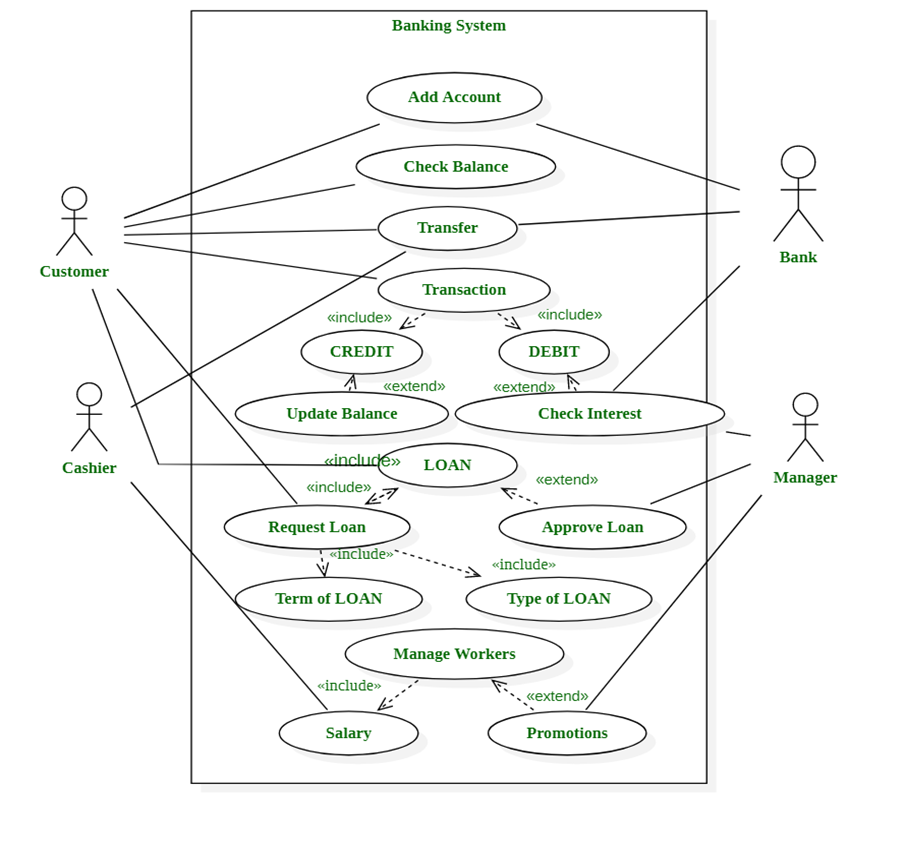


## **Use Case Diagram for Banking Applications**



## **HLR for Banking System**

|  |  |  |
| --- | --- | --- |
| 1 | **Bank Accounts** | Bank accounts hold information about their owner, balance, transactions, date of creation, etc. |
| 2 | **Money Transfer** | BankSystem supports two types of money transfers - internal and global / worldwide. |
| 3 | **Cards** | Cards are used for making purchases on other websites Example Credit/Debit Cards. |
| 4 | **Direct Payments** | Direct payments are a way to securely pay on websites directly through a bank account without the need to provide card details. |
| 5 | **Services** | Additional services that customers may want like (insurance, loan etc.). |
| 6 | **Branch or Employees** | Manager/Employee details of the concern bank. |

## **Test Scenarios for Banking Applications**

**New Branch**

* Create a new branch with valid and invalid test data.
* Create a new branch without data.
* Create a new branch with existing branch data.
* Verify the reset and cancel options.Update branch details with valid and invalid test data.
* Update branch details with existing branch test data.
* Verify if the new branch can be saved.
* Verify if the cancelation option is working.
* Verify the branch deletion with and without dependencies.
* Verify if the branch search option is working.

**New Role**

* Create a new role with valid and invalid test data.
* Create a new role without data.
* Verify if a new role can be created with existing test data.
* Verify the role description and role type.
* Verify that the cancelation and reset option is working.
* Verify the role deletion process with and without dependency.
* Verify the links in the role details page.
* Verify the admin login without test data.
* Verify all home links for the admin role.
* Verify if the admin can change the password with valid and invalid test data.
* Verify the admin log out successfully.

**User Login**

* Verify if all visitor and customer links are working properly.
* Verify the customer’s login with valid and invalid test data.
* Verify the customer’s login without any data.
* Verify the banker login without any data.
* Verify the banker’s login with valid or invalid test data.
* Verify if the customer or banker was able to log out successfully.

**Manage Users**

* Verify if the new user can be created with valid and invalid test data.
* Create a new user with existing branch test data
* Verify if the cancel and reset option is working properly.
* Update user details with valid and invalid test data.
* Verify the deletion of the new user.
* check if the new user can be verified.
* Verify mandatory input parameters.
* Verify optional input parameters.
* Verify if a user can be created without optional parameters.

