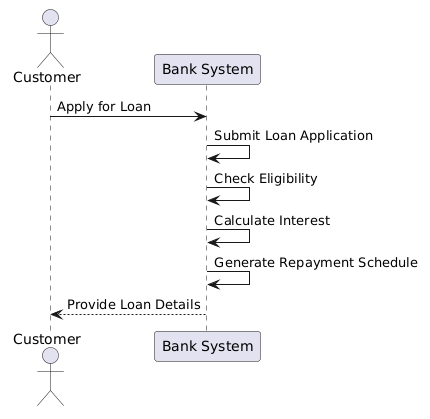
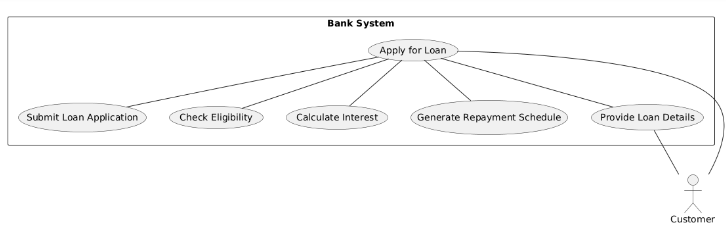
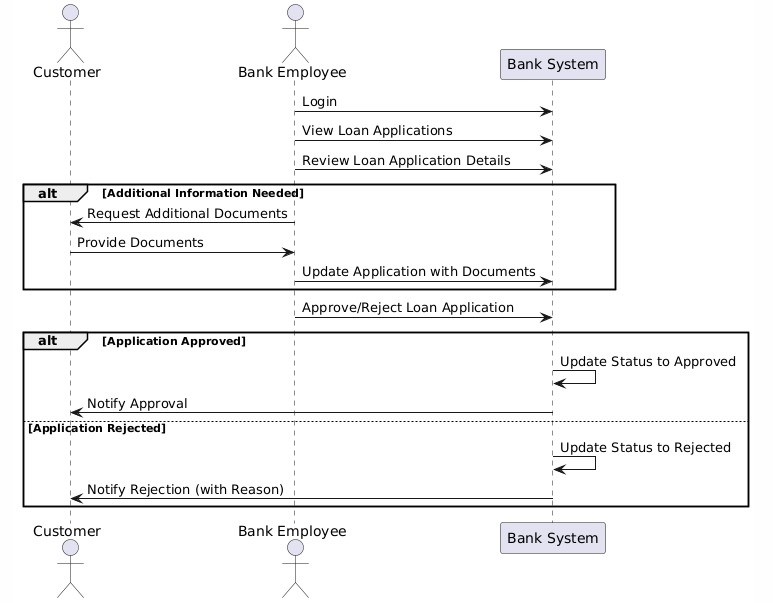
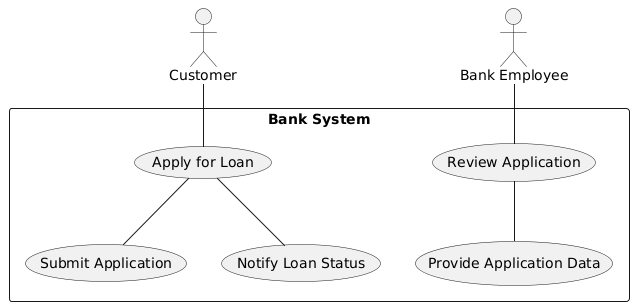
|  |  |
| --- | --- |
| UC Name | UC\_5 System shall handle loan logic (eligibility, interest, and repayment schedule). |
| Summary | This requirement specifies that the banking system must automatically manage the loan processing logic. It involves checking whether a customer meets the eligibility criteria, calculating the applicable interest based on defined parameters, and generating a detailed repayment schedule. Essentially, it streamlines and automates the evaluation and structuring of loans, ensuring consistent, accurate, and timely processing of loan applications. |
| Dependency | For this requirement the dependency primarly involves : Customer loan application  Customer account data(validated user or not) |
| Actors | Primary actor, the customer |
| Preconditions | There should be a valid loan application, a verified customer and the eligibility criteria and loan parameters should be added |
| Description of the main sequence | 1. The system receives a loan application from a customer. 2. The system retrieves the customer’s financial data (account balance, transaction history, income details, etc.). 3. The system checks the customer’s eligibility based on predefined criteria (credit score, income, debt-to-income ratio, etc.). 4. If the customer meets the criteria, the system calculates the loan interest rate based on the loan amount, duration and bank policies. 5. The system generates a repayment schedule detailing monthly payments, due dates, and total repayment amount. 6. The system stores the loan details and makes them available for review by a bank employee. 7. The system notifies the customer of the loan evaluation result. |
| Description of the alternative sequence | * If the customer doesn’t meet the eligibility criteria:  1. The system rejects the loan application. 2. The system generates a rejection message specifying the reason 3. The system notifies the customer for the result 4. The system logs the rejection for record-keeping.  * If the system encounters missing or invalid customer data:  1. The system flags the issue and does not proceed with the evaluation. 2. The system notifies the customer to update their details before reapplying. 3. The process ends until valid data is provided. |
| Non functional requirements | * Performance: The system shall process loan applications within 5 seconds to ensure efficiency. * Security: All customer financial data shall be encrypted to protect sensitive information * Accuracy: The system shall be accessible 24/7 to handle loan applications without downtime. * Scalability: The system should support simultaneous loan evaluations for multiple customers without performance degradation. |
