**Soar Test | User - Stories**

### User Story - 1

**Title:** Registered User Login From a New Device

**Description:**

As a registered user, I want to be able to login from a new device and logout from all other devices, so that I can access my account and services from anywhere and make sure that my account is secure and can only be accessed from one device at a time.

**Acceptance Criteria:**

Successful Login.

**Given** that my account has been already created in the system.

**When** I am on the login page.

**And**, I click on "Login" Button.

**Then** I should be able to enter my phone number and password.

**And** I should be able to click on next button.

**And** the system should validate my inputs.

**Then** I should be redirected to enter the received OTP.

**And** I should be able to click on "Login" button, and login successfully.

**And** the system should notify me with a successful login message.

**And** the system should send SMS notification to the user after the successful login from new

device.

**And** the user account should be Logged out from all other logged-in devices.

**And** I should be redirected to the home page.

## 

### Test Plan for "Registered User Login From a New Device"

**1. Test Objectives**

* Verify successful login from a new device.
* Ensure proper handling of login attempts from multiple devices.
* Validate user account logout from other devices upon successful login from a new device.
* Verify SMS notification is sent after successful login from a new device.
* Confirm user redirection to the home page after successful login.

**2. Test Scope**

* **In Scope:**
  + Login functionality from a new device.
  + OTP verification process.
  + Logout from other devices.
  + SMS notification for new device login.
  + User redirection after successful login.
* **Out of Scope:**
  + Account creation functionality (assumed to be working correctly).
  + Network connectivity issues.
  + Device-specific issues (e.g., browser compatibility, device limitations).
  + Performance and load testing.

**3. Test Strategy**

* **Testing Techniques:**
  + Functional Testing
  + Regression Testing
  + Smoke Testing
  + Integration Testing (if applicable)
* **Test Environments:**
  + Different browsers (Chrome, Firefox, Safari, Edge)
  + Different operating systems (Windows, macOS, iOS, Android)
  + Different devices (desktops, laptops, mobiles, tablets)
* **Test Data:**
  + Valid user credentials.
  + Invalid user credentials.
  + Different types of devices (phones, tablets, laptops).
  + Different network conditions (Wi-Fi, cellular).

**4. Test Deliverables**

* Test Plan document
* Test Cases document
* Test Execution Report
* Bug Reports (if any)

**5. Entry Criteria**

* Application build with the latest code changes.
* Test environment setup and ready for testing.
* Test data prepared.

**6. Exit Criteria**

* All test cases executed successfully with no critical or major defects.
* Test coverage meets the defined requirements.
* Test results documented and reviewed.

**7. Risks and Mitigation**

* **Risk:** Delay in application build or deployment.
  + **Mitigation:** Close communication with the development team and proactive risk management.
* **Risk:** Unstable test environment.
  + **Mitigation:** Regular environment checks and maintenance.
* **Risk:** Insufficient test data.
  + **Mitigation:** Thorough review and preparation of test data.

**8. Responsibilities**

* **Test Lead:** Oversee the entire testing process, coordinate with the team, and ensure timely completion.
* **Testers:** Execute test cases, log defects, and report test results.
* **Developers:** Fix identified defects and provide necessary support.

## **Risk-Based Testing (RBT)**

* **High Risk:**
  + **Risk:** Incorrect OTP generation or verification.
  + **Impact:** Failed login attempts, account security breaches.
  + **Mitigation:** Thorough testing of OTP generation, delivery, and verification mechanisms.
  + **Risk:** Data breaches due to insecure session management.
  + **Impact:** Unauthorized access to user accounts.
  + **Mitigation:** Focus on testing session management, authentication mechanisms, and data encryption.
* **Medium Risk:**
  + **Risk:** Incorrect SMS notification delivery.
  + **Impact:** User confusion, delayed account access.
  + **Mitigation:** Test SMS delivery on different devices and network conditions.
  + **Risk:** Incorrect handling of concurrent login attempts.
  + **Impact:** Account lockout, user frustration.
  + **Mitigation:** Test scenarios with multiple devices attempting to login simultaneously.
* **Low Risk:**
  + **Risk:** Minor UI/UX issues.
  + **Impact:** Minor usability concerns.
  + **Mitigation:** Thorough UI/UX testing and user feedback.

