# Test Cases Report

Generated 30 test cases on 2025-08-03 14:04:03

## Test Case 1: Boundary Value Test for Daily Single Limit Field

**Test ID:** TC_003

**Module:** Corporate Module - Entitlement Management

**Priority:** High | **Category:** Functional

**Description:** Test the boundary values for the 'Daily Single Limit' field during entitlement setup.

**Objective:** Ensure the system enforces minimum and maximum allowed values for the limit field.

**Test Steps:**
1. Navigate to the 'Governmental Payments' entitlement setup.
Test Data: navigation_path: /admin/entitlements/governmental-payments, user_credentials: admin_user/SuperAdmin!2024
Expected: Entitlement setup page is displayed.

2. Enter minimum allowed value (0.01) in 'Daily Single Limit' field.
Test Data: input_field_1: 0.01, field_name: Daily Single Limit
Expected: System accepts the minimum value.

3. Enter maximum allowed value (99999999.99) in 'Daily Single Limit' field.
Test Data: input_field_1: 99999999.99, field_name: Daily Single Limit
Expected: System accepts the maximum value.

4. Enter value exceeding maximum (100000000.00) and attempt to save.
Test Data: input_field_1: 100000000.00, field_name: Daily Single Limit
Expected: System displays validation error and does not save the value.

## Test Case 2: SWIFT Compliance Validation for Free Format Field

**Test ID:** TC_004

**Module:** Corporate Module - Form Validation

**Priority:** High | **Category:** Functional

**Description:** Test that the free format field only accepts SWIFT-compliant characters.

**Objective:** Ensure only allowed characters are accepted in SWIFT-compliant fields.

**Test Steps:**
1. Navigate to 'Custom Collection' payment form.
Test Data: navigation_path: /corporate/governmental-payments/custom-collection, user_credentials: corp_user3/Passw0rd!@#
Expected: Custom Collection payment form is displayed.

2. Enter SWIFT-compliant value in free format field.
Test Data: input_field_1: Invoice#1234-ABCD/2024, field_name: Free Format, allowed_characters: A-Z, 0-9, /, -, ?

Expected: Field accepts the input.

3. Enter non-compliant value with disallowed characters (e.g., 'Invoice@2024!').
Test Data: input_field_1: Invoice@2024!, field_name: Free Format, disallowed_characters: @, !
Expected: System displays error and does not accept the input.

## Test Case 3: Auto-Rejection of Unapproved Transactions after 45 Days

**Test ID:** TC_005

**Module:** Corporate Module - Transaction Handling

**Priority:** Critical | **Category:** Integration

**Description:** Verify that transactions not approved within 45 days are automatically rejected by the system.

**Objective:** Ensure scheduled job or workflow auto-rejects stale transactions.

**Test Steps:**
1. Insert a pending transaction dated 45 days ago into the database.
Test Data: transaction_id: TXN123456, status: Pending Approval, created_date: 2024-05-20, user: corp_user4
Expected: Transaction exists in database with correct status and date.

2. Trigger the scheduled auto-rejection job.
Test Data: job_name: AutoRejectPendingTransactions, execution_time: 2024-07-04 02:00:00
Expected: Job runs and processes eligible transactions.

3. Query the transaction status after job execution.
Test Data: transaction_id: TXN123456, database_query: SELECT status FROM transactions WHERE transaction_id='TXN123456';
Expected: Transaction status is updated to 'Rejected'.

## Test Case 4: Dynamic Field Display Based on GCIF Country Entitlement

**Test ID:** TC_006

**Module:** Corporate Module - UI Rendering

**Priority:** Medium | **Category:** Functional

**Description:** Test that 'Tax Collection' fields are only displayed when the GCIF is entitled for Egypt.

**Objective:** Ensure dynamic UI rendering based on entitlement and country.

**Test Steps:**
1. Login as 'corp_user5' with Egypt entitlement.
Test Data: user_credentials: corp_user5/Egypt@2024, GCIF: 100200, country: Egypt
Expected: User is authenticated.

2. Navigate to 'Governmental Payments' module.
Test Data: navigation_path: /corporate/governmental-payments, user: corp_user5
Expected: 'Tax Collection' sub-module is visible.