| Postconditions | * If the customer is eligible, the loan details are stored in the system and await approval from bank employees. * If the customer is not eligible, the system logs the rejection and notifies the customer. * The system has recorded loan request for future reference, regardless of the outcome. |



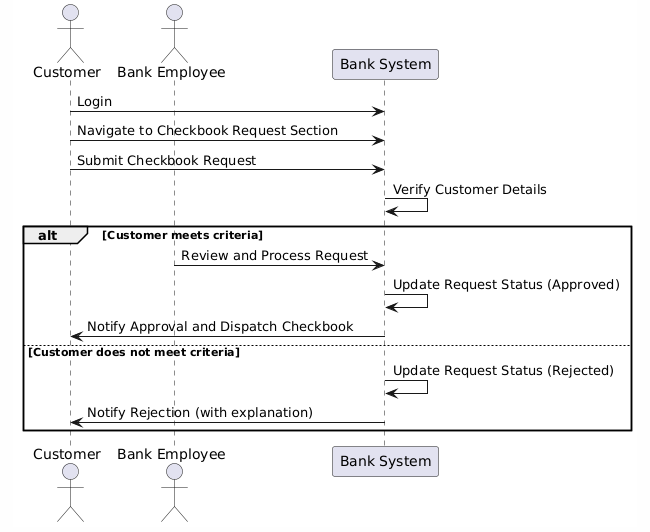


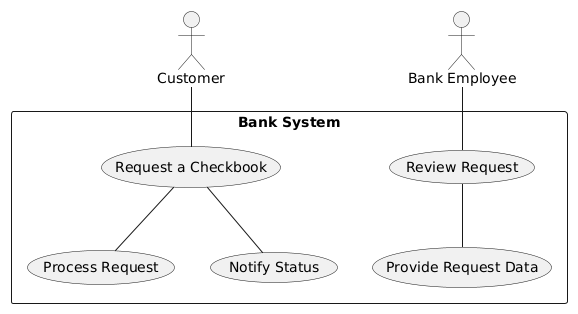
|  |  |
| --- | --- |
| UC Name | UC\_6 Bank employee shall manage and approve/reject customer loan applications. |
| Summary | This use case involves the bank employee reviewing customer loan applications and deciding whether to approve or reject them. |
| Dependency | |  | | --- | |  |   Depends on the "Handle Loan Logic" use case for the customer loan data being available. |
| Actors | Bank Employee |
| Preconditions | |  | | --- | | - The bank employee is logged in to the system.  Customer loan application has been submitted and is available for review after the system approved or rejected the loan application of the customer | |
| Description of the main sequence | 1. Bank employee logs into the system.  2. Bank employee views customer loan applications in the system.  3. Bank employee reviews the loan application details.  4. Bank employee approves or rejects the loan application.  5. System updates the loan application status to approved/rejected.  6. Customer is notified of the decision. |
| Description of the alternative sequence | * If the bank employee needs more information, they may request additional documents from the customer before making a decision. * If the employee rejects the application, the system notifies the customer with a rejection reason. |
| Non functional requirements | * Performance: The system shall process approval/rejection within 5 seconds to ensure efficiency. * Security: The system must ensure that customer data is handled securely during the review process. * Accuracy: The system shall be accessible 24/7 to handle approvals/rejections without downtime. * Scalability: The system should support simultaneous approvals/rejections for multiple bank employees without performance degradation. |
| Postconditions | * The loan application status is updated in the system as either approved or rejected. * The customer is notified of the outcome of the application. |





|  |  |
| --- | --- |
| UC Name | UC\_7 Customer shall be able to request a checkbook. |
| Summary | This use case involves the customer requesting a checkbook through the banking system. The bank processes the request and sends the checkbook to the customer. |
| Dependency | |  | | --- | |  |   none |
| Actors | Bank Employee, Customer |
| Preconditions | |  | | --- | | - The customer has a valid account.  - The customer is logged into the banking system. | |
| Description of the main sequence | 1. Customer logs into the system.  2. Customer navigates to the checkbook request section.  3. Customer submits a request for a checkbook.  4. System verifies customer details.  5. Bank employee reviews the request and processes it.  6. System updates the checkbook request status.  7. The customer is notified once the request is approved and the checkbook is sent. |
