

# Test Cases Report - 65 Test Cases

Total Test Cases	65
Categories	Integration, Usability, Boundary, Performance, Security, Functional, Error Handling
Priority Levels	Medium, Low, Critical, High
Generated On	2025-08-01 19:09:55

## Test Case 1: Successful Onboarding with Role-based Entitlements

Test ID	TC_001
Module	Corporate Module > Onboarding
Category	Functional
Priority	Critical
Test Type	Positive
Risk Level	High
Estimated Time	10 minutes
Description	Verify that a new user can be onboarded and assigned correct role-based entitlements for Governmental Payments
Objective	Ensure onboarding process assigns entitlements as per user role.
Preconditions	Admin is logged into the portal; Required roles and entitlements are configured
Test Steps	Navigate to onboarding page; Enter valid user details; Select Governmental Payments and sub-products; Assign
Expected Result	User is successfully onboarded with correct entitlements for all selected sub-products.
Validation Criteria	User record created; Entitlements mapped as per role
Dependencies	Role and entitlement configuration
Notes	Covers main onboarding and entitlement mapping flow.

## Test Case 2: Auto-rejection of Pending Transactions after 45 Days

Test ID	TC_002
Module	Workflow > Transaction Approval
Category	Functional
Priority	High
Test Type	Positive
Risk Level	Medium

<b>Estimated Time</b>	5 minutes (simulated time)
<b>Description</b>	Verify that transactions not approved or released within 45 days are automatically rejected.
<b>Objective</b>	Ensure workflow auto-rejection rule is enforced.
<b>Preconditions</b>	Transaction is pending approval for 45 days
<b>Test Steps</b>	Create a transaction; Leave it pending for 45 days; Check transaction status after 45 days
<b>Expected Result</b>	Transaction is automatically rejected after 45 days.
<b>Validation Criteria</b>	Status changes to Rejected; Audit log updated
<b>Dependencies</b>	Transaction creation; Time manipulation capability
<b>Notes</b>	Simulate time passage for testing.

### Test Case 3: Data Type Enforcement for Amount Field

<b>Test ID</b>	TC_003
<b>Module</b>	Corporate Module > Payment Entry
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Test that the Amount field only accepts decimal values with two digits as per AMT (XX, XX) definition.
<b>Objective</b>	Ensure strict data type validation for financial fields.
<b>Preconditions</b>	User is on payment entry form
<b>Test Steps</b>	Enter alphabetic characters in Amount field; Attempt to submit form
<b>Expected Result</b>	System displays validation error and prevents submission.
<b>Validation Criteria</b>	Error message displayed; No submission allowed
<b>Dependencies</b>	Payment entry form accessible
<b>Notes</b>	Critical for financial compliance.

### Test Case 4: Dropdown List Population for Sub-products

<b>Test ID</b>	TC_004
<b>Module</b>	Corporate Module > Entitlement Assignment
<b>Category</b>	Usability
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	2 minutes

Description	Verify that the sub-product dropdown lists all available governmental payment sub-types (Taxes, Customs, Bills).
Objective	Ensure UI displays all sub-products for entitlement assignment.
Preconditions	User is assigning entitlements
Test Steps	Navigate to entitlement assignment; Open sub-product dropdown
Expected Result	Dropdown lists Taxes, Customs, and Bills.
Validation Criteria	All sub-types visible; No missing options
Dependencies	Sub-products configured in system
Notes	Ensures completeness of UI options.

### Test Case 5: Admin Portal Beneficiary Addition for All Sub-types

Test ID	TC_005
Module	Admin Portal > Beneficiary Management
Category	Functional
Priority	Critical
Test Type	Positive
Risk Level	High
Estimated Time	7 minutes
Description	Test that admin can add beneficiaries for Taxes, Customs, and Bills from the admin portal.
Objective	Validate admin portal matches customer portal capabilities.
Preconditions	Admin logged in; No existing beneficiaries for sub-types
Test Steps	Navigate to beneficiary management; Select each sub-type; Add new beneficiary; Save changes
Expected Result	Beneficiaries are added for all sub-types without error.
Validation Criteria	Beneficiaries visible in list; No errors on save
Dependencies	Admin portal access; Sub-types enabled
Notes	Addresses compliance and operational risk.

### Test Case 6: SWIFT Compliance for Free Format Fields

Test ID	TC_006
Module	Corporate Module > Payment Entry
Category	Security
Priority	High
Test Type	Negative
Risk Level	High
Estimated Time	3 minutes
Description	Check that free format fields only accept SWIFT-compliant characters.

<b>Objective</b>	Ensure SWIFT compliance for relevant fields.
<b>Preconditions</b>	User is on payment entry form
<b>Test Steps</b>	Enter non-SWIFT characters in free format field; Attempt to submit
<b>Expected Result</b>	Validation error for non-compliant characters.
<b>Validation Criteria</b>	Only SWIFT characters accepted; Error for invalid input
<b>Dependencies</b>	Field configured for SWIFT compliance
<b>Notes</b>	Important for regulatory compliance.

## Test Case 7: Boundary Test for Fixed Length Fields

<b>Test ID</b>	TC_007
<b>Module</b>	Corporate Module > Data Entry
<b>Category</b>	Boundary
<b>Priority</b>	Medium
<b>Test Type</b>	Boundary
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Test minimum and maximum allowed characters for a fixed length alphanumeric field.
<b>Objective</b>	Validate enforcement of fixed length constraints.
<b>Preconditions</b>	User is on data entry form
<b>Test Steps</b>	Enter fewer than required characters; Attempt to submit; Enter more than allowed characters; Attempt to submit
<b>Expected Result</b>	Validation error for both underflow and overflow.
<b>Validation Criteria</b>	Error for under/overflow; No data saved
<b>Dependencies</b>	Field length defined in configuration
<b>Notes</b>	Prevents data integrity issues.

## Test Case 8: Unauthorized User Cannot Assign Entitlements

<b>Test ID</b>	TC_008
<b>Module</b>	Corporate Module > Entitlement Assignment
<b>Category</b>	Security
<b>Priority</b>	Critical
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	2 minutes
<b>Description</b>	Ensure users without admin privileges cannot assign entitlements.
<b>Objective</b>	Validate authorization controls for sensitive actions.

<b>Preconditions</b>	User logged in with no admin rights
<b>Test Steps</b>	Navigate to entitlement assignment; Attempt to assign entitlements
<b>Expected Result</b>	Access denied or action blocked.
<b>Validation Criteria</b>	No entitlement changes possible; Access denied message
<b>Dependencies</b>	User roles defined
<b>Notes</b>	Covers security risk of privilege escalation.