**Account Creation**

* Create a new account with valid and invalid user data.
* Verify if the user details can be updated.
* Verify if a new user can be saved.
* Create a new account with existing user data.
* Verify that the user can deposit the amount into the newly created account (and update the balance).
* Verify if the user can withdraw the amount from the new account (after depositing and updating the balance).
* In the case of salary, the account verifys the company name and other details provided by the user.
* Verify if the primary account number is provided in case of a secondary account.
* Verify the user details provided in the case of the current account.
* Verify the provided proof for the joint account in case of a joint account.
* Verify whether you are able to maintain a zero balance in your salary account.
* Verify whether you are able to maintain a zero balance or minimum balance for a non-salary account.
* Verify that the new user was able to log out successfully.

**Net Banking Application**

* Check if the user is able to open the bank site.
* Check if all the links on the site are working.
* Verify if the user is able to create a new account.
* Check if the user is able to login with a valid and invalid username and password.
* Verify if either the username or password is blank while logged in, the user should not be allowed to login and an alert message should be shown.
* Check if the user is allowed to change the password.
* If an invalid username or password is entered, a proper error message will be shown.
* Users with an invalid password should not be allowed to log in.
* Verify that after repeated attempts to log in with an incorrect password, the user should be shown an error message and blocked.
* Check if the user is able to perform some basic transactions.
* Verify that the user is able to add a beneficiary with valid and invalid details.
* Verify if the user can delete the beneficiary.
* Verify that the user is able to make transactions to the newly added beneficiary.
* After the transaction, verify if the accounts of both the user and beneficiary have been updated.
* Check if the user is able to enter the amount in decimal number.
* Verify if the user is not able to enter negative numbers in the amount field.
* Verify if the user is allowed to make transactions with or without a minimum balance.
* Verify if the user can make a new RD.
* Verify that the correct message is showing in case of transaction done with insufficient balance.
* Check if the user is asked for confirmation before any transaction is made.
* Verify if acknowledgment receipts are provided on each successful transaction.
* Verify if the user is able to transfer money to multiple accounts.
* Verify if the user can cancel the transaction.
* Verify that the account details reflect the financial transactions also.
* Verify that the timeout feature has been implemented.
* Verify that in case of session time out a user should log in again.
* Verify that the proper session time out is done in case of any inactivity.
* Verify that while doing the transaction the user is taken to secure mode.
* Verify if the user was able to log out successfully.
* Verify search and reset options.

### **Mobile Banking**

* Test the installation and setup of the mobile banking app.
* Verify the compatibility with different mobile device types (iOS, Android).
* Test the responsiveness of the app on various screen sizes and resolutions.
* Verify that mobile check deposit functions correctly.
* Test mobile bill payment and funds transfer.
* Verify the accuracy of push notifications and alerts.
* Test offline functionality, such as viewing account balances without an internet connection.
* Verify the ability to toggle between multiple user accounts.
* Test the mobile app's synchronization with the web-based banking platform.
* Verify that the app is accessible to users with disabilities (ADA compliance).
* Test the performance of the mobile app under low-network conditions.
* Verify that biometric authentication (e.g., fingerprint, face ID) works securely.
* Test mobile app updates and their impact on existing data.
* Verify that the mobile app supports language preferences and localization.
* Test mobile app-specific features like remote account deactivation.
* Verify that mobile app data is protected against device theft or loss.
* Test the mobile app's ability to generate and scan QR codes for transactions.
* Verify the handling of interruptions (e.g., phone calls) during app use.
* Test the mobile app's accessibility on both smartphones and tablets.
* Verify the integration of mobile app analytics for performance monitoring.

**Cross-platform Compatibility for Banking Applications**

* Test the application on different web browsers (Chrome, Firefox, Safari, Edge, etc.).
* Verify compatibility with various mobile devices (iOS and Android) and their versions.
* Test on different operating systems (Windows, macOS, Linux) for web-based platforms.
* Verify the responsiveness and usability of the application on different screen sizes.
* Test the application on various screen resolutions (e.g., HD, Full HD, 4K).
* Verify compatibility with different input devices (mouse, touchpad, touchscreen).
* Test on different network speeds and conditions (e.g., 3G, 4G, Wi-Fi, slow connections).
* Verify compatibility with screen reader software for accessibility.
* Test for compatibility with third-party plugins or extensions in web browsers.
* Verify compatibility with different versions of Java (if applicable).
* Test the application's performance on low-end and high-end hardware.
* Verify compatibility with different email clients for transaction notifications.
* Test the application's compatibility with various PDF readers for statements.
* Verify that the application works seamlessly with different virtual private networks (VPNs).
* Test compatibility with screen rotation on mobile devices.
* Verify the application's compatibility with various firewall settings.
* Test the application's behavior on devices with different default languages.
* Verify that the application's UI elements adapt to the platform's design guidelines.
* Test compatibility with different security software and antivirus programs.
* Verify compatibility with various ad-blocking extensions or software.