

# Test Cases Report - 65 Test Cases

Total Test Cases	65
Categories	Performance, Integration, Usability, Boundary, Functional, Error, Security
Priority Levels	Low, High, Critical, Medium
Generated On	2025-07-31 13:33:48

## Test Case 1: Validate assignment of entitlements during customer onboarding

Test ID	TC_001
Module	Entitlements > Customer On-Boarding
Category	Functional
Priority	Critical
Test Type	Positive
Risk Level	High
Estimated Time	10 minutes
Description	Test that entitlements are correctly assigned to customers during onboarding based on selected payment types.
Objective	Ensure entitlement assignment logic works as per business rules during onboarding.
Preconditions	Customer onboarding process initiated; Payment types configured
Test Steps	Start onboarding for a new customer; Select payment types; Complete onboarding
Expected Result	Entitlements are assigned as per selected payment types and displayed in customer profile.
Validation Criteria	Entitlements match selection; No missing entitlements
Dependencies	Payment type configuration
Notes	Covers initial entitlement logic for onboarding.

## Test Case 2: Verify auto-rejection of stale transactions after 45 days

Test ID	TC_002
Module	Common Handling > Workflow Rules
Category	Functional
Priority	Critical
Test Type	Positive

<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Test that unapproved or unreleased transactions are automatically rejected after 45 days.
<b>Objective</b>	Ensure compliance with workflow auto-rejection rule.
<b>Preconditions</b>	Transaction created and pending for 45 days
<b>Test Steps</b>	Create a transaction; Leave it unapproved for 45 days; Check transaction status after 45 days
<b>Expected Result</b>	Transaction is automatically rejected by the system.
<b>Validation Criteria</b>	Status updates to Rejected; Audit log entry present
<b>Dependencies</b>	Transaction creation; Time simulation
<b>Notes</b>	Time manipulation may be needed for test.

### Test Case 3: Test role-based access control for entitlement management

<b>Test ID</b>	TC_003
<b>Module</b>	Security > Authorization
<b>Category</b>	Security
<b>Priority</b>	Critical
<b>Test Type</b>	Positive/Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	8 minutes
<b>Description</b>	Verify that only authorized roles can access and modify entitlement settings.
<b>Objective</b>	Prevent unauthorized access to entitlement management.
<b>Preconditions</b>	User accounts with different roles exist
<b>Test Steps</b>	Login as admin; Access entitlement management; Login as regular user; Attempt to access entitlement management
<b>Expected Result</b>	Admin can access; regular user is denied access.
<b>Validation Criteria</b>	Access control enforced; Proper error message for unauthorized access
<b>Dependencies</b>	User role setup
<b>Notes</b>	Test both positive (admin) and negative (user) scenarios.

### Test Case 4: Validate SWIFT compliance for free-format fields

<b>Test ID</b>	TC_004
<b>Module</b>	Data Validation > Input Validation
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Positive/Negative

<b>Risk Level</b>	High
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Ensure that only SWIFT-compliant characters are accepted in free-format fields.
<b>Objective</b>	Prevent invalid data entry in SWIFT-compliant fields.
<b>Preconditions</b>	SWIFT-compliant field present in UI
<b>Test Steps</b>	Enter valid SWIFT characters; Submit; Enter invalid characters; Submit
<b>Expected Result</b>	Valid input accepted; invalid input rejected with error.
<b>Validation Criteria</b>	Error shown for invalid input; Valid input accepted
<b>Dependencies</b>	Field validation logic
<b>Notes</b>	Covers regulatory compliance.

## Test Case 5: Check mandatory field enforcement in entitlement forms

<b>Test ID</b>	TC_005
<b>Module</b>	UI > Form Validation
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Test that forms cannot be submitted if mandatory fields are left blank.
<b>Objective</b>	Ensure mandatory fields are enforced at UI and backend.
<b>Preconditions</b>	Entitlement form available
<b>Test Steps</b>	Open entitlement form; Leave mandatory fields blank; Attempt to submit
<b>Expected Result</b>	Form submission blocked with appropriate error message.
<b>Validation Criteria</b>	Error message shown; No data saved
<b>Dependencies</b>	UI validation
<b>Notes</b>	Test for all mandatory fields.

## Test Case 6: Boundary test for daily single limit field (max 12 digits)

<b>Test ID</b>	TC_006
<b>Module</b>	Data Validation > Field Length
<b>Category</b>	Boundary
<b>Priority</b>	High

<b>Test Type</b>	Boundary/Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Test that the daily single limit field accepts up to 12 digits and rejects more.
<b>Objective</b>	Validate field length enforcement.
<b>Preconditions</b>	Daily single limit field present
<b>Test Steps</b>	Enter 12-digit value; Submit; Enter 13-digit value; Submit
<b>Expected Result</b>	12-digit value accepted; 13-digit value rejected.
<b>Validation Criteria</b>	Max length enforced; Error for overflow
<b>Dependencies</b>	Field validation
<b>Notes</b>	Covers numeric field boundaries.

## Test Case 7: Verify dropdown default values in entitlement parameters

<b>Test ID</b>	TC_007
<b>Module</b>	UI > Dropdowns
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Ensure all entitlement parameter dropdowns have correct default values.
<b>Objective</b>	Validate UI default value logic.
<b>Preconditions</b>	Dropdowns configured with defaults
<b>Test Steps</b>	Open entitlement form; Check all dropdown default values
<b>Expected Result</b>	All dropdowns show correct default values as per config.
<b>Validation Criteria</b>	Defaults match config
<b>Dependencies</b>	Dropdown configuration
<b>Notes</b>	Covers UI/UX consistency.