## **Test Cases**

**Acceptance Test Cases:**

* **TC\_01**: Verify successful login from a new device with valid credentials and OTP.
* **TC\_02**: Verify successful login from a new web browser (Chrome, Firefox, Edge).
* **TC\_03**: Verify successful login from a new mobile device (Android, iOS).
* **TC\_04**: Verify successful login from a tablet device.
* **TC\_05**: Verify successful login with valid phone number and password.
* **TC\_06**: Verify successful OTP verification and login.
* **TC\_07**: Verify successful login with SMS notification.
* **TC\_08**: Verify successful logout from all other devices after login.
* **TC\_09**: Verify successful redirection to the home page after login.

**Negative Test Cases:**

* **TC\_10**: Verify login failure with invalid phone number.
* **TC\_11**: Verify login failure with invalid password.
* **TC\_12**: Verify login failure with incorrect OTP.
* **TC\_13**: Verify login failure with expired OTP.
* **TC\_14**: Verify login failure with locked account (due to multiple failed attempts).
* **TC\_15**: Verify login failure with account locked for security reasons.
* **TC\_16**: Verify login failure with invalid phone number format.
* **TC\_17**: Verify login failure with special characters in phone number.
* **TC\_18**: Verify login failure with special characters in password.
* **TC\_19**: Verify login failure with empty phone number field.
* **TC\_20**: Verify login failure with empty password field.

**Edge Case Test Cases:**

* **TC\_21**: Verify login behavior with intermittent network connectivity.
* **TC\_22**: Verify login with very long phone number.
* **TC\_23**: Verify login with very short phone number.
* **TC\_24**: Verify login with very long password.
* **TC\_25**: Verify login with very short password.
* **TC\_26**: Verify login with case-sensitive credentials.
* **TC\_27**: Verify login with two-factor authentication enabled (if applicable).
* **TC\_28**: Verify login with biometric authentication (if applicable).
* **TC\_29**: Verify login with multiple devices simultaneously.
* **TC\_30**: Verify login from different locations (if applicable).

**Test Runs**

* **Test Run 1:** Smoke Test - Execute a subset of critical test cases to ensure basic functionality.
* **Test Run 2:** Functional Test - Execute all acceptance and edge case test cases.
* **Test Run 3:** Regression Test - Re-execute critical test cases after code changes or bug fixes.

### User Story - 2

**Title**: Individual Investor - Upgrade To Premium

**Description:**

As a verified individual user, I want to upgrade my regular account to premium account, so

that I can be able to get all the premium account features.

**Acceptance criteria**

**Given** that I am a verified individual investor user.

**And** I want to upgrade to premium account.

**When** I click on "upgrade to premium" button.

**Then** I should be able to choose at least one of the following types:

• Do you have assets worth amount 3 million SAR?

• Do you work or has been worked before in the financial sector position related to

investment or finance for at least 3 years?

• Do you have a certificate in finance or investment sector from an accredited

internationally recognized organization?

**When** I selected one option from the previous page.

**Then** I should be able to upload documents, and prove ownership of these documents.

**And** the compliance department staff members should be able to view and approve the

upgrade request.

**And** the relation manager responsible of this user account should be notified by the

compliance team about the request status.

**And** I should get a notification once the documents are uploaded successfully.

## **Test Plan for "Individual Investor - Upgrade To Premium"**

**1. Test Objectives**

* Verify successful account upgrade from regular to premium.
* Ensure proper handling of user selection of upgrade eligibility criteria.
* Validate document upload functionality.
* Confirm notification to the user upon successful document upload.
* Verify notification to the compliance department and relationship manager.
* Ensure proper processing and approval of upgrade requests by the compliance team/role.

**2. Test Scope**

* **In Scope:**
  + Account upgrade process from regular to premium.
  + Eligibility criteria selection.
  + Document upload functionality.
  + Compliance team approval process.
  + Notifications to user, compliance team, and relationship manager.
* **Out of Scope:**
  + Account creation and verification (assumed to be working correctly).
  + Performance and load testing of the upgrade process.
  + Integration testing with external systems

**3. Test Strategy**

* **Testing Techniques:**
  + Functional Testing
  + Regression Testing
  + Smoke Testing
  + Usability Testing
* **Test Environments:**
  + Different browsers (Chrome, Firefox, Safari, Edge)
  + Different operating systems (Windows, macOS, iOS, Android)
  + Different devices (desktops, laptops, mobiles, tablets)
* **Test Data:**
  + Valid and invalid user credentials.
  + Valid and invalid documents (e.g., image formats, file sizes).
  + Different eligibility criteria combinations.