3. Login as 'corp_user6' with GCIF not entitled for Egypt.
Test Data: user_credentials: corp_user6/India@2024, GCIF: 100201, country: India
Expected: 'Tax Collection' sub-module is not visible.

## Test Case 5: Add Beneficiary via Admin Portal and Validate Parity

**Test ID:** TC_007

**Module:** Corporate Module - Admin Portal

**Priority:** High | **Category:** Integration

**Description:** Test adding a new beneficiary via admin portal and verify parity with customer portal.

**Objective:** Ensure admin and customer portals have equivalent beneficiary management capabilities.

**Test Steps:**
1. Navigate to 'Beneficiary Management' in admin portal.
Test Data: navigation_path: /admin/beneficiaries, user_credentials: admin_user/SuperAdmin!2024
Expected: Beneficiary management page is displayed.

2. Add new beneficiary with valid details.
Test Data: beneficiary_name: ABC Corp, account_number: 123456789012, bank_code: EGYBIC22, payment_type: Governmental Payment
Expected: Beneficiary is added successfully.

3. Login to customer portal and verify beneficiary appears.
Test Data: user_credentials: corp_user7/Beneficiary@2024, navigation_path: /corporate/beneficiaries
Expected: ABC Corp is listed as a beneficiary.

## Test Case 6: Attempt to Add Beneficiary with Invalid Account Number

**Test ID:** TC_008

**Module:** Corporate Module - Admin Portal

**Priority:** Medium | **Category:** Functional

**Description:** Test adding a beneficiary with an invalid (non-numeric/incorrect length) account number.

**Objective:** Ensure system enforces account number format and length validation.

**Test Steps:**
1. Go to 'Beneficiary Management' in admin portal.
Test Data: navigation_path: /admin/beneficiaries, user_credentials: admin_user/SuperAdmin!2024
Expected: Beneficiary management page is displayed.

2. Attempt to add beneficiary with invalid account number '12ABCD'.
Test Data: beneficiary_name: XYZ Ltd, account_number: 12ABCD, bank_code: EGYBIC22, payment_type: Governmental Payment
Expected: System displays validation error for account number.

3. Attempt to add beneficiary with too short account number '12345'.
Test Data: beneficiary_name: XYZ Ltd, account_number: 12345, bank_code: EGYBIC22, payment_type: Governmental Payment
Expected: System displays validation error for account number length.

## Test Case 7: Error Handling: Database Connection Failure During Payment Submission

**Test ID:** TC_010

**Module:** Corporate Module - Payment Processing

**Priority:** High | **Category:** Error Handling

**Description:** Test system behavior when a database connection error occurs during payment submission.

**Objective:** Ensure user receives a clear error and no data is lost or corrupted.

**Test Steps:**
1. Login as 'corp_user9' and navigate to payment form.
Test Data: user_credentials: corp_user9/DBError@2024, navigation_path: /corporate/governmental-payments/new
Expected: Payment form is displayed.

2. Enter valid payment details.
Test Data: amount: 5000.00, beneficiary: DEF Corp, payment_type: Tax Collection, reference: TAX-2024-001
Expected: Form fields accept input.

3. Simulate database outage and submit payment.
Test Data: database_status: Disconnected, submission_action: Submit
Expected: System displays error message: 'Unable to process payment at this time. Please try again later.'

4. Verify that no payment record is created in the database.
Test Data: database_query: SELECT * FROM payments WHERE reference='TAX-2024-001';
Expected: No record is found for the attempted payment.

## Test Case 8: Boundary Test: Maximum Field Length for Beneficiary Name

**Test ID:** TC_012

**Module:** Corporate Module - Beneficiary Management

**Priority:** High | **Category:** Boundary

**Description:** Test adding a beneficiary with the beneficiary name field set to its maximum allowed length (50 characters).

**Objective:** Ensure the system accepts beneficiary names up to the maximum length and rejects longer values.

**Test Steps:**
1. Navigate to Add Beneficiary and enter maximum length name
Test Data: beneficiary_name: ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMN1234567890, account_number: 1234567890123456, module: Governmental Payment, environment: UAT1, user_credentials: corpuser01/CorpPass!2024
Expected: Beneficiary name is accepted and no validation error is shown

2. Attempt to add beneficiary with name exceeding maximum length
Test Data: beneficiary_name: ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMN12345678901, account_number: 1234567890123456, module: Governmental Payment

Expected: System displays validation error for beneficiary name length

## Test Case 9: Security Test: SWIFT Compliance for Free Format Fields

**Test ID:** TC_013

**Module:** Corporate Module - Payment Initiation

**Priority:** High | **Category:** Security

**Description:** Test entry of allowed and disallowed special characters in a SWIFT-compliant free format field during payment initiation.