## Test Case 9: Naming Consistency in UI and API Responses

<b>Test ID</b>	TC_009
<b>Module</b>	Corporate Module > UI/API Integration
<b>Category</b>	Integration
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Verify that 'Governmental Payments' is consistently used in UI and API, not 'Bill Payments'.
<b>Objective</b>	Ensure naming consistency to avoid confusion.
<b>Preconditions</b>	System has both UI and API endpoints for payment modules
<b>Test Steps</b>	Access UI for payment modules; Fetch payment modules via API; Compare naming
<b>Expected Result</b>	'Governmental Payments' used consistently.
<b>Validation Criteria</b>	No naming mismatches; Consistent terminology
<b>Dependencies</b>	API and UI deployed
<b>Notes</b>	Reduces user and developer confusion.

## Test Case 10: Performance Under Bulk Entitlement Assignment

<b>Test ID</b>	TC_010
<b>Module</b>	Corporate Module > Entitlement Assignment
<b>Category</b>	Performance
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	15 minutes
<b>Description</b>	Test system performance when assigning entitlements to 1000 users in bulk.
<b>Objective</b>	Ensure system scales for bulk operations.
<b>Preconditions</b>	Bulk user data prepared

Test Steps	Initiate bulk entitlement assignment; Monitor system response and completion time
Expected Result	All entitlements assigned within acceptable time frame.
Validation Criteria	Completion within SLA; No errors or timeouts
Dependencies	Bulk assignment feature enabled
Notes	Validates scalability for large organizations.

### Test Case 11: Error Handling for Invalid Date Format

Test ID	TC_011
Module	Corporate Module > Data Entry
Category	Error Handling
Priority	Medium
Test Type	Negative
Risk Level	Medium
Estimated Time	2 minutes
Description	Check system response when date is entered in an incorrect format (e.g., MM/DD/YYYY instead of DD-MM-YYYY)
Objective	Ensure robust error handling for date fields.
Preconditions	User is on a form with date input
Test Steps	Enter date as '12/31/2024'; Attempt to submit
Expected Result	Validation error for incorrect date format.
Validation Criteria	Error message shown; No data saved
Dependencies	Date field present
Notes	Prevents downstream data issues.

### Test Case 12: Verifier and Releaser Intervention Workflow

Test ID	TC_012
Module	Corporate Module > Workflow
Category	Functional
Priority	High
Test Type	Positive
Risk Level	Medium
Estimated Time	6 minutes
Description	Ensure that transactions requiring verifier and releaser intervention follow the correct workflow.
Objective	Validate multi-step approval process.
Preconditions	Transaction created requiring both verifier and releaser
Test Steps	Submit transaction; Verifier approves; Releaser approves

<b>Expected Result</b>	Transaction moves to completed status after both approvals.
<b>Validation Criteria</b>	Correct status transitions; Audit logs updated
<b>Dependencies</b>	Workflow configuration
<b>Notes</b>	Ensures compliance with approval policies.

## Test Case 13: Attempt to Add Duplicate Beneficiary

<b>Test ID</b>	TC_013
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Negative
<b>Risk Level</b>	Low
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Test system behavior when admin tries to add a beneficiary that already exists.
<b>Objective</b>	Ensure duplicate prevention in beneficiary management.
<b>Preconditions</b>	Beneficiary already exists for sub-type
<b>Test Steps</b>	Attempt to add beneficiary with same details; Submit form
<b>Expected Result</b>	System prevents duplicate and displays error.
<b>Validation Criteria</b>	Duplicate not allowed; Error message shown
<b>Dependencies</b>	Existing beneficiary data
<b>Notes</b>	Prevents operational errors.

## Test Case 14: Adhoc Bill Parameter Default Value Check

<b>Test ID</b>	TC_014
<b>Module</b>	Corporate Module > Entitlement Parameters
<b>Category</b>	Functional
<b>Priority</b>	Low
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	2 minutes
<b>Description</b>	Ensure that the Adhoc Bill parameter defaults to the specified value during entitlement assignment.
<b>Objective</b>	Validate parameter defaulting logic.
<b>Preconditions</b>	Parameter configuration available
<b>Test Steps</b>	Start entitlement assignment; Check Adhoc Bill parameter value
<b>Expected Result</b>	Adhoc Bill parameter displays correct default.

Validation Criteria	Default value matches config
Dependencies	Parameter configuration
Notes	Ensures correct initialization of entitlement parameters.

### Test Case 15: Scalability for Adding New Payment Types Post Go-live

Test ID	TC_015
Module	Admin Portal > Payment Type Management
Category	Integration
Priority	Medium
Test Type	Positive
Risk Level	Medium
Estimated Time	5 minutes
Description	Test the process and system behavior when admin adds a new governmental payment type after go-live.
Objective	Validate system extensibility for future payment types.
Preconditions	Admin logged in; No existing payment type with same name
Test Steps	Navigate to payment type management; Add new payment type; Save and verify availability
Expected Result	New payment type is added and available for entitlement assignment.
Validation Criteria	Type added without error; Visible in entitlement dropdowns
Dependencies	Admin portal access; No naming conflicts
Notes	Addresses future scalability risk.

### Test Case 16: Verify Admin Can Add Beneficiary for All Governmental Payment Sub-Types

Test ID	TC_016
Module	Admin Portal > Beneficiary Management
Category	Functional
Priority	Critical
Test Type	Positive
Risk Level	High
Estimated Time	10 minutes
Description	Test that an admin user can add a beneficiary for Taxes, Customs, and Bills sub-types under Governmental Payments.
Objective	Ensure admin portal supports beneficiary addition for all sub-types, matching customer portal capabilities.
Preconditions	Admin user is logged in; All sub-types are enabled in the system
Test Steps	Navigate to Beneficiary Management; Select each sub-type (Taxes, Customs, Bills); Attempt to add a new beneficiary.



<b>Expected Result</b>	Beneficiary is successfully added for each sub-type without errors.
<b>Validation Criteria</b>	Beneficiary appears in list; No error messages
<b>Dependencies</b>	Sub-types must be configured; Admin user role assigned
<b>Notes</b>	Addresses compliance and operational risk.