## Test Case 8: Test conditional display of fields based on Egypt GCIF entitlement

<b>Test ID</b>	TC_008
<b>Module</b>	UI > Conditional Logic
<b>Category</b>	Functional

Priority	High
Test Type	Positive/Negative
Risk Level	Medium
Estimated Time	6 minutes
Description	Verify that fields appear or hide based on GCIF level Egypt country entitlement.
Objective	Ensure conditional UI logic is enforced.
Preconditions	GCIF entitlement logic implemented
Test Steps	Login as Egypt GCIF user; Open entitlement screen; Observe field visibility; Login as non-Egypt user; Repeat
Expected Result	Fields show/hide as per regional logic.
Validation Criteria	Fields visible only for Egypt GCIF
Dependencies	Regional logic
Notes	Critical for regional compliance.

## Test Case 9: Validate parameterization of entitlement options (Adhoc Bill, Next Authorizer, Verifier/Releaser)

Test ID	TC_009
Module	Entitlements > Parameterization
Category	Functional
Priority	High
Test Type	Positive
Risk Level	Medium
Estimated Time	7 minutes
Description	Test that entitlement options are parameterized and can be enabled/disabled via checkboxes/dropdowns.
Objective	Ensure entitlement options are configurable.
Preconditions	Parameterization logic implemented
Test Steps	Open entitlement config; Toggle Adhoc Bill, Next Authorizer, Verifier/Releaser; Save; Verify changes applied
Expected Result	Options can be enabled/disabled and changes reflected in entitlements.
Validation Criteria	Changes saved; Reflected in user entitlements
Dependencies	Parameterization feature
Notes	Supports future extensibility.

## Test Case 10: Test beneficiary addition via customer portal

Test ID	TC_010
Module	Admin Portal > Beneficiary Management
Category	Functional

<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Verify that beneficiaries can be added via the customer portal as per Egypt-specific requirements.
<b>Objective</b>	Ensure customer portal supports beneficiary addition.
<b>Preconditions</b>	Customer portal access; Beneficiary management enabled
<b>Test Steps</b>	Login to customer portal; Navigate to beneficiary management; Add new beneficiary; Save
<b>Expected Result</b>	Beneficiary added and visible in list.
<b>Validation Criteria</b>	Beneficiary appears in list; DB record created
<b>Dependencies</b>	Customer portal
<b>Notes</b>	Egypt-specific requirement.

## Test Case 11: Test beneficiary addition via admin portal

<b>Test ID</b>	TC_011
<b>Module</b>	Admin Portal > Beneficiary Management
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Verify that beneficiaries can be added via the admin portal and are visible to customers.
<b>Objective</b>	Ensure admin portal supports beneficiary addition.
<b>Preconditions</b>	Admin portal access; Beneficiary management enabled
<b>Test Steps</b>	Login to admin portal; Navigate to beneficiary management; Add new beneficiary; Save
<b>Expected Result</b>	Beneficiary added and visible to customers.
<b>Validation Criteria</b>	Beneficiary visible to customers
<b>Dependencies</b>	Admin portal
<b>Notes</b>	Compare with customer portal.

## Test Case 12: Verify audit trail creation for entitlement changes

<b>Test ID</b>	TC_012
<b>Module</b>	Compliance > Audit Trail
<b>Category</b>	Functional
<b>Priority</b>	Critical

<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Test that all changes to entitlements are logged in the audit trail.
<b>Objective</b>	Ensure auditability for compliance.
<b>Preconditions</b>	Audit trail feature enabled
<b>Test Steps</b>	Modify entitlement; Check audit trail for entry
<b>Expected Result</b>	Audit trail entry created with change details.
<b>Validation Criteria</b>	Audit entry contains user, timestamp, change
<b>Dependencies</b>	Audit trail implementation
<b>Notes</b>	Regulatory requirement.

### Test Case 13: Test input validation for numeric-only fields

<b>Test ID</b>	TC_013
<b>Module</b>	Data Validation > Field Validation
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Ensure that numeric-only fields reject alphabetic or special character input.
<b>Objective</b>	Prevent invalid data entry.
<b>Preconditions</b>	Numeric field present
<b>Test Steps</b>	Enter alphabetic characters; Submit; Enter special characters; Submit
<b>Expected Result</b>	Input rejected with error message.
<b>Validation Criteria</b>	Error for non-numeric input
<b>Dependencies</b>	Field validation
<b>Notes</b>	Covers NUM/N(XX) fields.

### Test Case 14: Check system behavior when adding new payment types post go-live

<b>Test ID</b>	TC_014
<b>Module</b>	Configuration Management > Payment Type Management
<b>Category</b>	Functional
<b>Priority</b>	Critical

Test Type	Positive
Risk Level	High
Estimated Time	8 minutes
Description	Test system's ability to add new governmental payment types via configuration after go-live.
Objective	Validate system extensibility.
Preconditions	Admin access; Payment type config enabled
Test Steps	Login as admin; Navigate to payment type config; Add new payment type; Save; Verify availability in entitlement a
Expected Result	New payment type available for entitlement assignment.
Validation Criteria	New type appears in UI; Assignable to users
Dependencies	Config management
Notes	Supports future extensibility.

## Test Case 15: Verify error handling for duplicate beneficiary addition

Test ID	TC_015
Module	Admin Portal > Beneficiary Management
Category	Error
Priority	High
Test Type	Negative
Risk Level	Medium
Estimated Time	4 minutes
Description	Ensure system prevents adding duplicate beneficiaries and shows appropriate error.
Objective	Prevent duplicate data entry.
Preconditions	Beneficiary already exists
Test Steps	Attempt to add beneficiary with existing details; Submit
Expected Result	Error message shown; duplicate not added.
Validation Criteria	No duplicate in DB; Error displayed
Dependencies	Beneficiary uniqueness logic
Notes	Data integrity check.