**4. Test Deliverables**

* Test Plan document
* Test Cases document
* Test Execution Report
* Bug Reports (if any)

**5. Entry Criteria**

* Application build with the latest code changes.
* Test environment setup and ready for testing.
* Test data prepared.
* Compliance team and relationship manager access granted.

**6. Exit Criteria**

* All test cases executed successfully with no critical or major defects.
* Test coverage meets the defined requirements.
* Test results documented and reviewed.
* All identified defects are fixed and retested.

**7. Risks and Mitigation**

* **Risk:** Incorrect processing of upgrade requests by the compliance team.
  + **Mitigation:** Detailed test cases covering all approval scenarios, regular review of compliance team workflows.
* **Risk:** Data loss or corruption during document upload.
  + **Mitigation:** Implement robust data validation and error handling mechanisms.
* **Risk:** Delayed notifications to users, compliance team, and relationship managers.
  + **Mitigation:** Monitor notification delivery mechanisms and investigate any delays.

**8. Responsibilities**

* **Test Lead:** Oversee the entire testing process, coordinate with the team, and ensure timely completion.
* **Testers:** Execute test cases, log defects, and report test results.
* **Developers:** Fix identified defects and provide necessary support.
* **Compliance Team:** Participate in testing and provide feedback on the upgrade process.
* **Relationship Managers:** Verify notification receipt and provide feedback on the upgrade process.

## **Risk-Based Testing (RBT)**

**High Risk:**

* **Risk:** Incorrect processing of upgrade requests by the compliance team.
  + **Impact:** Incorrect account upgrades, potential financial risks.
  + **Mitigation:** Prioritize testing of compliance team workflows and approval logic.
* **Risk:** Data breaches during document upload and storage.
  + **Impact:** Loss of sensitive user information.
  + **Mitigation:** Implement strong security measures for data storage and transmission.

**Medium Risk:**

* **Risk:** Incorrect eligibility criteria validation.
  + **Impact:** Incorrect account upgrades, potential fraud.
  + **Mitigation:** Comprehensive testing of all eligibility criteria combinations.
* **Risk:** Delayed or failed notifications.
  + **Impact:** User frustration, delayed account activation.
  + **Mitigation:** Monitor notification delivery mechanisms closely.

**Low Risk:**

* **Risk:** Minor UI/UX issues related to the upgrade process.
  + **Impact:** Minor usability concerns.
  + **Mitigation:** Thorough UI/UX testing and user feedback.

## **Test Cases**

**Acceptance Test Cases:**

* **TC\_01:** Verify successful account upgrade with valid eligibility criteria and documents.
* **TC\_02:** Verify successful document upload with different file formats (e.g., PDF, JPG).
* **TC\_03:** Verify notification to the user upon successful document upload.
* **TC\_04:** Verify notification to the compliance team upon upgrade request submission with documents uploaded.
* **TC\_05:** Verify notification to the relationship manager regarding the upgrade request.
* **TC\_06:** Verify successful approval of the upgrade request by the compliance team.
* **TC\_07:** Verify rejection of the upgrade request by the compliance team.
* **TC\_08:** Verify account upgrade status change after approval.
* **TC\_09:** Verify inability to submit account upgrade request / uploading documents without proving ownership of these documents.

**Edge Case Test Cases:**

* **TC\_10:** Verify account upgrade with invalid eligibility criteria.
* **TC\_11:** Verify account upgrade with invalid or missing documents.
* **TC\_12:** Verify account upgrade with large file uploads.
* **TC\_13:** Verify account upgrade with multiple simultaneous requests.
* **TC\_14:** Verify account upgrade with duplicate requests.
* **TC\_15:** Verify account upgrade with invalid user credentials.
* **TC\_16:** Verify account upgrade with inactive user accounts.
* **TC\_17:** Verify account upgrade form submission with expired user session.
* **TC\_18:** Verify account upgrade form submission with user deleted from another session.
* **TC\_19:** Verify "upgrade to premium" button not displayed for existing Premium user

**Test Runs**

* **Test Run 1:** Smoke Test - Execute a subset of critical test cases to ensure basic functionality.
* **Test Run 2:** Functional Test - Execute all acceptance and edge case test cases.
* **Test Run 3:** Regression Test - Re-execute critical test cases after code changes or bug fixes.
* **Test Run 4:** User Acceptance Testing (UAT) - Involve real users to test the upgrade process and gather feedback.