**Objective:** Ensure only SWIFT-compliant characters are accepted and others are rejected.

**Test Steps:**
1. Initiate payment and enter allowed SWIFT characters
Test Data: free_format_field: Invoice/2024-05-01:Ref#12345, amount: 10000.00, beneficiary: GovTaxDept, environment: UAT1, user_credentials: corpuser02/CorpPass!2024
Expected: Payment is accepted and proceeds to next step

2. Attempt to enter disallowed characters
Test Data: free_format_field: Invoice<>2024|DROP TABLE, amount: 10000.00, beneficiary: GovTaxDept
Expected: System displays validation error for disallowed characters

## Test Case 10: Error Handling: Auto-Rejection of Unapproved Transactions after 45 Days

**Test ID:** TC_015

**Module:** Corporate Module - Transaction Management

**Priority:** High | **Category:** Error Handling

**Description:** Create a transaction and simulate passage of 45 days without approval to verify automatic rejection.

**Objective:** Ensure system automatically rejects transactions not approved within 45 days.

**Test Steps:**
1. Initiate a governmental payment transaction
Test Data: transaction_id: TXN10001, amount: 75000.00, beneficiary: CustomsDept, date_initiated: 2024-05-01, environment: UAT1, user_credentials: corpuser03/CorpPass!2024
Expected: Transaction is created and pending approval

2. Simulate system date advance by 46 days
Test Data: system_date: 2024-06-16, transaction_id: TXN10001
Expected: Scheduled job runs and transaction status is updated

3. Check transaction status after 46 days
Test Data: transaction_id: TXN10001, environment: UAT1
Expected: Transaction status is 'Auto-Rejected'

## Test Case 11: Usability: Conditional Display of Tax Collection Fields

**Test ID:** TC_016

**Module:** Corporate Module - Payment Initiation

**Priority:** Medium | **Category:** Usability

**Description:** Test that Tax Collection fields are displayed only when the customer has Egypt country entitlement at GCIF level.

**Objective:** Ensure dynamic UI behavior for conditional field display.

**Test Steps:**
1. Login as customer with Egypt entitlement and navigate to payment initiation
Test Data: GCIF: 90012345, country: Egypt, user_credentials: corpuser04/CorpPass!2024, navigation_path: /payments/initiate
Expected: Tax Collection fields are visible in the form

2. Login as customer without Egypt entitlement and navigate to payment initiation
Test Data: GCIF: 90054321, country: UAE, user_credentials: corpuser05/CorpPass!2024, navigation_path: /payments/initiate
Expected: Tax Collection fields are not displayed

## Test Case 12: File Upload: Upload Valid and Invalid Beneficiary Files

**Test ID:** TC_017

**Module:** Corporate Module - Beneficiary Management

**Priority:** High | **Category:** Functional

**Description:** Test uploading beneficiary files in supported (CSV, XLSX) and unsupported (EXE, PDF) formats.

**Objective:** Ensure system accepts only supported file types and rejects others.

**Test Steps:**
1. Upload a valid CSV file with beneficiary data
Test Data: file_name: beneficiaries_valid.csv, file_type: text/csv, file_size: 45KB, environment: UAT1, user_credentials: corpuser06/CorpPass!2024
Expected: File is accepted and beneficiaries are imported

2. Upload a valid XLSX file with beneficiary data
Test Data: file_name: beneficiaries_valid.xlsx, file_type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet, file_size: 60KB
Expected: File is accepted and beneficiaries are imported

3. Attempt to upload an unsupported EXE file
Test Data: file_name: malware.exe, file_type: application/x-msdownload, file_size: 120KB
Expected: System rejects the file and displays an error message

4. Attempt to upload an unsupported PDF file
Test Data: file_name: beneficiaries.pdf, file_type: application/pdf, file_size: 80KB
Expected: System rejects the file and displays an error message