## Test Case 16: Test API endpoint for entitlement retrieval

Test ID	TC_016
Module	API > Entitlement Retrieval
Category	Integration
Priority	High



Test Type	Positive
Risk Level	Medium
Estimated Time	5 minutes
Description	Verify API returns correct entitlement data for a given user.
Objective	Ensure API correctness and data integrity.
Preconditions	API endpoint available; User with entitlements exists
Test Steps	Call API with user ID; Check response data
Expected Result	API returns correct entitlement data.
Validation Criteria	Data matches DB; No missing fields
Dependencies	API implementation
Notes	API contract validation.

## Test Case 17: Test API endpoint for entitlement update with invalid payload

Test ID	TC_017
Module	API > Entitlement Update
Category	Integration
Priority	High
Test Type	Negative/Error
Risk Level	Medium
Estimated Time	5 minutes
Description	Verify API rejects invalid payloads with proper error messages.
Objective	Ensure API input validation and error handling.
Preconditions	API endpoint available
Test Steps	Call API with invalid payload; Check error response
Expected Result	API returns error with details of invalid input.
Validation Criteria	Error response structure; No update performed
Dependencies	API input validation
Notes	Covers API robustness.

## Test Case 18: Test login with invalid credentials

Test ID	TC_018
Module	Security > Authentication
Category	Security
Priority	Critical

<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Ensure system rejects login attempts with invalid credentials.
<b>Objective</b>	Prevent unauthorized access.
<b>Preconditions</b>	Login page available
<b>Test Steps</b>	Enter invalid username/password; Submit
<b>Expected Result</b>	Login denied with error message.
<b>Validation Criteria</b>	No access granted; Error shown
<b>Dependencies</b>	Authentication system
<b>Notes</b>	Basic security test.

## Test Case 19: Test session timeout for inactive users

<b>Test ID</b>	TC_019
<b>Module</b>	Security > Session Management
<b>Category</b>	Security
<b>Priority</b>	High
<b>Test Type</b>	Positive/Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	7 minutes
<b>Description</b>	Verify user is logged out after a period of inactivity.
<b>Objective</b>	Prevent unauthorized access via stale sessions.
<b>Preconditions</b>	User logged in
<b>Test Steps</b>	Login; Remain inactive for timeout period; Attempt to perform action
<b>Expected Result</b>	User session expired; must login again.
<b>Validation Criteria</b>	Session invalidated; Action blocked
<b>Dependencies</b>	Session management config
<b>Notes</b>	Covers session security.

## Test Case 20: Test file upload for entitlement bulk assignment

<b>Test ID</b>	TC_020
<b>Module</b>	Entitlements > Bulk Assignment
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Positive

<b>Risk Level</b>	Medium
<b>Estimated Time</b>	10 minutes
<b>Description</b>	Verify system processes entitlement bulk assignment files correctly.
<b>Objective</b>	Ensure file upload and processing workflow is functional.
<b>Preconditions</b>	Bulk assignment feature enabled
<b>Test Steps</b>	Prepare valid entitlement file; Upload via UI; Check processing status
<b>Expected Result</b>	File processed successfully; entitlements assigned.
<b>Validation Criteria</b>	Entitlements assigned; No errors in log
<b>Dependencies</b>	Bulk assignment module
<b>Notes</b>	Covers file upload workflow.

## Test Case 21: Test file upload with invalid format for entitlement bulk assignment

<b>Test ID</b>	TC_021
<b>Module</b>	Entitlements > Bulk Assignment
<b>Category</b>	Error
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Verify system rejects entitlement files with invalid format.
<b>Objective</b>	Prevent processing of malformed files.
<b>Preconditions</b>	Bulk assignment feature enabled
<b>Test Steps</b>	Prepare invalid entitlement file; Upload via UI
<b>Expected Result</b>	File rejected with error message.
<b>Validation Criteria</b>	File not processed; Error shown
<b>Dependencies</b>	File validation logic
<b>Notes</b>	Covers negative file upload scenario.

## Test Case 22: Test entitlement assignment for Governmental Payments (Tax, Custom, Universal)

<b>Test ID</b>	TC_022
<b>Module</b>	Entitlements > Payment Type Assignment
<b>Category</b>	Functional
<b>Priority</b>	High

Test Type	Positive
Risk Level	Medium
Estimated Time	7 minutes
Description	Verify that entitlements for Tax, Custom, and Universal Collection are mapped under Governmental Payments.
Objective	Ensure correct mapping of sub-products.
Preconditions	Governmental payment types configured
Test Steps	Assign entitlement for Tax; Check mapping; Repeat for Custom and Universal
Expected Result	All sub-products mapped under Governmental Payments.
Validation Criteria	Correct mapping in DB/UI
Dependencies	Payment type config
Notes	Covers business rule mapping.

## Test Case 23: Test input validation for date fields (DD-MM-YYYY)

Test ID	TC_023
Module	Data Validation > Field Validation
Category	Functional
Priority	Medium
Test Type	Negative
Risk Level	Low
Estimated Time	3 minutes
Description	Ensure system rejects invalid date formats in date fields.
Objective	Prevent invalid date entry.
Preconditions	Date field present
Test Steps	Enter date as MM-DD-YYYY; Submit
Expected Result	Input rejected with format error.
Validation Criteria	Error for invalid format
Dependencies	Date field validation
Notes	Covers date format enforcement.

## Test Case 24: Test input validation for time fields (HH:MM:SS)

Test ID	TC_024
Module	Data Validation > Field Validation
Category	Functional
Priority	Medium
Test Type	Negative

<b>Risk Level</b>	Low
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Ensure system rejects invalid time formats in time fields.
<b>Objective</b>	Prevent invalid time entry.
<b>Preconditions</b>	Time field present
<b>Test Steps</b>	Enter time as 25:61:61; Submit
<b>Expected Result</b>	Input rejected with format error.
<b>Validation Criteria</b>	Error for invalid time
<b>Dependencies</b>	Time field validation
<b>Notes</b>	Edge case for time validation.