## Test Case 17: Enforce SWIFT Character Compliance in Free Format Fields

<b>Test ID</b>	TC_017
<b>Module</b>	Corporate Module > Data Entry
<b>Category</b>	Security
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Attempt to input invalid (non-SWIFT) characters in a field requiring SWIFT compliance.
<b>Objective</b>	Validate that only SWIFT-compliant characters are accepted.
<b>Preconditions</b>	User is on data entry form with SWIFT-compliant field
<b>Test Steps</b>	Enter invalid characters (e.g., #, \$, %, *); Submit the form
<b>Expected Result</b>	System rejects input and displays appropriate validation error.
<b>Validation Criteria</b>	Error message shown; No data saved
<b>Dependencies</b>	Field configured for SWIFT compliance
<b>Notes</b>	Prevents data corruption and compliance breaches.

## Test Case 18: Validate Auto-Rejection of Pending Transactions After 45 Days

<b>Test ID</b>	TC_018
<b>Module</b>	Workflow Engine > Transaction Approval
<b>Category</b>	Functional
<b>Priority</b>	Critical
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	10 minutes
<b>Description</b>	Create a transaction and leave it unapproved for 45 days to test auto-rejection.
<b>Objective</b>	Ensure system auto-rejects transactions not approved/released within 45 days.
<b>Preconditions</b>	Transaction created and pending approval

Test Steps	Create transaction; Simulate passage of 45 days; Check transaction status
Expected Result	Transaction is auto-rejected after 45 days.
Validation Criteria	Status updated to rejected; Audit log entry created
Dependencies	Time simulation capability
Notes	Covers workflow rigidity and audit compliance.

### Test Case 19: API Endpoint Security: Prevent Unauthorized Entitlement Assignment

Test ID	TC_019
Module	API > Entitlement Management
Category	Security
Priority	High
Test Type	Negative
Risk Level	Critical
Estimated Time	5 minutes
Description	Attempt to assign entitlements using API with insufficient privileges.
Objective	Verify that only authorized users can assign entitlements via API.
Preconditions	API endpoint is accessible; User token with insufficient privileges
Test Steps	Call entitlement assignment API with low-privilege token; Observe response
Expected Result	API returns 403 Forbidden or equivalent error.
Validation Criteria	No entitlement assigned; Proper error code returned
Dependencies	API authentication implemented
Notes	Prevents privilege escalation.

### Test Case 20: Boundary Test: Maximum Length for Fixed-Length Numeric Field

Test ID	TC_020
Module	Corporate Module > Data Validation
Category	Boundary
Priority	Medium
Test Type	Boundary
Risk Level	Medium
Estimated Time	6 minutes
Description	Test system behavior when entering data at, below, and above the maximum allowed length for a fixed-length numeric field.
Objective	Ensure strict enforcement of field length constraints.

<b>Preconditions</b>	User is on data entry form with fixed-length numeric field
<b>Test Steps</b>	Enter value at max length; Enter value below max length; Enter value above max length
<b>Expected Result</b>	System accepts values at or below max length, rejects above max length.
<b>Validation Criteria</b>	Accepted/rejected as per length
<b>Dependencies</b>	Field length defined in schema
<b>Notes</b>	Prevents data integrity issues.

## Test Case 21: Integration Test: Onboarding Entitlement Mapping

<b>Test ID</b>	TC_021
<b>Module</b>	Onboarding > Entitlement Assignment
<b>Category</b>	Integration
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	8 minutes
<b>Description</b>	Verify that a newly onboarded user is correctly mapped to entitled payment modules and sub-products.
<b>Objective</b>	Ensure onboarding process assigns correct entitlements based on role.
<b>Preconditions</b>	Role-based entitlement rules configured
<b>Test Steps</b>	Onboard new user with specific role; Check assigned entitlements
<b>Expected Result</b>	User has access to correct payment modules and sub-products.
<b>Validation Criteria</b>	Entitlements visible in user profile; Access matches role
<b>Dependencies</b>	Entitlement mapping logic implemented
<b>Notes</b>	Covers onboarding and entitlement risk.

## Test Case 22: Usability: Consistent Naming of Governmental Payments in UI

<b>Test ID</b>	TC_022
<b>Module</b>	UI/UX > Navigation
<b>Category</b>	Usability
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Check that 'Governmental Payments' is used consistently as the parent category in all UI elements.
<b>Objective</b>	Reduce user confusion and ensure naming consistency.

<b>Preconditions</b>	UI deployed with all modules
<b>Test Steps</b>	Navigate through menus and forms; Observe category names
<b>Expected Result</b>	'Governmental Payments' is consistently used; no alternate terms like 'Bill Payments'.
<b>Validation Criteria</b>	No inconsistent naming found
<b>Dependencies</b>	UI elements loaded
<b>Notes</b>	Addresses business risk of user confusion.

## Test Case 23: Error Handling: Invalid Date Format in Transaction Entry

<b>Test ID</b>	TC_023
<b>Module</b>	Corporate Module > Transaction Entry
<b>Category</b>	Error Handling
<b>Priority</b>	Medium
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Attempt to enter a date in an incorrect format (e.g., MM-DD-YYYY instead of DD-MM-YYYY).
<b>Objective</b>	Ensure system validates and rejects invalid date formats.
<b>Preconditions</b>	User is on transaction entry form
<b>Test Steps</b>	Enter date as 12-31-2025; Submit form
<b>Expected Result</b>	System displays error and does not accept the date.
<b>Validation Criteria</b>	Error message displayed; No transaction created
<b>Dependencies</b>	Date validation logic implemented
<b>Notes</b>	Prevents data entry errors.

## Test Case 24: Performance: File Upload Response Time Under Load

<b>Test ID</b>	TC_024
<b>Module</b>	Corporate Module > File Upload
<b>Category</b>	Performance
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	12 minutes
<b>Description</b>	Measure response time when uploading a large beneficiary file during peak usage.

<b>Objective</b>	Ensure file upload performs within acceptable limits under load.
<b>Preconditions</b>	System under simulated load; Large file prepared
<b>Test Steps</b>	Initiate file upload; Record time to completion
<b>Expected Result</b>	File uploads within defined SLA (e.g., <10 seconds).
<b>Validation Criteria</b>	Upload time within SLA; No errors
<b>Dependencies</b>	Load testing tools; SLA defined
<b>Notes</b>	Critical for operational efficiency.