### User Story - 3

**Title**: Approvals Notifications | SMS

**Description**:

As a loan requester, I want to receive timely notifications about the status of my loan

request so that I can stay informed throughout the approval process.

**Acceptance Criteria**:

**Scenario 1**: Approval Notification

**Given**: that all relevant departments and the relationship manager have approved a loan

request.

**When** the system processes the final approval.

**Then** the loan requester should receive an SMS indicating the approval status.

**Scenario 2**: Rejection Notification with Reasons

**Given** that a loan request has been rejected at any stage of the approvals process with

specified reasons.

**When** the system processes the rejection at any step during departments approvals stage.

**Then** the Relation manager responsible of this request should be notified to communicate

the reasons with the loan requester and solve the issues.

**When** the relation manager rejects the loan request. (Final Rejection)

**Then** the loan requester should receive an SMS indicating the rejection with the specified

reasons provided by the relation manager only.

## **Test Plan for "Approvals Notifications | SMS"**

**1. Test Objectives**

* **Scenario 1: Approval Notification**
  + Verify that the loan requester receives an SMS notification upon final loan approval.
  + Ensure the SMS notification contains accurate and relevant information (e.g., loan amount, approval status).
* **Scenario 2: Rejection Notification with Reasons**
  + Verify that the loan requester receives an SMS notification upon rejection of the loan request.
  + Ensure the SMS notification includes the reason for rejection (if provided by any department).
  + Verify that the relationship manager is notified about rejections at any stage.
  + Ensure the relationship manager notification includes relevant information (e.g., loan request ID, rejection reason).

**2. Test Scope**

* **In Scope:**
  + SMS notification delivery for loan approvals and rejections.
  + Accuracy and completeness of SMS notification content.
  + Notification delivery to the loan requester and relationship manager.
* **Out of Scope:**
  + Loan application process itself.
  + Internal approval workflows within the system.
  + Network connectivity issues.
  + User-specific SMS delivery issues (e.g., network outages).

**3. Test Strategy**

* **Testing Techniques:**
  + Functional Testing
  + Integration Testing (with SMS gateway)
  + Smoke Testing
* **Test Environments:**
  + Simulated loan application scenarios.
  + Different SMS gateways (if applicable).
  + Different mobile devices and network conditions.
* **Test Data:**
  + Valid loan applications with different approval/rejection scenarios.
  + Test phone numbers for loan requesters and relationship managers.

**4. Test Deliverables**

* Test Plan document
* Test Cases document
* Test Execution Report
* Bug Reports (if any)

**5. Entry Criteria**

* Application build with the latest code changes.
* Test environment setup and ready for testing.
* Test data prepared.
* SMS gateway integration configured and tested.

**6. Exit Criteria**

* All test cases executed successfully with no critical or major defects.
* Test coverage meets the defined requirements.
* Test results documented and reviewed.
* All identified defects are fixed and retested.

**7. Risks and Mitigation**

* **Risk:** Incorrect SMS delivery due to integration issues with the SMS gateway.
  + **Mitigation:** Thorough testing of SMS gateway integration, monitoring delivery rates, and addressing any integration problems.
* **Risk:** Inaccurate or incomplete information in SMS notifications.
  + **Mitigation:** Careful review of SMS templates and data mapping and total allowable character limit.
* **Risk:** Delays in SMS delivery.
  + **Mitigation:** Monitor SMS delivery times and investigate any delays.

**8. Responsibilities**

* **Test Lead:** Oversee the entire testing process, coordinate with the team, and ensure timely completion.
* **Testers:** Execute test cases, log defects, and report test results.
* **Developers:** Fix identified defects and provide necessary support.
* **SMS Gateway Provider:** Collaborate on integration and troubleshooting if needed.

## **Risk-Based Testing (RBT)**

**High Risk:**

* **Risk:** Incorrect delivery of rejection notifications to the loan requester.
  + **Impact:** Misinformed users, potential legal issues.
  + **Mitigation:** Prioritize testing of rejection notification scenarios with different rejection reasons and at various stages of the approval process.
* **Risk:** Data breaches due to insecure SMS transmission.
  + **Impact:** Sensitive user information exposure.
  + **Mitigation:** Implement secure SMS encryption and authentication mechanisms.

