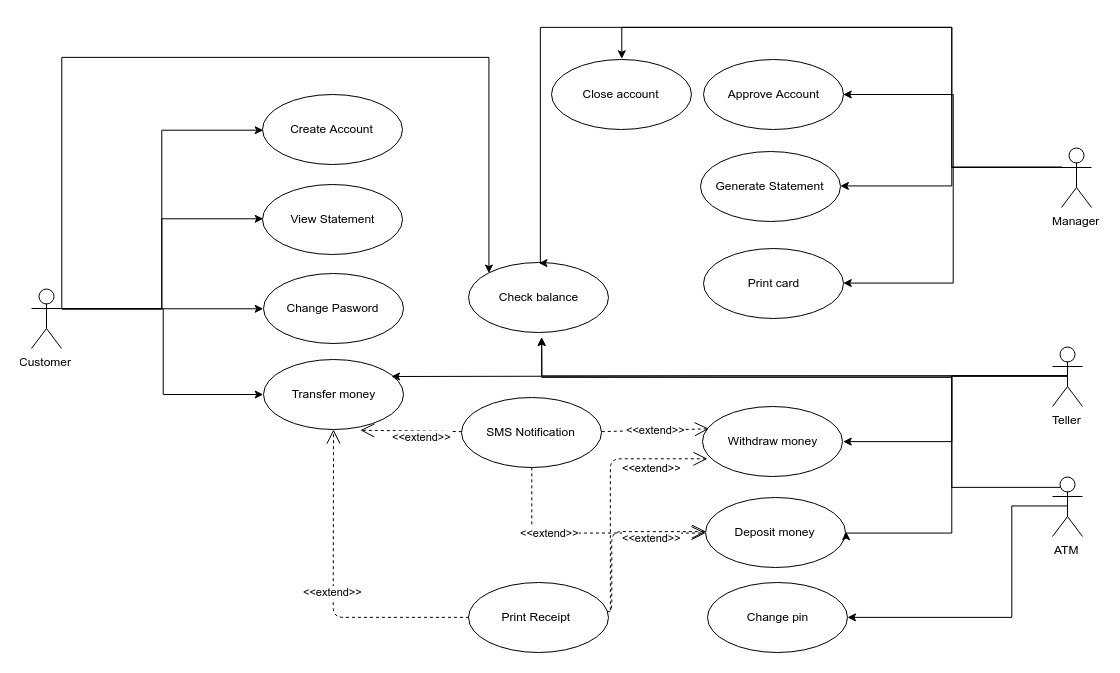
**Smart Banking System SRS Document**

1. **Smart Banking system repository on Github**

<https://github.com/daveabipomy/SmartBankingSystem/tree/develop>

1. **Use Case Diagram for Smart banking system**
2. **Smart Banking System: Use Case Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case Number: 1** | | | |
| **Name** | | Teller CRUD | |
| **Brief description** | | This use case allows manager to create tellers. | |
| **Actors** | | Manager | |
|  | | | |
| **Preconditions: manager must have logged in to the system** | | | |
|  | | | |
| **Flows of events** | | | |
| 1. **Basic Flows** | | | |
| **1.1.0 Add Teller** | | | |
| Step | User Actions | | System Actions |
| 1 | The manager calls the create teller command. | | The system displays the teller creation form. |
| 2 | The manager fills out the form and request the system to save the details. | | The system verifies if there is no other teller with same social security number and returns a success message or a fail message in case of an exception. It should return a specific message when a duplicate teller exists |
| **Postconditions:** The teller is persisted in the system | | | |
|  | | | |
| **Business Rules:** Two tellers can’t have same social security number | | | |
|  | | | |
| **1.1.1 Read Teller** | | | |
| **step** | **User Actions** | | **System Action** |
| 1 | The manager selects to view a list of tellers | | The system returns a list of all tellers. |
| 2 | The manager selects to view a teller of one of the tellers from the list of tellers | | The system returns the information of the teller. |
| **1.1.1 Update Teller** | | | |
| **step** | **User Actions** | | **System Action** |
| 1 | The manager selects to view a list of tellers | | The system returns a list of all tellers. |
| 2 | The manager selects the teller he wants to update | | The system displays an editable teller form pre-populated with the teller details. |
| 3 | The manager updates the data of the teller they want to update and requests system to save the updated data. | | The system updates the record and returns the success message or a fail message on exception |
|  | | | |
| **Postconditions:** The teller details will be updated. | | | |
|  | | | |
| **1.1.3 Delete Teller** | | | |
| **step** | **User Actions** | | **System Action** |
| 1 | The manager selects to view a list of tellers | | The system returns a list of all tellers. |
| 2 | The manager selects the teller he wants to delete | | The system displays a confirmation dialogue window |
| 3 | The manager selects OK on the confirmation dialog window to confirm deleting the profile | | The system deletes the teller and returns message a success confirms. |
|  | | | |
| **Postconditions:** A teller will be deleted | | | |
|  |  | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 2** | | | |
| **Name:** | | Manager Operations | |
| **Brief description:** | | This use case provides manager functionality | |
| **Actors:** | | Manager | |
|  | | | |
| **Preconditions:** The Manager must logged in to the system | | | |
|  | | | |
| **Flows of Events:** | | | |
| **1. Basic Flows** | | | |
| **1.1.0 Approve Account** | | | |
| **Step** | **User Action** | | **System Action** |
| **1** | The Manager call view list of new customers waiting for approval | | The system displays list of new customers that are waiting for the manager’s approval |
| **2** | The Manger selects the customer and approve the account creation | | The system create a new account for the approved customer. |
| **Post-Condition:** The new customer account is added to the system database. | | | |
|  | | | |
| **Business Rules** | | | |
| No duplicate customer accounts. A unique account is identified by account number. | | | |
|  | | | |
| **1.1.1 Close Account** | | | |
| **Step** | **User Action** | | **System Action** |
| **1** | The Manager call view list of customers command | | The system displays list of customers. |
| **2** | The Manager selects the account which is needing to be closed | | The system deletes the selected account from the database. |
| **Post-Condition :** Customer account is deleted from the database. | | | |
|  | | | |
| 1.1.2 Generate Statement | | | |
| **Step** | **User Action** | | **System Action** |
| **1** | The Manager call generate customer reports command | | The system displays the field for statement generation |
| **2** | The Manager fills the required information for statement generation | | The system will generate customer’s statement. |
| **Post-Condition** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case Number: 3** | | | |
| **Name** | | Create account | |
| **Brief description** | | This use case allows a customer to create an account which needs to be approved by the manager. | |
| **Actors** | | customer | |
|  | | | |
| **Preconditions:** Have a valid Social security number | | | |