## Test Case 25: Security: File Upload Virus Scan Enforcement

<b>Test ID</b>	TC_025
<b>Module</b>	Corporate Module > File Upload
<b>Category</b>	Security
<b>Priority</b>	Critical
<b>Test Type</b>	Negative
<b>Risk Level</b>	Critical
<b>Estimated Time</b>	7 minutes
<b>Description</b>	Attempt to upload a file containing known malware signature.
<b>Objective</b>	Ensure system scans uploaded files and blocks infected files.
<b>Preconditions</b>	Virus scanning enabled on file upload
<b>Test Steps</b>	Prepare file with EICAR test string; Upload file; Observe system response
<b>Expected Result</b>	File is rejected and user receives security alert.
<b>Validation Criteria</b>	File not accepted; Alert generated
<b>Dependencies</b>	Virus scanning service integrated
<b>Notes</b>	Mitigates malware risk.

## Test Case 26: API: Validate Entitlement Parameter Defaults

<b>Test ID</b>	TC_026
<b>Module</b>	API > Entitlement Parameters
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Call API to fetch entitlement parameters and verify default values for each sub-product.
<b>Objective</b>	Ensure API returns correct default values for entitlement parameters.

<b>Preconditions</b>	API endpoint available; Entitlement parameters configured
<b>Test Steps</b>	Call entitlement parameters API; Check returned values
<b>Expected Result</b>	Default values match documentation for each sub-product.
<b>Validation Criteria</b>	Returned values match expected defaults
<b>Dependencies</b>	API and parameters configured
<b>Notes</b>	Ensures configuration integrity.

## Test Case 27: Negative Test: Upload Unsupported File Type

<b>Test ID</b>	TC_027
<b>Module</b>	Corporate Module > File Upload
<b>Category</b>	Error Handling
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Attempt to upload a file with an unsupported extension (e.g., .exe).
<b>Objective</b>	Ensure system rejects unsupported file types.
<b>Preconditions</b>	File upload feature enabled
<b>Test Steps</b>	Select .exe file; Attempt upload
<b>Expected Result</b>	System rejects file and displays error message.
<b>Validation Criteria</b>	Error message shown; File not uploaded
<b>Dependencies</b>	File type whitelist configured
<b>Notes</b>	Prevents security and operational issues.

## Test Case 28: Boundary Test: Maximum Daily Single Limit Value

<b>Test ID</b>	TC_028
<b>Module</b>	Corporate Module > Entitlement Parameters
<b>Category</b>	Boundary
<b>Priority</b>	Medium
<b>Test Type</b>	Boundary
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Test system behavior when setting the daily single limit to its maximum allowed value.
<b>Objective</b>	Ensure system accepts maximum allowed value and rejects values above it.
<b>Preconditions</b>	Entitlement parameter configuration access

<b>Test Steps</b>	Set daily single limit to max value; Attempt to set above max value
<b>Expected Result</b>	Max value accepted; above max value rejected.
<b>Validation Criteria</b>	Accepted/rejected as per rule
<b>Dependencies</b>	Max value defined in system
<b>Notes</b>	Prevents financial risk.

## Test Case 29: Integration: Add New Governmental Payment Type Post Go-Live

<b>Test ID</b>	TC_029
<b>Module</b>	Admin Portal > Payment Type Management
<b>Category</b>	Integration
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	9 minutes
<b>Description</b>	Test admin's ability to add a new governmental payment type after system go-live.
<b>Objective</b>	Validate system's scalability and configuration-driven management.
<b>Preconditions</b>	System is live; Admin access granted
<b>Test Steps</b>	Navigate to Payment Type Management; Add new payment type; Verify appearance in UI and API
<b>Expected Result</b>	New payment type is available for entitlements and transactions.
<b>Validation Criteria</b>	Type visible in UI/API; Can assign entitlements
<b>Dependencies</b>	Dynamic payment type management implemented
<b>Notes</b>	Addresses scalability risk.

## Test Case 30: Usability: Dropdown and Checkbox UI Accessibility

<b>Test ID</b>	TC_030
<b>Module</b>	UI/UX > Form Controls
<b>Category</b>	Usability
<b>Priority</b>	Low
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Test accessibility of dropdown lists and checkboxes for entitlement parameters.
<b>Objective</b>	Ensure all users, including those using assistive technologies, can interact with form controls.
<b>Preconditions</b>	Form with dropdowns and checkboxes loaded

Test Steps	Navigate using keyboard; Use screen reader; Select options
Expected Result	All controls are accessible and usable.
Validation Criteria	Keyboard navigation; Screen reader output
Dependencies	Accessibility features enabled
Notes	Improves inclusivity.

### Test Case 31: Verify Admin Can Add Beneficiary for All Sub-Types

Test ID	TC_031
Module	Admin Portal > Beneficiary Management
Category	Functional
Priority	Critical
Test Type	Positive
Risk Level	High
Estimated Time	10 minutes
Description	Test if admin can add beneficiaries for all governmental payment sub-types (Taxes, Customs, Bills) via the admin portal.
Objective	Ensure admin portal matches customer portal capabilities for beneficiary management.
Preconditions	Admin user logged in; All sub-types enabled
Test Steps	Navigate to Beneficiary Management in Admin Portal; Select each sub-type (Taxes, Customs, Bills); Attempt to add beneficiary.
Expected Result	Admin can successfully add beneficiaries for all sub-types without errors.
Validation Criteria	Beneficiaries visible under each sub-type; No errors during addition
Dependencies	Sub-types configured; Admin access
Notes	Addresses compliance and operational risk.

### Test Case 32: Auto-Rejection After 45 Days - Workflow Expiry

Test ID	TC_032
Module	Workflow Engine > Transaction Approval
Category	Functional
Priority	High
Test Type	Positive
Risk Level	Medium
Estimated Time	8 minutes
Description	Verify that transactions not approved or released within 45 days are auto-rejected.
Objective	Validate strict workflow rule for compliance and audit.
Preconditions	Transaction pending approval; System time manipulation allowed
Test Steps	Initiate a transaction; Do not approve or release; Advance system date by 45 days; Check transaction status



<b>Expected Result</b>	Transaction is auto-rejected after 45 days.
<b>Validation Criteria</b>	Status changes to auto-rejected; Audit log updated
<b>Dependencies</b>	Time manipulation capability
<b>Notes</b>	Critical for regulatory compliance.

## Test Case 33: Field Validation - SWIFT Compliance Characters

<b>Test ID</b>	TC_033
<b>Module</b>	Data Entry > Field Validation
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Test that only SWIFT-compliant characters are accepted in designated fields.
<b>Objective</b>	Ensure data validation and compliance.
<b>Preconditions</b>	User logged in; Field with SWIFT compliance requirement available
<b>Test Steps</b>	Navigate to form with SWIFT-compliant field; Enter valid SWIFT characters; Submit form
<b>Expected Result</b>	Form is accepted and data saved.
<b>Validation Criteria</b>	No validation errors; Data stored as entered
<b>Dependencies</b>	Field configured for SWIFT compliance
<b>Notes</b>	Covers compliance and data integrity.