**Medium Risk:**

* **Risk:** Delays in SMS delivery for critical notifications (e.g., loan approvals).
  + **Impact:** User frustration, potential missed opportunities.
  + **Mitigation:** Monitor SMS delivery times closely and investigate any delays.
* **Risk:** Incorrect formatting or content in SMS notifications.
  + **Impact:** User confusion, difficulty understanding the message.
  + **Mitigation:** Carefully review and test SMS templates for clarity and accuracy.

**Low Risk:**

* **Risk:** Minor issues with SMS delivery to specific devices or networks.
  + **Impact:** Minor impact on user experience.
  + **Mitigation:** Test SMS delivery on different devices and network conditions.

## **Test Cases**

**Scenario 1: Approval Notification**

* **TC\_01:** Verify SMS notification is sent to the loan requester upon final loan approval.
* **TC\_02:** Verify SMS notification includes loan amount and approval status.
* **TC\_03:** Verify SMS notification is sent within a reasonable timeframe after final approval.

**Scenario 2: Rejection Notification with Reasons**

* **TC\_04:** Verify SMS notification is sent to the loan requester upon rejection by the relation manager.
* **TC\_05:** Verify SMS notification includes the reason for rejection (if provided).
* **TC\_06:** Verify notification to the relationship manager upon rejection at any stage.
* **TC\_07:** Verify relationship manager notification includes loan request ID and rejection reason.

**Edge Case Test Cases:**

* **TC\_08:** Verify SMS delivery on different mobile devices and network conditions.
* **TC\_09:** Verify SMS delivery with international phone numbers.
* **TC\_10:** Verify SMS delivery with incorrect phone numbers.
* **TC\_11:** Verify SMS delivery with network outages.
* **TC\_12:** Verify SMS delivery with different SMS gateways.
* **TC\_13:** Verify SMS delivery with different invalid characters on the message
* **TC\_14:** Verify SMS delivery with incomplete message received due to long texts
* **TC\_15:** Verify notification to load requester if there is a delay in processing the request
* **TC\_16:** Verify the loan is processed and statuses updated to the loan requester even if his account is deleted in-between during the approval process.

**Test Runs**

* **Test Run 1:** Smoke Test - Execute a subset of critical test cases to ensure basic functionality.
* **Test Run 2:** Functional Test - Execute all acceptance and edge case test cases.
* **Test Run 3:** Regression Test - Re-execute critical test cases after code changes or bug fixes.

### User Story - 4

**Title**: Departments Approval

**Description**: As a department staff member, I want to review and approve loan requests

from my department, so that I can contribute to the overall approval process.

**Acceptance Criteria**:

**Scenario 1**: Approving a Request In Review

**Given** that a loan request is currently in review by a specific department.

**When** the department representative reviews the request.

**Then** they should see an "Approve" button.

**When** the representative clicks "Approve".

**Then** the request should be marked as approved from their department.

**Scenario 2**: Viewing Department Approvals

**Given** that a loan request has been approved by multiple departments.

**When** the system user views the details of the request.

**Then** they should see a section displaying all departments that have approved the

request, with the staff members who gave the approval and the timestamp of each

approval.

## **Test Plan for "Departments Approval"**

**1. Test Objectives**

* **Scenario 1: Approving a Request In Review**
  + Verify the presence of the "Approve" button for requests assigned to the department.
  + Ensure successful approval of requests by department representatives.
  + Validate the update of request status after approval.
* **Scenario 2: Viewing Department Approvals**
  + Verify the display of all departments that have approved the request.
  + Ensure the correct display of approving staff members for each department.
  + Validate the display of timestamps for each department approval.

**2. Test Scope**

* **In Scope:**
  + Department approval process for loan requests.
  + Display of approval history for each loan request.
  + Access control for department representatives.
* **Out of Scope:**
  + Loan application process itself.
  + Loan processing logic by other departments.
  + Performance and load testing of the approval system.

**3. Test Strategy**

* **Testing Techniques:**
  + Functional Testing
  + Usability Testing
  + Regression Testing
  + Smoke Testing
* **Test Environments:**
  + Different browsers (Chrome, Firefox, Safari, Edge)
  + Different operating systems (Windows, macOS, iOS, Android)
  + Different user roles (department representatives, administrators)
* **Test Data:**
  + Valid loan requests in different stages of the approval process.
  + Multiple departments with different approval authorities.