| Description of the alternative sequence | -If the customer does not meet the criteria for receiving a checkbook, the request is rejected, and the customer is notified.  -If the customer is not eligible, a message is displayed explaining the reason. |
| Non functional requirements | * Performance: The system should process checkbook requests within 5 seconds. * Security: The system must ensure secure handling of customer data during the request process, customer data should be stored securely and be kept private. * Scalability: The system should support simultaneous checkbook requrests for multiple bank employees without performance degradation. |
| Postconditions | * The checkbook request status is updated in the system. * The customer is notified of the outcome of the request. |





|  |  |
| --- | --- |
| UC Name | UC\_8 Bank employee shall be able to approve or reject account creation requests.. |
| Summary | This use case involves the customer requesting a checkbook through the banking system. The bank processes the request and sends the checkbook to the customer. |
| Dependency | |  | | --- | |  |   The customer must have submitted a valid account creation request |
| Actors | Bank Employee, Customer |
| Preconditions | |  | | --- | | - The customer has a valid account.  - The customer is logged into the banking system. | |
| Description of the main sequence | |  | | --- | | 1. The customer submits an account creation request. |  |  |  | | --- | --- | |  | 2. The system processes the request and validates the provided details. |  |  |  | | --- | --- | |  | 3. The bank employee reviews the request and checks customer details. |  |  |  | | --- | --- | |  | 4. The bank employee approves or rejects the account creation request. |  |  |  | | --- | --- | |  | 5. The system updates the customer account status (approved or rejected) and notifies the customer. | |
| Description of the alternative sequence | -If the customer’s details are invalid, the system rejects the request and notifies the customer.  -if the customer’s details are valid and the bank employee approved the account creation, the system creates the account and the customer is notified, also a PIN is created.  - If the bank employee decides to reject the request, the system notifies the customer with a rejection notice |
| Non functional requirements | * Performance: The system should process the account creation requests within 5 seconds. * Security: The system must ensure secure handling of customer data during the request process, customer data should be stored securely and be kept private. * Scalability: The system should support simultaneous account requests for multiple customers without performance degradation. |
| Postconditions | * The customer request status is updated in the system. * The customer is notified of the outcome of the request. |