## Test Case 34: Field Validation - Reject Non-SWIFT Characters

<b>Test ID</b>	TC_034
<b>Module</b>	Data Entry > Field Validation
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Ensure that non-SWIFT-compliant characters are rejected in fields requiring SWIFT compliance.
<b>Objective</b>	Prevent invalid data entry and enforce compliance.
<b>Preconditions</b>	User logged in; Field with SWIFT compliance requirement available
<b>Test Steps</b>	Navigate to form with SWIFT-compliant field; Enter invalid characters (e.g., #, \$, %, ^, *); Submit form
<b>Expected Result</b>	Validation error displayed; form not submitted.

<b>Validation Criteria</b>	Error message shown; No data saved
<b>Dependencies</b>	Field configured for SWIFT compliance
<b>Notes</b>	Addresses data validation risk.

## Test Case 35: Boundary Test - Maximum Field Length Enforcement

<b>Test ID</b>	TC_035
<b>Module</b>	Data Entry > Field Validation
<b>Category</b>	Boundary
<b>Priority</b>	Medium
<b>Test Type</b>	Boundary
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Test that fields with fixed length do not accept input beyond the specified maximum.
<b>Objective</b>	Verify enforcement of field length constraints.
<b>Preconditions</b>	User logged in; Field with fixed length configured
<b>Test Steps</b>	Navigate to fixed length field; Enter maximum allowed characters; Attempt to enter one additional character
<b>Expected Result</b>	Additional character is not accepted; field enforces length limit.
<b>Validation Criteria</b>	No overflow; Error or block on extra input
<b>Dependencies</b>	Field length defined in configuration
<b>Notes</b>	Prevents data truncation and errors.

## Test Case 36: Integration - Entitlement Assignment During Onboarding

<b>Test ID</b>	TC_036
<b>Module</b>	Onboarding > Entitlement Management
<b>Category</b>	Integration
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	7 minutes
<b>Description</b>	Verify that entitlements to payment modules and sub-products are correctly assigned during user onboarding.
<b>Objective</b>	Ensure integration between onboarding and entitlement modules.
<b>Preconditions</b>	New user onboarding initiated
<b>Test Steps</b>	Complete onboarding steps; Assign entitlements to user; Verify assigned entitlements in user profile

<b>Expected Result</b>	User has correct entitlements for selected modules and sub-products.
<b>Validation Criteria</b>	Correct entitlements in user profile; No missing/extra entitlements
<b>Dependencies</b>	Entitlement module available
<b>Notes</b>	Covers critical onboarding workflow.

## Test Case 37: Integration - Admin Adds New Governmental Payment Type

<b>Test ID</b>	TC_037
<b>Module</b>	Admin Portal > Product Management
<b>Category</b>	Integration
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Test the ability to add a new governmental payment type post go-live via admin portal.
<b>Objective</b>	Validate system scalability and dynamic configuration.
<b>Preconditions</b>	Admin user logged in; Access to product management
<b>Test Steps</b>	Navigate to Product Management; Add new governmental payment type; Verify new type appears in selection list
<b>Expected Result</b>	New payment type is available for entitlement and transaction.
<b>Validation Criteria</b>	Type visible in UI; Type selectable in entitlements
<b>Dependencies</b>	Dynamic configuration supported
<b>Notes</b>	Tests future scalability.

## Test Case 38: Performance - Bulk Beneficiary Addition

<b>Test ID</b>	TC_038
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Performance
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	12 minutes
<b>Description</b>	Assess system performance when adding a large number of beneficiaries in quick succession.
<b>Objective</b>	Identify performance bottlenecks in beneficiary management.
<b>Preconditions</b>	Admin user logged in; Bulk data available
<b>Test Steps</b>	Prepare bulk beneficiary data; Initiate bulk addition; Monitor system response and completion time

<b>Expected Result</b>	System processes all additions without timeouts or errors.
<b>Validation Criteria</b>	No errors; Acceptable response time
<b>Dependencies</b>	Bulk addition feature enabled
<b>Notes</b>	Ensures operational efficiency at scale.

## Test Case 39: Error Handling - Invalid Data Type Entry

<b>Test ID</b>	TC_039
<b>Module</b>	Data Entry > Field Validation
<b>Category</b>	Error Handling
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Test system response when entering alphabetic characters in a numeric-only field.
<b>Objective</b>	Ensure robust error handling and data validation.
<b>Preconditions</b>	User logged in; Numeric field available
<b>Test Steps</b>	Navigate to numeric-only field; Enter alphabetic characters; Attempt to submit form
<b>Expected Result</b>	Validation error displayed; form not submitted.
<b>Validation Criteria</b>	Error message shown; No data saved
<b>Dependencies</b>	Field data type enforced
<b>Notes</b>	Prevents data corruption.

## Test Case 40: Security - Unauthorized Beneficiary Addition Attempt

<b>Test ID</b>	TC_040
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Security
<b>Priority</b>	Critical
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Verify that users without admin rights cannot add beneficiaries via the admin portal.
<b>Objective</b>	Ensure proper access control and security.
<b>Preconditions</b>	Non-admin user logged in
<b>Test Steps</b>	Navigate to Beneficiary Management; Attempt to add a beneficiary; Observe system response

<b>Expected Result</b>	Access denied; action not permitted.
<b>Validation Criteria</b>	No beneficiary added; Access denied message shown
<b>Dependencies</b>	RBAC implemented
<b>Notes</b>	Addresses compliance and security risk.

## Test Case 41: Usability - Dropdown List Default Value

<b>Test ID</b>	TC_041
<b>Module</b>	UI/UX > Field Controls
<b>Category</b>	Usability
<b>Priority</b>	Low
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Verify that dropdown lists display the correct default value as specified in configuration.
<b>Objective</b>	Ensure consistency and usability in UI controls.
<b>Preconditions</b>	User logged in; Dropdown field present
<b>Test Steps</b>	Navigate to form with dropdown list; Observe default value; Compare with configuration
<b>Expected Result</b>	Dropdown displays correct default value.
<b>Validation Criteria</b>	Default matches config; No blank/default errors
<b>Dependencies</b>	Configuration file available
<b>Notes</b>	Improves user experience.