**4. Test Deliverables**

* Test Plan document
* Test Cases document
* Test Execution Report
* Bug Reports (if any)

**5. Entry Criteria**

* Application build with the latest code changes.
* Test environment setup and ready for testing.
* Test data prepared.
* User accounts created for different department representatives.

**6. Exit Criteria**

* All test cases executed successfully with no critical or major defects.
* Test coverage meets the defined requirements.
* Test results documented and reviewed.
* All identified defects are fixed and retested.

**7. Risks and Mitigation**

* **Risk:** Incorrect approval logic leading to incorrect loan approvals.
  + **Mitigation:** Thorough testing of approval workflows and access controls.
* **Risk:** Data inconsistencies in approval history.
  + **Mitigation:** Regular data integrity checks and validation.
* **Risk:** Lack of visibility for department representatives on their approval queue.
  + **Mitigation:** Ensure clear and intuitive user interface for viewing and managing pending approvals.

**8. Responsibilities**

* **Test Lead:** Oversee the entire testing process, coordinate with the team, and ensure timely completion.
* **Testers:** Execute test cases, log defects, and report test results.
* **Developers:** Fix identified defects and provide necessary support.
* **Department Representatives (if possible):** Participate in user acceptance testing.

## **Risk-Based Testing (RBT)**

**High Risk:**

* **Risk:** Incorrect approval of loan requests.
  + **Impact:** Financial losses, legal issues.
  + **Mitigation:** Prioritize testing of approval logic, access controls, and data validation.
* **Risk:** Data inconsistencies in approval history.
  + **Impact:** Difficulty in tracking loan request statuses, potential audit issues.
  + **Mitigation:** Implement robust data integrity checks and validation mechanisms.

**Medium Risk:**

* **Risk:** Poor user experience for department representatives.
  + **Impact:** Reduced efficiency, increased frustration.
  + **Mitigation:** Conduct usability testing with department representatives to identify and address any usability issues.
* **Risk:** Lack of clear audit trails for approvals.
  + **Impact:** Difficulty in tracking who approved what and when.
  + **Mitigation:** Implement robust audit logging for all approval actions.

**Low Risk:**

* **Risk:** Minor UI/UX issues related to the approval interface.
  + **Impact:** Minor usability concerns.
  + **Mitigation:** Thorough UI/UX testing and user feedback.

## **Test Cases**

**Scenario 1: Approving a Request In Review**

* **TC\_01\_S1**: Verify the presence of the "Approve" button for requests assigned to the department.
* **TC\_02\_S1**: Verify successful approval of a request by a department representative.
* **TC\_03\_S1**: Verify the update of request status to "Approved" by the department after approval.
* **TC\_04\_S1**: Verify approval action is logged correctly in the system audit trail.
* **TC\_05\_S1**: Verify approval action is logged with the correct timestamp.
* **TC\_06\_S1**: Verify approval action is logged with the correct user ID/name of the approving representative.
* **TC\_07\_S1**: Verify approval action cannot be undone once approved.
* **TC\_08\_S1**: Verify "Approve" button is not visible for requests already approved by the department.
* **TC\_09\_S1**: Verify "Approve" button is not visible for requests rejected by other departments.
* **TC\_10\_S1**: Verify approval functionality for different types of loan requests (e.g., personal loan, business loan).
* **TC\_11\_S1**: Verify application status set to in-review for department once staff picks the request for review, so the task is no longer available for other staff members of the department

**Scenario 2: Viewing Department Approvals**

* **TC\_01\_S2**: Verify the display of all departments that have approved the request.
* **TC\_02\_S2**: Verify the correct display of approving staff members for each department.
* **TC\_03\_S2**: Verify the display of timestamps for each department approval.
* **TC\_04\_S2**: Verify the accuracy and completeness of approval history information.
* **TC\_05\_S2**: Verify the approval history section is easily accessible and understandable.
* **TC\_06\_S2**: Verify the approval history section is visually appealing and user-friendly.
* **TC\_07\_S2**: Verify the approval history section is updated in real-time after each approval.

**Edge Case Scenarios:**

* **TC\_01\_E**: Verify approval behavior when multiple users from the same department attempt to approve the same request simultaneously.
* **TC\_02\_E**: Verify approval behavior when the system experiences temporary downtime during the approval process.
* **TC\_03\_E**: Verify approval behavior with different user roles (e.g., administrators, managers).
* **TC\_04\_E**: Verify approval behavior with different access permissions for different users.
* **TC\_05\_E**: Verify access control for department representatives (only authorized users should be able to approve requests).
* **TC\_06\_E**: Verify approval behavior with network connectivity issues.