|  | | | |
| **Flows of events** | | | |
| 1. **Basic Flows** | | | |
| **1.1.0 Create Account** | | | |
| Step | User Actions | | System Actions |
| 1 | The customer calls the create account command. | | The system displays the customer creation forms. |
| 2 | The customer fills out the form and request the system to send the details for manager approval. | | The system verifies if there is no other customer with this national ID/Passport number, name holding chosen account type in the database. The system generates a successful message on success or a fail message in case of failure. |
| **Postconditions:** The opened account is sent for approval to the manager. | | | |
|  | | | |
| **Business Rules:** | | | |
| No duplicate account type for one customer | | | |
| **1.1.1 view Account** | | | |
| **step** | **User Actions** | | **System Action** |
| 1 | The user clicks on view command | | The system returns a list of all tellers. |
| **1.1.2 Update Teller** | | | |
| **step** | **User Actions** | | **System Action** |
| 1 | The customer clicks on update | | The system displays an editable form pre-populated with his data |
| 2 | The customer updates his data | | The system updates the record and returns the success message or a fail message on exception |
|  | | | |
| **Postconditions:** The customer details will be updated. | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case Number: 4** | | | |
| **Name** | | Check Balance | |
| **Brief description** | | This use case allows:  -customer to check his balance online or through ATM  -Teller to check balance for a specific customer | |
| **Actors** | | Customer, Teller, ATM | |
|  | | | |
| **Preconditions: The customer (via web app or ATM) and teller must have logged using their credentials.** | | | |
|  | | | |
| **Flows of events** | | | |
| 1. **Basic Flows** | | | |
| **1.1.0 Check balance** | | | |
| Step | User Actions | | System Actions |
| 1 | The customer/ teller /ATM calls the check balance command. | | The system displays all accounts linked to the customer. |
| 2 | The customer / teller/ATM chooses the account. | | The system displays the current balance of the chosen account. |
| **Postconditions:** | | | |
|  | | | |
| **Business Rules:** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case Number: 5** | | | |
| **Name** | | Deposit Money | |
| **Brief description** | | This use case allows the tellers and customers through ATM machine to deposit money in the system | |
| **Actors** | | Teller, ATM machine | |
|  | | | |
| **Preconditions: Teller and customer through ATM machine must have logged in to the system** | | | |
|  | | | |
| **Flows of events** | | | |
| 1. **Basic Flows** | | | |
| **1.1.0 Deposit Money** | | | |
| Step | User Actions | | System Actions |
| 1 | The Teller and ATM machine calls the deposit money command. | | The system asks to which account type the deposit will be made. |
| 2 | The teller and customer select the account type from the list | | The system displays the deposit money form with the field the amount of money that is going to be deposited |
| 3. | The teller and customer fill the amount of money to be deposited | | The system deposits the money to the current balance in the account and returns confirmation message. |
|  |  | |  |
| **Postconditions:** The deposit amount of money is added to the current balance and persisted in the system | | | |
|  | | | |
| **Business Rules:** | | | |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use case Number: 6** | | | |
| **Name** | | Transfer Money | |
| **Brief description** | | This use case allows the teller and customers through ATM machine to transfer money in the system | |
| **Actors** | | Teller, ATM machine | |
|  | | | |
| **Preconditions: Teller and customer through ATM machine must have logged in to the system** | | | |
|  | | | |
| **Flows of events** | | | |
| 1. **Basic Flows** | | | |
| **1.1.0 Deposit Money** | | | |
| Step | User Actions | | System Actions |
| 1 | The Teller and ATM machine calls the transfer money command. | | The system asks to which account type the deposit will be made. |
| 2 | The teller and customer select the account type from the list | | The system displays the transfer money form with the field account id to which the money is going to transfer |
| 3. | The teller and customer fill the account id | | The system requests the amount of money to be transferred |
| 4. | The teller and customer fill the amount of money | | The system verifies that the customer has that amount of money in the account and transfer the money and returns confirmation message |
|  |  | |  |
| **Postconditions:** The transfer amount of money is deducted from the customer account and added to the transferred account | | | |
|  | | | |
| **Business Rules: Customer must have sufficient balance in his account** | | | |
|  | | | |