## Test Case 42: Error Handling - Missing Mandatory Field Submission

<b>Test ID</b>	TC_042
<b>Module</b>	Data Entry > Field Validation
<b>Category</b>	Error Handling
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Test system response when mandatory fields are left blank during form submission.
<b>Objective</b>	Ensure mandatory field validation is enforced.
<b>Preconditions</b>	User logged in; Form with mandatory fields available
<b>Test Steps</b>	Navigate to form; Leave one or more mandatory fields blank; Attempt to submit form

Expected Result	Form submission blocked; error message displayed.
Validation Criteria	Error shown; No incomplete data saved
Dependencies	Mandatory field attribute set
Notes	Prevents incomplete data entry.

### Test Case 43: Performance - Transaction Approval Under Load

Test ID	TC_043
Module	Workflow Engine > Transaction Approval
Category	Performance
Priority	Medium
Test Type	Positive
Risk Level	Medium
Estimated Time	10 minutes
Description	Evaluate system performance when multiple transactions are approved simultaneously.
Objective	Identify performance issues under heavy workflow load.
Preconditions	Multiple pending transactions; Approver logged in
Test Steps	Initiate approval of multiple transactions in parallel; Monitor system response time; Check for any failures or delays
Expected Result	All transactions approved without significant delay or error.
Validation Criteria	Response time < threshold; No errors
Dependencies	Bulk approval feature enabled
Notes	Ensures workflow scalability.

### Test Case 44: Integration - Entitlement Revocation Propagation

Test ID	TC_044
Module	Entitlement Management > Entitlement Revocation
Category	Integration
Priority	High
Test Type	Positive
Risk Level	High
Estimated Time	6 minutes
Description	Verify that revoking a user's entitlement removes access to related modules and sub-products immediately.
Objective	Ensure entitlement changes propagate across integrated modules.
Preconditions	User with entitlements assigned; Admin logged in
Test Steps	Revoke entitlement from user; Attempt to access related modules as user; Observe access status
Expected Result	User loses access to revoked modules/sub-products instantly.

Validation Criteria	Access denied post-revocation; No delay in propagation
Dependencies	Entitlement management integrated
Notes	Prevents unauthorized access.

### Test Case 45: Usability - Consistent Naming in UI

Test ID	TC_045
Module	UI/UX > Naming Conventions
Category	Usability
Priority	Medium
Test Type	Positive
Risk Level	Medium
Estimated Time	5 minutes
Description	Ensure that 'Governmental Payments' is used consistently as the parent category in all UI elements.
Objective	Reduce user confusion and ensure documentation alignment.
Preconditions	User logged in; Multiple UI screens available
Test Steps	Navigate through all relevant UI screens; Check naming of parent payment category; Compare with documentation
Expected Result	'Governmental Payments' consistently used in all locations.
Validation Criteria	No inconsistent names; Matches documentation
Dependencies	UI screens implemented
Notes	Improves user training and reduces errors.

### Test Case 46: Verify mandatory field validation for Governmental Payment onboarding

Test ID	TC_046
Module	Corporate Module > Onboarding
Category	Functional
Priority	Critical
Test Type	Negative
Risk Level	High
Estimated Time	5 minutes
Description	Test that all mandatory fields for Governmental Payment onboarding enforce correct data types and cannot be left empty.
Objective	Ensure mandatory fields are validated and enforced as per data type requirements.
Preconditions	User has access to onboarding form; No fields are pre-filled
Test Steps	Navigate to onboarding form; Leave all mandatory fields empty; Attempt to submit the form
Expected Result	Form submission is blocked and appropriate error messages are displayed for each empty mandatory field.

<b>Validation Criteria</b>	All mandatory fields must be enforced; Error messages are user-friendly
<b>Dependencies</b>	UI form rendering; Field validation logic
<b>Notes</b>	Covers both UI and backend validation.

## Test Case 47: Check admin ability to add beneficiary for all sub-types

<b>Test ID</b>	TC_047
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	7 minutes
<b>Description</b>	Test that admin can add beneficiaries for Taxes, Customs, and Bills under Governmental Payments.
<b>Objective</b>	Validate admin portal matches customer portal in beneficiary management capabilities.
<b>Preconditions</b>	Admin user is logged in; No beneficiaries exist for sub-types
<b>Test Steps</b>	Navigate to beneficiary management; Select each sub-type (Taxes, Customs, Bills); Add a new beneficiary for each
<b>Expected Result</b>	Beneficiaries are successfully added for all sub-types and visible in the list.
<b>Validation Criteria</b>	Beneficiaries can be added for all sub-types; No errors during addition
<b>Dependencies</b>	Admin portal access; Beneficiary database
<b>Notes</b>	Addresses compliance and operational risk.

## Test Case 48: Validate SWIFT compliance for free format fields

<b>Test ID</b>	TC_048
<b>Module</b>	Corporate Module > Data Entry
<b>Category</b>	Security
<b>Priority</b>	Critical
<b>Test Type</b>	Boundary
<b>Risk Level</b>	High
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Test that free format fields only accept SWIFT-compliant characters.
<b>Objective</b>	Ensure SWIFT compliance for designated fields.
<b>Preconditions</b>	User is on form with free format field
<b>Test Steps</b>	Enter valid SWIFT characters in the field; Attempt to submit; Enter invalid characters (e.g., @, #, \$); Attempt to submit
<b>Expected Result</b>	Form accepts SWIFT-compliant input and rejects non-compliant characters with an error.



<b>Validation Criteria</b>	Only SWIFT characters accepted; Clear error for invalid input
<b>Dependencies</b>	Field validation logic
<b>Notes</b>	Ensures financial compliance.

## Test Case 49: Test auto-rejection of pending transactions after 45 days

<b>Test ID</b>	TC_049
<b>Module</b>	Workflow > Transaction Approval
<b>Category</b>	Functional
<b>Priority</b>	Critical
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	N/A (time simulation required)
<b>Description</b>	Verify that transactions not approved or released within 45 days are automatically rejected.
<b>Objective</b>	Ensure workflow rule for auto-rejection is enforced.
<b>Preconditions</b>	Transaction created and pending approval
<b>Test Steps</b>	Create a new transaction; Do not take any action for 45 days; Check transaction status after 45 days
<b>Expected Result</b>	Transaction is automatically rejected after 45 days.
<b>Validation Criteria</b>	Auto-rejection occurs exactly at 45 days; Status updated correctly
<b>Dependencies</b>	Workflow engine; Scheduler
<b>Notes</b>	Simulate time passage for test.