## Test Case 25: Test system response to entitlement assignment with missing dependencies

<b>Test ID</b>	TC_025
<b>Module</b>	Entitlements > Assignment Logic
<b>Category</b>	Error
<b>Priority</b>	High
<b>Test Type</b>	Negative/Error
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Verify system handles missing payment type dependency gracefully.
<b>Objective</b>	Ensure error handling for missing dependencies.
<b>Preconditions</b>	Payment type dependency not configured
<b>Test Steps</b>	Attempt to assign entitlement for missing payment type; Submit
<b>Expected Result</b>	Error message shown; assignment blocked.
<b>Validation Criteria</b>	Error for missing dependency
<b>Dependencies</b>	Dependency validation logic
<b>Notes</b>	Covers dependency checks.

## Test Case 26: Test performance of entitlement retrieval under load

<b>Test ID</b>	TC_026
<b>Module</b>	Performance > API
<b>Category</b>	Performance
<b>Priority</b>	Critical

Test Type	Positive
Risk Level	High
Estimated Time	15 minutes
Description	Simulate multiple concurrent entitlement retrievals and measure response time.
Objective	Ensure system performs under high load.
Preconditions	Load testing tools available
Test Steps	Simulate 1000 concurrent API calls; Measure response time
Expected Result	All requests served within acceptable response time.
Validation Criteria	Response time SLA met
Dependencies	API; Load test setup
Notes	Performance benchmark.

## Test Case 27: Test UI usability for entitlement assignment screen

Test ID	TC_027
Module	UI > Usability
Category	Usability
Priority	Medium
Test Type	Positive
Risk Level	Low
Estimated Time	8 minutes
Description	Assess ease of use and intuitiveness of entitlement assignment UI.
Objective	Ensure good user experience.
Preconditions	UI deployed
Test Steps	Navigate to entitlement assignment; Assign entitlements; Observe workflow
Expected Result	User can complete assignment without confusion.
Validation Criteria	No usability blockers
Dependencies	UI design
Notes	Subjective, may require feedback.

## Test Case 28: Test SQL injection vulnerability on entitlement forms

Test ID	TC_028
Module	Security > Input Validation
Category	Security
Priority	Critical

<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Verify system is protected against SQL injection in entitlement forms.
<b>Objective</b>	Ensure no security vulnerabilities.
<b>Preconditions</b>	Entitlement form available
<b>Test Steps</b>	Enter SQL injection payload; Submit
<b>Expected Result</b>	Input rejected; no DB compromise.
<b>Validation Criteria</b>	No DB changes; Error for malicious input
<b>Dependencies</b>	Input sanitization
<b>Notes</b>	Covers OWASP Top 10.

## Test Case 29: Test XSS vulnerability on free-format fields

<b>Test ID</b>	TC_029
<b>Module</b>	Security > Input Validation
<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 system is protected against XSS in free-format fields.
<b>Objective</b>	Prevent cross-site scripting attacks.
<b>Preconditions</b>	Free-format field in UI
<b>Test Steps</b>	Enter <script>alert('XSS')</script>; Submit; Check page rendering
<b>Expected Result</b>	Script not executed; input sanitized.
<b>Validation Criteria</b>	No script execution
<b>Dependencies</b>	XSS protection
<b>Notes</b>	Security compliance.

## Test Case 30: Test cross-browser compatibility for entitlement screens

<b>Test ID</b>	TC_030
<b>Module</b>	UI > Cross-Browser
<b>Category</b>	Usability
<b>Priority</b>	Medium

<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	10 minutes
<b>Description</b>	Verify entitlement screens render and function correctly in Chrome, Firefox, Edge, and Safari.
<b>Objective</b>	Ensure consistent user experience across browsers.
<b>Preconditions</b>	Browsers installed
<b>Test Steps</b>	Open entitlement screen in each browser; Check rendering and functionality
<b>Expected Result</b>	Screens display and function identically in all browsers.
<b>Validation Criteria</b>	No rendering issues; All features work
<b>Dependencies</b>	Browser compatibility
<b>Notes</b>	Cross-browser check.

## Test Case 31: Test backup and recovery of entitlement configuration

<b>Test ID</b>	TC_031
<b>Module</b>	Configuration Management > Backup/Recovery
<b>Category</b>	Functional
<b>Priority</b>	Critical
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	12 minutes
<b>Description</b>	Verify entitlement configuration can be backed up and restored without data loss.
<b>Objective</b>	Ensure disaster recovery readiness.
<b>Preconditions</b>	Backup/restore tools available
<b>Test Steps</b>	Backup current config; Make changes; Restore backup; Verify config restored
<b>Expected Result</b>	Config restored to pre-change state.
<b>Validation Criteria</b>	No data loss; Config matches backup
<b>Dependencies</b>	Backup/restore system
<b>Notes</b>	Disaster recovery test.

## Test Case 32: Test entitlement deletion and audit trail update

<b>Test ID</b>	TC_032
<b>Module</b>	Entitlements > Deletion
<b>Category</b>	Functional
<b>Priority</b>	High



<b>Test Type</b>	Positive
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Verify entitlement deletion is logged in the audit trail.
<b>Objective</b>	Ensure traceability of deletions.
<b>Preconditions</b>	Entitlement exists
<b>Test Steps</b>	Delete entitlement; Check audit trail
<b>Expected Result</b>	Audit log entry created for deletion.
<b>Validation Criteria</b>	Deletion logged
<b>Dependencies</b>	Audit trail
<b>Notes</b>	Compliance requirement.