**Test Runs**

* **Test Run 1:** Smoke Test - Execute a subset of critical test cases to ensure basic functionality.
* **Test Run 2:** Functional Test - Execute all acceptance and edge case test cases.
* **Test Run 3:** Regression Test - Re-execute critical test cases after code changes or bug fixes.
* **Test Run 4:** User Acceptance Testing (UAT) - Involve department representatives in user acceptance testing.

### User Story - 5

**Title**: Corporate investor - Commercial Registration Verification Via [X]

**Description**:

As a system, I want to confirm the commercial register for corporate investors using third-

party service [X], so that I can retrieve all necessary information about corporate clients

who want to invest in our system.

**Acceptance Criteria**:

**Given** that the system is integrated with [X].

**And** I have finished my commercial registration successfully.

**Then** the system should send my commercial registration number to [X].

**Scenario1**: Valid Data

**Given** that I want to verify corporate investor commercial registration number.

**When** the system gets result from third party (X) that my commercial registration number

inputs is valid.

**Then** I should be able to proceed now to the validation process.

**Scenario 2**: Invalid Data

**Given** that I have entered an invalid commercial registration number.

**When** the system gets result from third party (X) that my commercial registration number

is invalid.

**Then** the system should ask me to enter the valid number for commercial registration.

**And** after filling the new commercial registration number, the system should restart the

verification process with third party (X) with the new number.

**Scenario 3**: Validation for the Start Date is Negative

**Given** that the verification process via third party (X) is successfully done.

**When** the system got the result from a third party (X).

**Then** the system should validate the output number:15 (company start date).

**And**:

• If it is less than two years, system will notify the user that he will not be able to

proceed.

• If it is more than two years, the system will proceed with the user to the next step.

## **Test Plan for "Corporate Investor - Commercial Registration Verification Via [X]"**

**1. Test Objectives**

* **Verify successful integration with third-party service [X] for commercial registration verification.**
* **Validate data exchange between the system and service [X].**
* **Ensure accurate retrieval of corporate information from service [X].**
* **Test the handling of valid and invalid commercial registration numbers.**
* **Verify system behavior for companies with less than two years of operation.**
* **Confirm appropriate user notifications and error messages.**

**2. Test Scope**

* **In Scope:**
  + Integration with third-party service [X].
  + Data exchange with service [X] (commercial registration number, response data).
  + Validation of commercial registration number.
  + Handling of invalid registration numbers.
  + Start date validation and user notifications.
* **Out of Scope:**
  + Internal system logic not directly related to the integration with service [X].
  + Performance and load testing of the integration.
  + Security testing of the integration.

**3. Test Strategy**

* **Testing Techniques:**
  + Functional Testing
  + Integration Testing
  + Smoke Testing
  + Data Driven Testing
* **Test Environments:**
  + Test environment with access to service [X] (if available).
  + Simulated environment with mocked responses from service [X].
* **Test Data:**
  + Valid and invalid commercial registration numbers.
  + Test data for companies with different start dates.

**4. Test Deliverables**

* Test Plan document
* Test Cases document
* Test Execution Report
* Bug Reports (if any)

**5. Entry Criteria**

* Application build with the latest code changes.
* Test environment setup and ready for testing.
* Access to service [X] (or a suitable test environment).
* Test data prepared.

**6. Exit Criteria**

* All test cases executed successfully with no critical or major defects.
* Test coverage meets the defined requirements.
* Test results documented and reviewed.
* All identified defects are fixed and retested.

**7. Risks and Mitigation**

* **Risk:** Integration issues with third-party service [X].
  + **Mitigation:** Thorough testing of the integration, including data exchange, error handling, and authentication.
* **Risk:** Data accuracy and reliability from service [X].
  + **Mitigation:** Validate the accuracy of data received from service [X] against known sources.
* **Risk:** System downtime due to service [X] unavailability.
  + **Mitigation:** Implement appropriate error handling and fallback mechanisms.

**8. Responsibilities**

* **Test Lead:** Oversee the entire testing process, coordinate with the team, and ensure timely completion.
* **Testers:** Execute test cases, log defects, and report test results.
* **Developers:** Fix identified defects and provide necessary support.
* **Integration Team:** Collaborate on integration testing with service [X].