## Test Case 50: Check dropdown list population for entitlement parameters

<b>Test ID</b>	TC_050
<b>Module</b>	Corporate Module > Entitlement Management
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Test that all entitlement parameter dropdowns are populated with correct values as per configuration.
<b>Objective</b>	Ensure dropdowns reflect configuration-driven values.
<b>Preconditions</b>	User has access to entitlement management
<b>Test Steps</b>	Navigate to entitlement management; Open each entitlement parameter dropdown; Verify available options

<b>Expected Result</b>	Dropdowns display all configured options with correct default selected.
<b>Validation Criteria</b>	Dropdowns match configuration; Default values correct
<b>Dependencies</b>	Configuration files; UI rendering
<b>Notes</b>	Covers configuration management.

## Test Case 51: Attempt onboarding with invalid data types

<b>Test ID</b>	TC_051
<b>Module</b>	Corporate Module > Onboarding
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Test that the system rejects onboarding submissions with invalid data types (e.g., letters in numeric fields).
<b>Objective</b>	Validate strict data type enforcement.
<b>Preconditions</b>	User is on onboarding form
<b>Test Steps</b>	Enter letters in a numeric-only field; Enter special characters in an alphanumeric field; Submit the form
<b>Expected Result</b>	Submission is blocked and relevant error messages are displayed.
<b>Validation Criteria</b>	All invalid data types rejected; Clear error messages
<b>Dependencies</b>	Field validation logic
<b>Notes</b>	Prevents data corruption.

## Test Case 52: Verify role-based entitlement mapping during onboarding

<b>Test ID</b>	TC_052
<b>Module</b>	Corporate Module > Entitlement Management
<b>Category</b>	Integration
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Test that users receive correct entitlements based on their assigned roles during onboarding.
<b>Objective</b>	Ensure role-based access control is enforced.
<b>Preconditions</b>	User onboarding process initiated; Roles defined in system
<b>Test Steps</b>	Assign a specific role to new user; Complete onboarding; Check assigned entitlements

Expected Result	User receives only the entitlements mapped to their role.
Validation Criteria	Entitlements match role; No unauthorized entitlements assigned
Dependencies	Role mapping configuration; Entitlement database
Notes	Critical for security and compliance.

### Test Case 53: Test scalability: Add new governmental payment type post go-live

Test ID	TC_053
Module	Admin Portal > Configuration Management
Category	Integration
Priority	Medium
Test Type	Positive
Risk Level	Medium
Estimated Time	8 minutes
Description	Verify that admin can dynamically add a new governmental payment type without code changes.
Objective	Test configuration-driven scalability for payment types.
Preconditions	Admin user logged in; No existing payment type with same name
Test Steps	Navigate to payment type management; Add a new payment type; Verify new type is available in onboarding and
Expected Result	New payment type is added and selectable in relevant modules.
Validation Criteria	New type appears without errors; No system restart required
Dependencies	Configuration backend; UI refresh logic
Notes	Validates future-proofing.

### Test Case 54: Check for inconsistent naming in UI and backend

Test ID	TC_054
Module	Corporate Module > UI/UX
Category	Usability
Priority	Medium
Test Type	Negative
Risk Level	Medium
Estimated Time	10 minutes
Description	Test that 'Governmental Payments' and its sub-types are consistently named across UI and backend.
Objective	Prevent user confusion and integration errors due to naming inconsistencies.
Preconditions	System deployed with at least one payment type
Test Steps	Navigate through all modules referencing payment types; Compare naming in UI and API responses; Check data

<b>Expected Result</b>	Naming is consistent across UI, backend, and database.
<b>Validation Criteria</b>	No mismatches found; Consistent terminology
<b>Dependencies</b>	UI code; API; Database schema
<b>Notes</b>	Addresses business risk.

## Test Case 55: Test error handling for database connection failure during beneficiary addition

<b>Test ID</b>	TC_055
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Error Handling
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Simulate a database failure and verify system displays appropriate error without data loss or crash.
<b>Objective</b>	Ensure graceful error handling for backend failures.
<b>Preconditions</b>	Admin user logged in; Database temporarily unavailable
<b>Test Steps</b>	Attempt to add a new beneficiary; Observe system response; Check for error logs
<b>Expected Result</b>	User sees a clear error message; no partial data is saved.
<b>Validation Criteria</b>	No data corruption; User informed of issue
<b>Dependencies</b>	Database connectivity; Error handling logic
<b>Notes</b>	Critical for operational stability.

## Test Case 56: Performance test: Load entitlement management with 10,000 users

<b>Test ID</b>	TC_056
<b>Module</b>	Corporate Module > Entitlement Management
<b>Category</b>	Performance
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	10 minutes
<b>Description</b>	Test system performance when loading entitlement management page with a large user base.
<b>Objective</b>	Ensure UI and backend can handle high user volume efficiently.
<b>Preconditions</b>	10,000 users with entitlements exist in database

Test Steps	Login as admin; Navigate to entitlement management; Measure page load time
Expected Result	Page loads within acceptable performance threshold (e.g., <5 seconds).
Validation Criteria	Load time <5s; No UI errors
Dependencies	Database size; UI rendering
Notes	Validates scalability.

### Test Case 57: Boundary test: Maximum allowed amount entry for payment

Test ID	TC_057
Module	Corporate Module > Payment Processing
Category	Functional
Priority	High
Test Type	Boundary
Risk Level	Medium
Estimated Time	4 minutes
Description	Test that the system accepts the maximum allowed value for amount fields and rejects values above it.
Objective	Validate boundary conditions for amount entry.
Preconditions	User is on payment form
Test Steps	Enter the maximum allowed amount (e.g., 99.99); Submit the form; Enter an amount above the maximum (e.g., 100.00); Submit the form.
Expected Result	Maximum value is accepted; above maximum is rejected with an error.
Validation Criteria	Boundary values enforced; Clear error for out-of-range
Dependencies	Field validation logic
Notes	Prevents financial errors.

### Test Case 58: Usability test: Add beneficiary flow in admin portal

Test ID	TC_058
Module	Admin Portal > Beneficiary Management
Category	Usability
Priority	Low
Test Type	Positive
Risk Level	Low
Estimated Time	3 minutes
Description	Test the usability of the beneficiary addition process in the admin portal, ensuring clear instructions and minimal steps.
Objective	Ensure the process is user-friendly and efficient.
Preconditions	Admin user logged in

Test Steps	Navigate to beneficiary management; Start add beneficiary process; Follow on-screen instructions to completion
Expected Result	Beneficiary is added with minimal steps and clear guidance.
Validation Criteria	Process is intuitive; No unnecessary steps
Dependencies	UI design; Instructional text
Notes	Improves admin efficiency.