### Test Case 33: Test unauthorized API access for entitlement update

<b>Test ID</b>	TC_033
<b>Module</b>	API > Security
<b>Category</b>	Security
<b>Priority</b>	Critical
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Verify API rejects unauthorized requests.
<b>Objective</b>	Prevent unauthorized API access.
<b>Preconditions</b>	API endpoint available
<b>Test Steps</b>	Call API without authentication token; Check response
<b>Expected Result</b>	API returns 401 Unauthorized.
<b>Validation Criteria</b>	401 error returned
<b>Dependencies</b>	API security
<b>Notes</b>	API security check.

### Test Case 34: Test entitlement assignment for users with multiple roles

<b>Test ID</b>	TC_034
<b>Module</b>	Entitlements > Role Management
<b>Category</b>	Functional

Priority	High
Test Type	Positive
Risk Level	Medium
Estimated Time	7 minutes
Description	Verify entitlement assignment logic for users with multiple roles.
Objective	Ensure correct entitlement aggregation.
Preconditions	User with multiple roles exists
Test Steps	Assign entitlements for each role; Check user's cumulative entitlements
Expected Result	User receives all entitlements from assigned roles.
Validation Criteria	All entitlements present
Dependencies	Role management
Notes	Role aggregation logic.

## Test Case 35: Test system behavior when entitlement assignment exceeds daily single limit

Test ID	TC_035
Module	Entitlements > Limit Enforcement
Category	Boundary
Priority	High
Test Type	Negative/Boundary
Risk Level	Medium
Estimated Time	4 minutes
Description	Verify system enforces daily single limit during entitlement assignment.
Objective	Prevent limit breaches.
Preconditions	Daily single limit set
Test Steps	Attempt to assign entitlement above limit; Submit
Expected Result	Assignment blocked; error message shown.
Validation Criteria	Assignment blocked; Error shown
Dependencies	Limit enforcement logic
Notes	Boundary test.

## Test Case 36: Test UI accessibility for entitlement screens (WCAG compliance)

Test ID	TC_036
Module	UI > Accessibility

Category	Usability
Priority	Medium
Test Type	Positive
Risk Level	Low
Estimated Time	10 minutes
Description	Verify entitlement screens meet accessibility standards (e.g., keyboard navigation, screen reader support).
Objective	Ensure accessibility compliance.
Preconditions	Accessibility tools available
Test Steps	Navigate screens with keyboard; Test with screen reader
Expected Result	All features accessible; no accessibility blockers.
Validation Criteria	All controls accessible
Dependencies	UI accessibility
Notes	Covers WCAG compliance.

## Test Case 37: Test database rollback on entitlement assignment failure

Test ID	TC_037
Module	Database > Transaction Management
Category	Error
Priority	Critical
Test Type	Negative
Risk Level	High
Estimated Time	8 minutes
Description	Verify database rolls back changes if entitlement assignment fails.
Objective	Ensure data integrity.
Preconditions	DB transaction management enabled
Test Steps	Start entitlement assignment; Force failure; Check DB state
Expected Result	No partial data saved; DB rolled back.
Validation Criteria	DB state unchanged
Dependencies	DB transaction logic
Notes	Data integrity check.

## Test Case 38: Test system notification for entitlement changes

Test ID	TC_038
Module	Entitlements > Notifications

<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Verify users receive notifications when their entitlements are changed.
<b>Objective</b>	Ensure user awareness of changes.
<b>Preconditions</b>	Notification system enabled
<b>Test Steps</b>	Change user entitlement; Check user notifications
<b>Expected Result</b>	User receives notification of change.
<b>Validation Criteria</b>	Notification sent
<b>Dependencies</b>	Notification system
<b>Notes</b>	User communication check.

### Test Case 39: Test entitlement assignment for non-Egypt region

<b>Test ID</b>	TC_039
<b>Module</b>	Entitlements > Regional Logic
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive/Negative
<b>Risk Level</b>	Low
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Verify entitlement assignment logic for non-Egypt regions.
<b>Objective</b>	Ensure regional logic is not incorrectly applied.
<b>Preconditions</b>	Non-Egypt region user
<b>Test Steps</b>	Assign entitlement to non-Egypt user; Check field visibility and assignment
<b>Expected Result</b>	No Egypt-specific fields or logic applied.
<b>Validation Criteria</b>	Only standard fields visible
<b>Dependencies</b>	Regional logic
<b>Notes</b>	Regional logic test.

### Test Case 40: Test system behavior on entitlement assignment with expired session

<b>Test ID</b>	TC_040
<b>Module</b>	Security > Session Management

<b>Category</b>	Security
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	7 minutes
<b>Description</b>	Verify system blocks entitlement assignment if user session has expired.
<b>Objective</b>	Prevent unauthorized actions.
<b>Preconditions</b>	Session timeout configured
<b>Test Steps</b>	Login; Wait for session to expire; Attempt to assign entitlement
<b>Expected Result</b>	Action blocked; user prompted to login.
<b>Validation Criteria</b>	Action blocked after expiry
<b>Dependencies</b>	Session management
<b>Notes</b>	Session security.

## Test Case 41: Test entitlement assignment with special characters in free format field

<b>Test ID</b>	TC_041
<b>Module</b>	Data Validation > Input Validation
<b>Category</b>	Functional
<b>Priority</b>	High
<b>Test Type</b>	Positive/Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Verify system only accepts allowed special characters in free-format fields per SWIFT.
<b>Objective</b>	Ensure SWIFT compliance.
<b>Preconditions</b>	Free format field in UI
<b>Test Steps</b>	Enter allowed special characters; Submit; Enter disallowed special characters; Submit
<b>Expected Result</b>	Allowed characters accepted; others rejected.
<b>Validation Criteria</b>	Only allowed chars accepted
<b>Dependencies</b>	SWIFT validation logic
<b>Notes</b>	Covers special char validation.

