**Maharishi International Bank (MIB): Use Case Description**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case Number: 1** | | | | | |
| **Name User Profile CRUD** | | | | | |
| Brief description This use case allows the admin to create profiles for teller and other admin | | | | | |
| **Actors Admin** | | | | | |
|  | | | | | |
| **Preconditions** | | | | | |
| The admin must log in to the system | | | | | |
| **Flows of Events:** | | | | | |
| 1. Basic Flows | | | | | |
| 1.1.0 Create User Profile | | | | | |
| Step | | | User Actions | System Actions | |
| 1 | | | The admin calls the create user profile command | The system displays the user profile form with the fields for username, user type, status and password. | |
| 2 | | | The Admin fills out the form and requests the system to save the details | The system verifies that there’s no other profile in the database with the same username and saves the user and returns the success message on success or a fail message in case of failure. In case another profile exists with the username, the system returns the message indicating a duplicate entry exists. | |
| **Postconditions** | | | | | |
| The user profile is persisted in the system | | | | | |
|  | | | | | |
| **Business Rules** | | | | | |
| No duplicate user profiles. A unique profile identified by username. | | | | | |
| All fields could not be empty. | | | | | |
|  | | | | | |
| **1.1.1 Read/View User Profile** | | | | | |
| **Step** | | **User Action** | | **System Actions** | |
| 1 | | The admin selects to view a list of user profiles. | | The system returns a list of all user profiles. | |
| 2 | | The admin selects to view a profile of one of the users from the list of profiles. | | The system returns the profile of the user as a string with the username, status and user type. | |
|  | | | | | |
| **1.1.2 Update Faculty Profile** | | | | | |
| **Step** | **User Actions** | | | **System Action** | |
| 1 | The admin selects to view a list of user profiles | | | The system returns a list of all user profiles. | |
| 2 | The admin selects the profile of the user they want to update | | | The system displays an editable user profile form pre-populated with the user profile details | |
| 3 | The admin updates the fields they want to update and requests system to save the new details | | | The system updates the record and returns the success message or a fail message on exception. | |
| **Post Condition** | | | | | |
| The user profile will be updated | | | | | |
| **Business Rule** | | | | | |
|  | | | | | |
|  | | | | | |
| **1.1.3 Update User Profile** | | | | | |
| **Step** | | **User Actions** | | | **System Actions** |
| 1 | | The admin selects to view a list of user profiles | | | The system returns a list of all user profiles. |
| 2 | | The admin selects to delete a user profile from a list of user profiles | | | The system displays a confirmation dialogue window. |
| 3 | | The admin selects OK on the confirmation dialog window to confirm deleting the profile | | | The system confirms the user is not active and deletes the profile. The system returns message a success message on success or a failure message. The system should return a message indicating the profile could not be deleted because it is still active. |
| **Post Condition** | | | | | |
| The User profile will be deleted | | | | | |
| **Business Rule** | | | | | |
| Only user that is inactive. | | | | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 2** | | |
| **Name Bank Account CRUD** | | |
| Brief description. This use case provides functionality on creating customer account | | |
| **Actors Teller** | | |
|  | | |
| **Preconditions** | | |
| The user must be logged in to the system | | |
| **Flows of Events:** | | |
| **2. Basic Flows** | | |
| **2.1.0 Create Account** | | |
| Step | User Actions | System Actions |
| 1. | User puts in customer information and assigns a unique account number | System checks if account number already exists, returns success message on successful creation of account on database or a fail message if account number already |
| **Postconditions** | | |
| Account is created on database | | |
| **Business Rules** | | |
| The account number is unique. | | |
| **2.1.1 Account status change** | | |
| Step | User Actions | System Actions |
| 1. | User puts in customer account information | System checks if account information exists and returns details of account |
| 2. | User selects desired account status “Active or Inactive” | System updates account status |
| **Postconditions** | | |
| Account status is updated on database | | |
| **Business Rules** | | |
| Account must be selected | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 3** | | | |
| **Name** Bank Account CRUD | | | |
| Brief description: This use case provides functionality on customer account operations | | | |
| Actors: Customer/Teller | | | |
|  | | | |
| **Preconditions** | | | |
| User must be logged in to an account | | | |
| **Flows of Events:** | | | |
| 1. Basic Flows | | | |
| **3.1.0 Withdraw Money** | | | |
| Step | User Actions | System Actions | |
| 1 | User selects which account to withdraw money from | System checks if the selected account can be withdrawn from | |
| 2 | User inputs amount of money to withdraw | System checks if the account has enough fund to process the transaction and dispense the money. | |
| **Postconditions** | | | |
| 1. Account balance is updated according to the amount withdrawn | | | |
| 1. The withdrawal information will be persisted in the database | | | |
| **Business Rules** | | | |
| 1. The account must have sufficient balance | | | |
| **3.1.1 Deposit Money** | | | |
| Step | User Actions | | System Actions |
| 1. | User puts in account number for the account where money is to be deposited | | System checks if account exists and returns account name on successful validation or a fail message if account does not exist |
| 2. | User inputs amount of money to be deposited | | System updates the balance of the account selected |
| **Postconditions** | | | |
| Account balance is updated according to amount deposited | | | |
| The deposit information will be persisted in the database | | | |
| **Business Rules** | | | |
| The account to be credited must be in the database | | | |
| Account status can be active or Inactive | | | |