1. **Supplementary Specification**

The supplementary specifications cover all the nonfunctional requirements of the system that are not easily defined in the use case Model. It gives us the idea about the nonfunctional requirements and features such as Usability Reliability performance and all other features.

**Reliability**

The main system must be running 97% of the time.

**Performance**

The system is compatible with the os. All the tasks should take less than 3 seconds to execute. The system shall support up to 100 simultaneous users against the central database at any given time.

**Usability**

Smart banking system is a user-friendly system. In order to use the system, the user don’t need any kind of training. The software has a few buttons and the user have to press those buttons to achieve any functionality of the system. And the user also can understand the user manual easily.

1. **Glossary**

**Checking Account** - bank account for everyday expenses

**Saving Account** - bank account where you can store money you don't need right away but still keep it easily accessible, all while earning interest

**Credit Account** - an account which a customer has with a shop which allows him or her to buy goods and pay for them later

**Withdraw**- the removal of money from your account at a bank

**Deposit**- putting money into a bank account.

**Transfer**- transferring money from one account to another account

**ATM machine-** an electronic banking outlet that allows customers to complete basic transactions without the aid of teller

**Balance**- the amount of money in the account

**Pin**- Personal Identification Number, allows you to access your account information

**Statement**- monthly document that summarizes your activity to show what money went in and out of a bank account.