## Test Case 42: Test multi-dropdown selection for entitlement parameters

<b>Test ID</b>	TC_042
----------------	--------

<b>Module</b>	Entitlements > Parameterization
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Verify multi-dropdown fields allow multiple selections and save correctly.
<b>Objective</b>	Ensure multi-selection works.
<b>Preconditions</b>	Multi-dropdown field present
<b>Test Steps</b>	Select multiple options; Save; Verify selections saved
<b>Expected Result</b>	All selected options saved and displayed.
<b>Validation Criteria</b>	All selections saved
<b>Dependencies</b>	Multi-dropdown implementation
<b>Notes</b>	Multi-select test.

### Test Case 43: Test system behavior for entitlement assignment with invalid dropdown value

<b>Test ID</b>	TC_043
<b>Module</b>	Entitlements > Parameterization
<b>Category</b>	Error
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	5 minutes
<b>Description</b>	Verify system rejects entitlement assignment with invalid dropdown value (e.g., tampered payload).
<b>Objective</b>	Prevent invalid parameter values.
<b>Preconditions</b>	Dropdown field present
<b>Test Steps</b>	Tamper payload with invalid value; Submit; Check response
<b>Expected Result</b>	Assignment rejected; error shown.
<b>Validation Criteria</b>	Invalid value rejected
<b>Dependencies</b>	Dropdown validation
<b>Notes</b>	Covers input tampering.

### Test Case 44: Test system logging for failed entitlement assignments

Test ID	TC_044
Module	Compliance > Logging
Category	Error
Priority	Medium
Test Type	Negative
Risk Level	Low
Estimated Time	4 minutes
Description	Verify system logs failed entitlement assignments with error details.
Objective	Ensure traceability of failures.
Preconditions	Logging enabled
Test Steps	Attempt invalid assignment; Check logs
Expected Result	Error logged with details.
Validation Criteria	Error details in log
Dependencies	Logging system
Notes	Failure traceability.

## Test Case 45: Test entitlement assignment with fixed length field (exact length)

Test ID	TC_045
Module	Data Validation > Field Validation
Category	Boundary
Priority	Medium
Test Type	Positive/Boundary
Risk Level	Low
Estimated Time	3 minutes
Description	Verify system accepts values of exact required length for fixed length fields.
Objective	Enforce fixed length requirement.
Preconditions	Fixed length field present
Test Steps	Enter value of required length; Submit
Expected Result	Value accepted.
Validation Criteria	Accepted if length matches
Dependencies	Field validation
Notes	Fixed length test.

## Test Case 46: Test entitlement assignment with fixed length field (too short/long)

<b>Test ID</b>	TC_046
<b>Module</b>	Data Validation > Field Validation
<b>Category</b>	Boundary
<b>Priority</b>	Medium
<b>Test Type</b>	Negative/Boundary
<b>Risk Level</b>	Low
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Verify system rejects values not matching required length for fixed length fields.
<b>Objective</b>	Prevent incorrect field length.
<b>Preconditions</b>	Fixed length field present
<b>Test Steps</b>	Enter value too short; Submit; Enter value too long; Submit
<b>Expected Result</b>	Values rejected with error.
<b>Validation Criteria</b>	Error for wrong length
<b>Dependencies</b>	Field validation
<b>Notes</b>	Fixed length enforcement.

## Test Case 47: Test entitlement assignment with auto-fill field

<b>Test ID</b>	TC_047
<b>Module</b>	Data Validation > Field Validation
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Verify auto-fill fields populate correctly based on related data.
<b>Objective</b>	Ensure auto-fill logic works.
<b>Preconditions</b>	Auto-fill field present
<b>Test Steps</b>	Trigger auto-fill (e.g., select related field); Check auto-filled value
<b>Expected Result</b>	Field auto-filled with correct value.
<b>Validation Criteria</b>	Auto-fill correct
<b>Dependencies</b>	Auto-fill logic
<b>Notes</b>	Auto-fill feature.

## Test Case 48: Test user logout functionality

<b>Test ID</b>	TC_048
----------------	--------



<b>Module</b>	Security > Authentication
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Verify user can log out and session is terminated.
<b>Objective</b>	Ensure session termination.
<b>Preconditions</b>	User logged in
<b>Test Steps</b>	Click logout; Attempt to access protected page
<b>Expected Result</b>	Session terminated; access denied.
<b>Validation Criteria</b>	Session invalidated
<b>Dependencies</b>	Session management
<b>Notes</b>	Basic logout test.

## Test Case 49: Test entitlement assignment with optional fields left blank

<b>Test ID</b>	TC_049
<b>Module</b>	Entitlements > Assignment
<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>	Verify system accepts form submission with optional fields left blank.
<b>Objective</b>	Ensure optional fields are not enforced.
<b>Preconditions</b>	Optional fields present
<b>Test Steps</b>	Leave optional fields blank; Submit
<b>Expected Result</b>	Form submitted successfully.
<b>Validation Criteria</b>	Submission succeeds
<b>Dependencies</b>	Field configuration
<b>Notes</b>	Optional field logic.

## Test Case 50: Test entitlement assignment for user with no entitlements

Test ID	TC_050
Module	Entitlements > Assignment
Category	Functional
Priority	Low
Test Type	Positive/Negative
Risk Level	Low
Estimated Time	3 minutes
Description	Verify system handles users with no entitlements gracefully.
Objective	Ensure no errors for zero-entitlement users.
Preconditions	User with no entitlements exists
Test Steps	Login as user; Check entitlement screen
Expected Result	Screen displays 'No entitlements assigned' message.
Validation Criteria	No errors; Message displayed
Dependencies	User setup
Notes	Zero state handling.