|  |  |  |
| --- | --- | --- |
| **3.1.2 Transfer Money** | | |
| Step | User Actions | System Actions |
|  |  |  |
| 1 | The user enters the account number of the target customer | The system checks if the account number exists returns and displays a field with a message “enter the amount to transfer” or fail message if the account does not exist |
| 2 | The user enters the amount hit a transfer command | The system verifies if the The system responds the money has been transferred |
| **Postconditions** | | |
| The transfer information will be persisted in the database | | |
|  | | |
| **Business Rules** | | |
| 1.The account must have sufficient balance | | |
| 2.The account field must not be blank | | |
|  | | |
| Alternative path | | |
| Shows a message “Inefficient balance in account,please enter smaller amount” if the account has insufficient balance | | |
| Shows a message “the field must not be blank” if the account field is blank | | |

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 4** | | |
| **Name: Utility Payment CRUD** | | |
| Brief description: This use case provides functionality to make utility payments | | |
| **Actors: Customer** | | |
|  | | |
| **Preconditions** | | |
| User must be logged in to an account | | |
| **Flows of Events:** | | |
| 1. Basic Flows | | |
| **4.1.0 Utility payment** | | |
| Step | User Actions | System Actions |
| 1 | User selects the entities available to make payment for | System displays amount input and date for payment processing blocks |
| 2 | User enters payment amount and date that the fee is going to be processed on | System checks if the user account has sufficient fund to process the payment and transfer to the entity |
| **Postconditions** | | |
| 1. Account balance is updated according to the amount withdrawn | | |
| 1. The transaction information will be persisted in the database | | |
| **Business Rules** | | |
| The account must have sufficient balance | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Number: 5** | | | |
| **Name Loan Application CRUD** | | | |
| Brief description This use case allows the Teller to create and view Loan Application for Student | | | |
| **Actors Teller** | | | |
|  | | | |
| **Preconditions** | | | |
| The teller must log in to the system | | | |
| **Flows of Events:** | | | |
| 1. Basic Flows | | | |
| **5.1.0 Create Loan Application** | | | |
| Step | | User Actions | System Actions |
| 1 | | The teller calls the create loan application command | The system displays the loan application form with the fields for loan\_number which is auto generate, studentid, loan\_amount, interest\_rate, loan\_date, status. |
| 2 | | The teller fills out the form and requests the system to generate the schedule | The system generates the schedule as payment date, principle, interest, outstanding balance, is paid and display to the teller. |
| 3 | | The teller verifies the schedule and request the system to save the information | The system verifies that there’s no loan application with the active status associate with the studentid in the database and saves the user and returns the success message on success or a fail message in case of failure. In case studentid associate with the active loan status, the system returns the error message indicating that could not create loan application that has the previous application which haven’t inactive . |
|  | |  |  |
| **Postconditions** | | | |
| The loan application and loan schedule is persisted in the system. | | | |
|  | | | |
| **Business Rules** | | | |
| No more than one active loan application at the same time. Use LoanNumber as a unique identifier. | | | |
| All fields could not be empty. | | | |
|  | | | |
| **5.1.1 Read/View Loan Application** | | | |
| **Step** | **User Action** | | **System Actions** |
| 1 | The teller selects to view a list of loan application. | | The system returns a list of all loan application. |
| 2 | The teller selects to view a loan application of applications from the list of applications. | | The system returns the loan application info and schedule. |

|  |  |  |
| --- | --- | --- |
| **Use Case Number: 6** | | |
| **Name: Interest Rate crud** | | |
| admin creates the interest rates | | |
| **Actors :Admin** | | |
|  | | |
| **Preconditions** | | |
| The admin must be logged in to the system | | |
| **Flows of Events:** | | |
| 1. Basic Flows | | |
| **6.1.0 Create interest rate** | | |
| Step | User Actions | System Actions |
| 1 | The user calls the create interest rate command | The system displays a screen with the loan interest and saving interest text fields |
| 2 | The user inserts the values in the text fields | The system checks if there are values in database and if there is a value ,the system will update the values otherwise it will be created and the system responds the opreration is successful |
|  |  |  |
| **Post conditions** | | |
| The interest Rate information will be persisted in the database | | |
|  | | |
| **Business Rules** | | |
| 1.The input fields must not be null | | |
|  | | |
| **Alternative path** | | |
| If the input value is null the system should display “the fields must not be blank” | | |
|  | | |