## Test Case 13: Security: Attempt Unauthorized Entitlement Assignment

**Test ID:** TC_019

**Module:** Corporate Module - Admin Portal - Entitlement Management

**Priority:** Critical | **Category:** Security

**Description:** Test that a non-admin user cannot assign governmental payment entitlements to a customer.

**Objective:** Ensure role-based access control prevents unauthorized entitlement assignment.

**Test Steps:**
1. Attempt to assign 'Governmental Payment' entitlement as a regular user
Test Data: user_credentials: supportuser01/SupportPass!2024, target_GCIF: 90067890, entitlement: Governmental Payment, environment: UAT1
Expected: System denies access and displays insufficient privileges error

## Test Case 14: Auto-Rejection of Unapproved Transactions after 45 Days

**Test ID:** TC_023

**Module:** Corporate Module - Transaction Processing

**Priority:** Critical | **Category:** Functional

**Description:** Check that transactions pending approval for more than 45 days are automatically rejected by the system.

**Objective:** Ensure compliance with business rule for transaction expiry.

**Test Steps:**
1. Create a new governmental payment transaction and leave it pending approval.
Test Data: transaction_id: TXN100045, amount: 25000.00, beneficiary: Customs Authority, creation_date: 2024-05-01, user_credentials: corp_user2/Passw0rd!, environment: UAT
Expected: Transaction is created and status is 'Pending Approval'.

2. Advance system date to 2024-06-16 (46 days after creation).
Test Data: system_date: 2024-06-16, transaction_id: TXN100045, environment: UAT
Expected: System date is updated for test environment.

3. Run scheduled job for auto-rejection and check transaction status.
Test Data: scheduled_job: AutoRejectPendingTransactions, transaction_id: TXN100045, environment: UAT
Expected: Transaction status is updated to 'Rejected' with reason 'Approval expired'.

## Test Case 15: Boundary Test for Daily Single Limit Field

**Test ID:** TC_024

**Module:** Corporate Module - Entitlement Management

**Priority:** High | **Category:** Boundary

**Description:** Verify that the Daily Single Limit field enforces minimum and maximum boundaries as per system configuration.

**Objective:** Ensure boundary values are handled correctly.

**Test Steps:**
1. Enter minimum allowed value for Daily Single Limit.
Test Data: Daily Single Limit: 0.01, entitlement_type: Governmental Payment, environment: UAT, user_credentials: admin_user2/Adm1nPass!, additional_params: N/A
Expected: Field accepts minimum value and saves successfully.

2. Enter maximum allowed value for Daily Single Limit.
Test Data: Daily Single Limit: 1000000.00, entitlement_type: Governmental Payment, environment: UAT
Expected: Field accepts maximum value and saves successfully.

3. Enter value exceeding maximum limit.
Test Data: Daily Single Limit: 1000000.01, entitlement_type: Governmental Payment, environment: UAT
Expected: Validation error is displayed indicating limit exceeded.

## Test Case 16: Integration Test: Beneficiary Addition via Admin Portal

**Test ID:** TC_025

**Module:** Corporate Module - Beneficiary Management

**Priority:** Medium | **Category:** Integration

**Description:** Ensure that beneficiaries added via admin portal are immediately available in customer portal for entitled users.

**Objective:** Validate integration between admin and customer portals for beneficiary management.

**Test Steps:**
1. Admin adds a new beneficiary in the admin portal.
Test Data: beneficiary_name: National Tax Authority, account_number: EG1234567890123456789012, beneficiary_type: Tax Collection, environment: UAT, user_credentials: admin_user3/Adm1nPwd#, additional_params: N/A
Expected: Beneficiary is added and confirmation message displayed.

2. Customer portal user logs in and checks beneficiary list.
Test Data: user_credentials: corp_user3/CorpPwd!2024, navigation_path: /beneficiaries/list, environment: UAT
Expected: New beneficiary 'National Tax Authority' is visible in the list.