## Test Case 59: Security test: Attempt unauthorized access to entitlement management

Test ID	TC_059
Module	Corporate Module > Entitlement Management
Category	Security
Priority	Critical
Test Type	Negative
Risk Level	High
Estimated Time	4 minutes
Description	Test that users without proper roles cannot access or modify entitlement settings.
Objective	Ensure access control is enforced for sensitive modules.
Preconditions	User without entitlement management role
Test Steps	Login as unauthorized user; Attempt to access entitlement management; Try to modify entitlements
Expected Result	Access is denied and user is redirected or shown an error.
Validation Criteria	No unauthorized access possible; Clear error or redirect
Dependencies	Role-based access control
Notes	Prevents privilege escalation.

## Test Case 60: Integration test: Entitlement update reflects in payment processing

Test ID	TC_060
Module	Corporate Module > Integration
Category	Integration
Priority	High
Test Type	Positive
Risk Level	Medium
Estimated Time	6 minutes
Description	Verify that changes in user entitlements are immediately reflected in payment processing eligibility.
Objective	Ensure real-time integration between entitlement and payment modules.

<b>Preconditions</b>	User has initial entitlement; Entitlement is updated
<b>Test Steps</b>	Update user's entitlement in entitlement management; Attempt payment processing as the user; Verify eligibility r
<b>Expected Result</b>	Payment processing eligibility matches updated entitlements.
<b>Validation Criteria</b>	No delay in update; Eligibility logic correct
<b>Dependencies</b>	Entitlement module; Payment processing module
<b>Notes</b>	Ensures modules are properly integrated.

## Test Case 61: Error handling: Attempt to add duplicate beneficiary

<b>Test ID</b>	TC_061
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Error Handling
<b>Priority</b>	Medium
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Test that the system prevents adding a beneficiary with details identical to an existing one.
<b>Objective</b>	Ensure uniqueness constraint is enforced for beneficiaries.
<b>Preconditions</b>	A beneficiary with given details already exists
<b>Test Steps</b>	Attempt to add a beneficiary with the same details as an existing one; Submit the form
<b>Expected Result</b>	System blocks addition and displays a duplicate entry error.
<b>Validation Criteria</b>	No duplicates allowed; Clear error message
<b>Dependencies</b>	Database uniqueness constraint
<b>Notes</b>	Prevents data integrity issues.

## Test Case 62: Admin Portal - Add Beneficiary for All Governmental Payment Sub-types

<b>Test ID</b>	TC_061
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Functional
<b>Priority</b>	Critical
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	15 minutes
<b>Description</b>	Verify that the admin portal allows adding beneficiaries for all governmental payment sub-types (Taxes, Customs,

<b>Objective</b>	Ensure admin portal matches customer portal capabilities for beneficiary management and meets compliance requirements
<b>Preconditions</b>	Admin user is logged into the admin portal; All governmental payment sub-types are enabled in the system
<b>Test Steps</b>	Navigate to Beneficiary Management in the admin portal; Select each sub-type: Taxes, Customs, Bills; Attempt to add beneficiaries
<b>Expected Result</b>	Admin can successfully add beneficiaries for all sub-types, and beneficiaries are visible in the respective lists.
<b>Validation Criteria</b>	Beneficiary creation succeeds for all sub-types; Beneficiaries are retrievable via admin portal
<b>Dependencies</b>	Beneficiary management module implemented; Sub-types configured in system
<b>Notes</b>	Addresses compliance and operational risk due to incomplete admin portal functionality.

## Test Case 63: Data Validation - SWIFT Compliance for Free Format Fields

<b>Test ID</b>	TC_062
<b>Module</b>	Entitlement Management > Field Validation
<b>Category</b>	Security
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Test that only SWIFT-compliant characters are accepted in fields designated as 'F' (Free format, SWIFT-compliant)
<b>Objective</b>	Prevent data entry errors and ensure compliance with SWIFT requirements.
<b>Preconditions</b>	User is on the entitlement creation/edit screen; At least one field is marked as 'F' (SWIFT-compliant)
<b>Test Steps</b>	Enter a string with invalid characters (e.g., #, \$, %, ^) into the free format field; Attempt to save the record; Observe system response
<b>Expected Result</b>	System rejects input with invalid characters and displays a clear validation error.
<b>Validation Criteria</b>	Invalid input is not accepted; Error message specifies allowed characters
<b>Dependencies</b>	Field validation logic implemented
<b>Notes</b>	SWIFT compliance is critical for financial and regulatory reasons.

## Test Case 64: Workflow Handling - Auto-Rejection After 45 Days

<b>Test ID</b>	TC_063
<b>Module</b>	Workflow Management > Transaction Approval
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Boundary
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	10 minutes
<b>Description</b>	Verify that transactions not approved or released within 45 days are automatically rejected by the system.



Objective	Ensure workflow rules are enforced for operational efficiency and audit compliance.
Preconditions	A transaction is created and left in pending state; System date/time can be manipulated or simulated
Test Steps	Create a new governmental payment transaction; Do not approve or release the transaction; Simulate passage of time
Expected Result	Transaction is automatically rejected after 45 days of inactivity.
Validation Criteria	Transaction status updates correctly; Audit trail reflects auto-rejection
Dependencies	Workflow engine supports time-based triggers
Notes	Tests strict workflow rule and boundary condition.

## Test Case 65: Cross-Browser Compatibility - Entitlement Onboarding UI

Test ID	TC_064
Module	Onboarding > Entitlement Assignment
Category	Usability
Priority	Medium
Test Type	Positive
Risk Level	Medium
Estimated Time	20 minutes
Description	Verify that the entitlement onboarding UI renders and functions correctly across Chrome, Firefox, and Edge.
Objective	Ensure consistent user experience and prevent browser-specific issues.
Preconditions	Entitlement onboarding UI is deployed and accessible
Test Steps	Open the onboarding UI in Chrome; Repeat in Firefox and Edge; Perform entitlement assignment for a new user
Expected Result	UI renders correctly and all onboarding actions succeed in all tested browsers.
Validation Criteria	No rendering issues; All actions are functional in each browser
Dependencies	UI supports multiple browsers
Notes	Addresses cross-browser compatibility focus area.