## Test Case 51: Test concurrent entitlement assignment by multiple admins

Test ID	TC_051
Module	Entitlements > Assignment
Category	Performance
Priority	High
Test Type	Positive/Negative
Risk Level	Medium
Estimated Time	10 minutes
Description	Simulate multiple admins assigning entitlements at the same time to check for race conditions.
Objective	Ensure data consistency under concurrency.
Preconditions	Multiple admin accounts
Test Steps	Admin A and B assign entitlements to same user simultaneously; Check final entitlements
Expected Result	No data corruption; latest assignment reflected.
Validation Criteria	No data loss/corruption
Dependencies	Concurrency handling
Notes	Race condition test.

## Test Case 52: Test entitlement assignment with invalid amount format (non-numeric)

Test ID	TC_052
Module	Data Validation > Field Validation
Category	Functional
Priority	Medium
Test Type	Negative
Risk Level	Low
Estimated Time	3 minutes
Description	Verify system rejects non-numeric input in amount fields.
Objective	Prevent invalid amount entry.
Preconditions	Amount field present
Test Steps	Enter 'abc' in amount field; Submit
Expected Result	Input rejected with error.
Validation Criteria	Error for non-numeric input
Dependencies	Amount field validation
Notes	Amount validation.

### Test Case 53: Test entitlement assignment with amount in excess of two decimal places

Test ID	TC_053
Module	Data Validation > Field Validation
Category	Boundary
Priority	Medium
Test Type	Negative/Boundary
Risk Level	Low
Estimated Time	3 minutes
Description	Verify system rejects amounts with more than two decimal places.
Objective	Enforce amount format.
Preconditions	Amount field present
Test Steps	Enter 123.456; Submit
Expected Result	Input rejected with error.
Validation Criteria	Error for >2 decimals
Dependencies	Amount field validation
Notes	Decimal place enforcement.

### Test Case 54: Test entitlement assignment with leading/trailing spaces in alphanumeric fields

<b>Test ID</b>	TC_054
<b>Module</b>	Data Validation > Field Validation
<b>Category</b>	Functional
<b>Priority</b>	Low
<b>Test Type</b>	Negative
<b>Risk Level</b>	Low
<b>Estimated Time</b>	2 minutes
<b>Description</b>	Verify system trims or rejects leading/trailing spaces in alphanumeric fields.
<b>Objective</b>	Prevent invalid formatting.
<b>Preconditions</b>	Alphanumeric field present
<b>Test Steps</b>	Enter value with leading/trailing spaces; Submit
<b>Expected Result</b>	Spaces trimmed or error shown.
<b>Validation Criteria</b>	No spaces in stored value
<b>Dependencies</b>	Field validation
<b>Notes</b>	Alphanumeric field check.

## Test Case 55: Test entitlement assignment with alphabetic-only field containing numbers

<b>Test ID</b>	TC_055
<b>Module</b>	Data Validation > Field Validation
<b>Category</b>	Functional
<b>Priority</b>	Low
<b>Test Type</b>	Negative
<b>Risk Level</b>	Low
<b>Estimated Time</b>	2 minutes
<b>Description</b>	Verify system rejects numbers in alphabetic-only fields.
<b>Objective</b>	Prevent invalid data entry.
<b>Preconditions</b>	Alphabetic field present
<b>Test Steps</b>	Enter 'ABC123' in alphabetic field; Submit
<b>Expected Result</b>	Input rejected with error.
<b>Validation Criteria</b>	Error for numeric input
<b>Dependencies</b>	Field validation
<b>Notes</b>	Alphabetic field enforcement.

## Test Case 56: Test entitlement assignment with dropdown default override

<b>Test ID</b>	TC_056
<b>Module</b>	Entitlements > Parameterization
<b>Category</b>	Functional
<b>Priority</b>	Low
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	3 minutes
<b>Description</b>	Verify user can override default dropdown values and changes are saved.
<b>Objective</b>	Ensure user control over defaults.
<b>Preconditions</b>	Dropdowns with defaults present
<b>Test Steps</b>	Change default value; Save; Check saved value
<b>Expected Result</b>	Overridden value saved and displayed.
<b>Validation Criteria</b>	Override saved
<b>Dependencies</b>	Dropdown logic
<b>Notes</b>	Default override.

## Test Case 57: Test entitlement assignment with checkbox toggle

<b>Test ID</b>	TC_057
<b>Module</b>	Entitlements > Parameterization
<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>	Verify checkbox parameters can be toggled and changes are saved.
<b>Objective</b>	Ensure checkbox logic works.
<b>Preconditions</b>	Checkbox parameter present
<b>Test Steps</b>	Toggle checkbox; Save; Check saved state
<b>Expected Result</b>	Checkbox state saved and reflected in UI.
<b>Validation Criteria</b>	State updated
<b>Dependencies</b>	Checkbox logic
<b>Notes</b>	Checkbox parameter test.

## Test Case 58: Test entitlement assignment with future-dated effective date

<b>Test ID</b>	TC_058
<b>Module</b>	Entitlements > Assignment
<b>Category</b>	Functional
<b>Priority</b>	Medium
<b>Test Type</b>	Positive
<b>Risk Level</b>	Low
<b>Estimated Time</b>	4 minutes
<b>Description</b>	Verify system allows entitlement assignment with a future effective date.
<b>Objective</b>	Support future-dated assignments.
<b>Preconditions</b>	Effective date field present
<b>Test Steps</b>	Set future date; Assign entitlement; Check assignment
<b>Expected Result</b>	Entitlement scheduled for future activation.
<b>Validation Criteria</b>	Scheduled correctly
<b>Dependencies</b>	Effective date logic
<b>Notes</b>	Future-dated assignment.