## Test Case 17: Performance Test: Bulk Beneficiary Addition

**Test ID:** TC_027

**Module:** Corporate Module - Beneficiary Management

**Priority:** Medium | **Category:** Performance

**Description:** Test system performance when adding a large number of beneficiaries via admin portal bulk upload.

**Objective:** Ensure system can handle bulk operations efficiently.

**Test Steps:**
1. Prepare CSV file with 100 unique beneficiaries.
Test Data: file_name: bulk_beneficiaries_100.csv, file_content: 100 rows with fields: Name, Account Number, Type, sample_row: Beneficiary 001,EG1234567890000000000001,Universal Collection, environment: UAT
Expected: CSV file is ready for upload.

2. Upload CSV file via admin portal bulk upload feature.
Test Data: upload_path: /admin/beneficiaries/bulk-upload, file_name: bulk_beneficiaries_100.csv, user_credentials: admin_user4/PerfTest#2024, environment: UAT

Expected: Bulk upload process starts and progress indicator is shown.

3. Measure time taken for upload and processing.
Test Data: start_time: 2024-06-01 10:00:00, end_time: System timestamp after completion, expected_max_duration: 60 seconds, environment: UAT
Expected: All 100 beneficiaries are added within 60 seconds.

## Test Case 18: Verify mandatory field enforcement during Governmental Payment initiation

**Test ID:** TC_031

**Module:** Corporate Module - Governmental Payment Initiation

**Priority:** Critical | **Category:** Functional

**Description:** Test that the system enforces mandatory fields and displays appropriate error messages when attempting to submit a governmental payment with missing required data.

**Objective:** Ensure mandatory fields are enforced and user is prevented from submitting incomplete forms.

**Test Steps:**
1. Attempt to submit the payment form with all fields empty.
Test Data: amount: , beneficiary: , payment_type: , date: , environment: UAT1, user_credentials: corporate_user1/password123, additional_params: browser=Chrome 125
Expected: System displays error messages for all mandatory fields.

2. Fill in only the 'amount' and 'date' fields, leave 'beneficiary' and 'payment_type' empty, then submit.
Test Data: amount: 1500.00, beneficiary: , payment_type: , date: 03-08-2025, environment: UAT1, user_credentials: corporate_user1/password123, additional_params: browser=Chrome 125
Expected: System displays error messages for missing 'beneficiary' and 'payment_type' fields.

3. Fill in all mandatory fields with valid data and submit.
Test Data: amount: 1500.00, beneficiary: BEN123456, payment_type: Tax Collection, date: 03-08-2025, environment: UAT1, user_credentials: corporate_user1/password123, additional_params: browser=Chrome 125
Expected: Payment form is submitted successfully and confirmation message is displayed.

## Test Case 19: Validate SWIFT compliance for free-format fields

**Test ID:** TC_032

**Module:** Corporate Module - Payment Data Entry

**Priority:** High | **Category:** Functional

**Description:** Ensure that only SWIFT-compliant characters are accepted in free-format fields and non-compliant characters are rejected.

**Objective:** Verify field-level validation for SWIFT compliance.

**Test Steps:**
1. Enter a SWIFT-compliant string in the 'Payment Reference' free-format field.
Test Data: payment_reference: INV-2025/08*ABC, environment: UAT2, user_credentials: corporate_user2/securePass!, additional_params: browser=Firefox 120
Expected: Field accepts the input without error.

2. Enter a string with non-compliant characters (e.g., emoji, <, >, &) in the 'Payment Reference' field.
Test Data: payment_reference: Payment■&, environment: UAT2, user_credentials: corporate_user2/securePass!, additional_params: browser=Firefox 120
Expected: System displays an error indicating invalid characters.

3. Attempt to submit the form with the non-compliant string.
Test Data: payment_reference: Payment■&, amount: 1000.00, beneficiary: BEN654321, payment_type: Custom Collection, date: 04-08-2025
Expected: Form submission is blocked and user is prompted to correct the input.

## Test Case 20: Verify auto-rejection of unapproved transactions after 45 days

**Test ID:** TC_034

**Module:** Corporate Module - Transaction Approval Workflow

**Priority:** High | **Category:** Functional

**Description:** Test that transactions not approved within 45 days are automatically rejected by the system.

**Objective:** Ensure compliance with business rule for transaction expiry.

**Test Steps:**
1. Create a governmental payment transaction and leave it pending approval.
Test Data: transaction_id: TXN20250601, amount: 2500.00, beneficiary: BEN789012, payment_type: Universal Collection, date_created: 01-06-2025, environment: UAT3, user_credentials: corporate_user3/Pass#2025, additional_params: browser=Edge 115
Expected: Transaction is created and status is 'Pending Approval'.