## **Risk-Based Testing (RBT)**

**High Risk:**

* **Risk:** Incorrect validation of commercial registration numbers.
  + **Impact:** Incorrect data entry, fraudulent activities.
  + **Mitigation:** Prioritize testing of various scenarios with valid and invalid registration numbers.
* **Risk:** Data privacy and security breaches during data exchange with service [X].
  + **Impact:** Legal and reputational damage.
  + **Mitigation:** Implement robust security measures for data transmission and storage.

**Medium Risk:**

* **Risk:** Incorrect handling of start date validation.
  + **Impact:** Incorrect user guidance, potential business losses.
  + **Mitigation:** Thoroughly test start date validation logic and user notifications.
* **Risk:** Service [X] unavailability or downtime.
  + **Impact:** System interruptions, user frustration.
  + **Mitigation:** Implement robust error handling and fallback mechanisms.

**Low Risk:**

* **Risk:** Minor UI/UX issues related to the registration verification process.
  + **Impact:** Minor usability concerns.
  + **Mitigation:** Conduct usability testing to identify and address any UI/UX issues.

## **Test Cases**

**Scenario 1: Valid Data**

* **TC\_01:** Verify successful retrieval of valid corporate information from service [X] for a valid registration number.
* **TC\_02:** Verify correct data mapping and storage of retrieved information (company name, address, registration date, etc.).
* **TC\_03:** Verify successful progression to the next step in the user journey after successful verification.
* **TC\_04:** Verify the system handles different formats of valid registration numbers (e.g., with/without spaces, hyphens, etc.).
* **TC\_05:** Verify the system handles different character encodings for registration numbers.

**Scenario 2: Invalid Data**

* **TC\_06:** Verify error handling and user notification for invalid registration numbers (e.g., incorrect format, non-existent number).
* **TC\_07:** Verify system prompts the user to re-enter the registration number with clear instructions.
* **TC\_08:** Verify successful re-verification with the corrected registration number.
* **TC\_09:** Verify the system prevents further processing until a valid registration number is entered.
* **TC\_10:** Verify the system limits the number of re-entry attempts for invalid numbers.

**Scenario 3: Validation for the Start Date is Negative**

* **TC\_11:** Verify system behavior for companies with less than two years of operation.
* **TC\_12:** Verify the notification to the user about the restriction with clear and concise messaging.
* **TC\_13:** Verify system prevents further processing for companies with less than two years of operation.
* **TC\_14:** Verify the notification message accurately reflects the company's start date and the two-year restriction.

**Edge Case Test Cases**

* **TC\_15:** Test with edge cases for start dates (e.g., exactly two years old, less than two years old by a few days).
* **TC\_16:** Test with service [X] experiencing temporary unavailability (simulate network issues, timeouts).
* **TC\_17:** Test with invalid or unexpected responses from service [X] (e.g., empty response, malformed data).
* **TC\_18:** Test with high volume of concurrent requests to service [X].
* **TC\_19:** Test with different user roles and access permissions.
* **TC\_20:** Test with different browser types and versions.
* **TC\_21:** Test with different device types (desktops, laptops, mobile devices).
* **TC\_21:** Test with various formats of registration numbers (e.g., with/without spaces, hyphens)

**Added Test Cases:**

* **TC\_22:** Verify system behavior when no registration number is entered.
* **TC\_23:** Verify system behavior when special characters are entered in the registration number field.
* **TC\_24:** Verify system behavior when the user leaves the registration number field empty.
* **TC\_25:** Verify system behavior when the user enters a registration number that is too long or too short.
* **TC\_26:** Verify system behavior when the user enters a registration number in a different language or format.
* **TC\_27:** Verify system behavior when the user attempts to bypass or manipulate the registration number field.
* **TC\_28:** Verify system behavior when the user closes the browser or navigates away during the verification process.

**Test Runs**

* **Test Run 1:** Smoke Test - Execute a subset of critical test cases to ensure basic functionality.
* **Test Run 2:** Functional Test - Execute all acceptance and edge case test cases.
* **Test Run 3:** Integration Test - Test the integration with service [X] in a realistic environment.
* **Test Run 4:** Performance Test - Test system performance under high load conditions.
* **Test Run 5:** Regression Test - Re-execute critical test cases after code changes or bug fixes.
* **Test Run 6:** Usability Test - Involve users to assess the ease of use and user experience of the registration verification process.