## Test Case 59: Test entitlement assignment with past effective date

<b>Test ID</b>	TC_059
<b>Module</b>	Entitlements > Assignment
<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>	Verify system rejects entitlement assignment with a past effective date.
<b>Objective</b>	Prevent backdated assignments.
<b>Preconditions</b>	Effective date field present
<b>Test Steps</b>	Set past date; Assign entitlement; Check error
<b>Expected Result</b>	Assignment rejected with error.
<b>Validation Criteria</b>	Error for past date
<b>Dependencies</b>	Effective date validation
<b>Notes</b>	Backdating prevention.

## Test Case 60: Test entitlement assignment with maximum number of entitlements per user

<b>Test ID</b>	TC_060
<b>Module</b>	Entitlements > Assignment
<b>Category</b>	Boundary
<b>Priority</b>	Medium
<b>Test Type</b>	Negative/Boundary
<b>Risk Level</b>	Low
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Verify system enforces maximum number of entitlements per user.
<b>Objective</b>	Prevent exceeding entitlement cap.
<b>Preconditions</b>	Max entitlement cap configured
<b>Test Steps</b>	Assign entitlements up to cap; Attempt to assign one more; Check error
<b>Expected Result</b>	Cap enforced; additional assignment blocked.
<b>Validation Criteria</b>	Cap enforced
<b>Dependencies</b>	Cap configuration
<b>Notes</b>	Entitlement cap.

## Test Case 61: Test entitlement assignment with simultaneous API and UI updates

<b>Test ID</b>	TC_061
<b>Module</b>	Integration > Concurrency
<b>Category</b>	Integration
<b>Priority</b>	High
<b>Test Type</b>	Positive/Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	10 minutes
<b>Description</b>	Simulate updating entitlements via API and UI at the same time to check for data consistency.
<b>Objective</b>	Ensure no conflicts between API and UI updates.
<b>Preconditions</b>	API and UI access
<b>Test Steps</b>	Update entitlement via API; Simultaneously update via UI; Check final state
<b>Expected Result</b>	No data corruption; latest update reflected.
<b>Validation Criteria</b>	No data loss/corruption
<b>Dependencies</b>	API and UI
<b>Notes</b>	Concurrency integration.

## Test Case 62: Test system response to entitlement assignment when database is unavailable

<b>Test ID</b>	TC_062
<b>Module</b>	Database > Error Handling
<b>Category</b>	Error
<b>Priority</b>	Critical
<b>Test Type</b>	Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	6 minutes
<b>Description</b>	Verify system handles DB outages gracefully during entitlement assignment.
<b>Objective</b>	Prevent crashes and provide user feedback.
<b>Preconditions</b>	Ability to simulate DB outage
<b>Test Steps</b>	Simulate DB outage; Attempt entitlement assignment; Check error handling
<b>Expected Result</b>	User receives error; system does not crash.
<b>Validation Criteria</b>	Graceful error; No crash
<b>Dependencies</b>	Error handling logic
<b>Notes</b>	Resilience test.

## Test Case 63: Test compliance with regulatory reporting for entitlement changes

<b>Test ID</b>	TC_063
<b>Module</b>	Compliance > Regulatory Reporting
<b>Category</b>	Functional
<b>Priority</b>	Critical
<b>Test Type</b>	Positive
<b>Risk Level</b>	High
<b>Estimated Time</b>	7 minutes
<b>Description</b>	Verify all entitlement changes are reflected in regulatory reports.
<b>Objective</b>	Ensure regulatory compliance.
<b>Preconditions</b>	Regulatory reporting enabled
<b>Test Steps</b>	Change entitlement; Generate report; Verify change in report
<b>Expected Result</b>	Change appears in regulatory report.
<b>Validation Criteria</b>	Change in report
<b>Dependencies</b>	Reporting system
<b>Notes</b>	Regulatory requirement.

## Test Case 64: Test system behavior on entitlement assignment with simultaneous user deletion



<b>Test ID</b>	TC_064
<b>Module</b>	Entitlements > Assignment
<b>Category</b>	Error
<b>Priority</b>	High
<b>Test Type</b>	Negative
<b>Risk Level</b>	Medium
<b>Estimated Time</b>	8 minutes
<b>Description</b>	Simulate assigning entitlement while user is being deleted to check for error handling.
<b>Objective</b>	Ensure system handles race conditions gracefully.
<b>Preconditions</b>	User deletion and assignment can be triggered simultaneously
<b>Test Steps</b>	Start entitlement assignment; Simultaneously delete user; Check system response
<b>Expected Result</b>	Assignment blocked; error message shown.
<b>Validation Criteria</b>	No assignment to deleted user
<b>Dependencies</b>	User management
<b>Notes</b>	Race condition check.

## Test Case 65: Test system recovery after entitlement assignment failure and rollback

<b>Test ID</b>	TC_065
<b>Module</b>	Database > Recovery
<b>Category</b>	Error
<b>Priority</b>	Critical
<b>Test Type</b>	Positive/Negative
<b>Risk Level</b>	High
<b>Estimated Time</b>	10 minutes
<b>Description</b>	Verify system can recover from entitlement assignment failure and resume normal operations.
<b>Objective</b>	Ensure system stability after errors.
<b>Preconditions</b>	Ability to simulate failure
<b>Test Steps</b>	Simulate assignment failure; Rollback DB; Attempt new assignment
<b>Expected Result</b>	System resumes normal operation; new assignment succeeds.
<b>Validation Criteria</b>	System resumes; No residual errors
<b>Dependencies</b>	Recovery procedures
<b>Notes</b>	Recovery scenario.