2. Advance system date/time by 46 days and run the auto-rejection job.
Test Data: system_date: 17-07-2025, job_name: AutoRejectUnapprovedTransactions, environment: UAT3
Expected: Transaction status is updated to 'Rejected' with reason 'Approval time exceeded 45 days'.

3. Check transaction status and audit logs.
Test Data: transaction_id: TXN20250601, audit_log_query: SELECT * FROM audit_logs WHERE transaction_id='TXN20250601'
Expected: Status is 'Rejected' and audit log records the auto-rejection event.

## Test Case 21: Boundary test for daily single limit parameter

**Test ID:** TC_035

**Module:** Corporate Module - Entitlement Configuration

**Priority:** Critical | **Category:** Boundary

**Description:** Test the system's handling of minimum and maximum allowed values for the daily single limit parameter during entitlement configuration.

**Objective:** Ensure boundary values are correctly enforced and validated.

**Test Steps:**
1. Attempt to set the daily single limit to the minimum allowed value (0.01).

Test Data: product: Governmental Payments, parameter: Daily Single Limit, value: 0.01, environment: AdminPortal-UAT, user_credentials: admin_user2/Secure#456, additional_params: session_id=AP56789
Expected: System accepts the value and saves the configuration.

2. Attempt to set the daily single limit to the maximum allowed value (99999999.99).
Test Data: product: Governmental Payments, parameter: Daily Single Limit, value: 99999999.99, environment: AdminPortal-UAT, user_credentials: admin_user2/Secure#456, additional_params: session_id=AP56789
Expected: System accepts the value and saves the configuration.

3. Attempt to set the daily single limit to an invalid value (100000000.00).
Test Data: product: Governmental Payments, parameter: Daily Single Limit, value: 100000000.00, environment: AdminPortal-UAT, user_credentials: admin_user2/Secure#456, additional_params: session_id=AP56789
Expected: System displays an error and does not save the configuration.

## Test Case 22: Integration test for beneficiary management parity (Admin vs Customer Portal)

**Test ID:** TC_036

**Module:** Corporate Module - Beneficiary Management

**Priority:** High | **Category:** Integration

**Description:** Ensure that beneficiary management (add, view) is consistent between admin and customer portals.

**Objective:** Validate integration and parity of beneficiary management features.

**Test Steps:**
1. Admin adds a new beneficiary 'BEN20250803' via the admin portal.
Test Data: beneficiary_id: BEN20250803, name: ABC Trading Co., account_number: 123456789012, bank_code: EG123, environment: AdminPortal-UAT, user_credentials: admin_user3/Adm1n!2025, additional_params: session_id=AP78901
Expected: Beneficiary is added and visible in the admin portal.

2. Customer logs in and verifies that the new beneficiary is visible in their portal.
Test Data: login_id: corporate_user4, password: Cust#2025, navigation_path: Dashboard > Beneficiaries
Expected: Beneficiary 'BEN20250803' is listed in the customer portal.

3. Customer adds a new beneficiary 'BEN20250804' via the customer portal.
Test Data: beneficiary_id: BEN20250804, name: XYZ Exports Ltd., account_number: 987654321098, bank_code: EG456, environment: CustomerPortal-UAT, user_credentials: corporate_user4/Cust#2025, additional_params: browser=Safari 17
Expected: Beneficiary is added and visible in the customer portal.

4. Admin verifies that the customer-added beneficiary is visible in the admin portal.
Test Data: beneficiary_id: BEN20250804, environment: AdminPortal-UAT, user_credentials: admin_user3/Adm1n!2025
Expected: Beneficiary 'BEN20250804' is listed in the admin portal.

## Test Case 23: Security test for unauthorized entitlement modification

**Test ID:** TC_037

**Module:** Corporate Module - Entitlement Management

**Priority:** Critical | **Category:** Security

**Description:** Test that users without admin privileges cannot modify entitlements for products or sub-products.

**Objective:** Ensure role-based access control is enforced for entitlement changes.

**Test Steps:**
1. Attempt to access the entitlement management page.
Test Data: navigation_path: Admin > Entitlement Management, user_credentials: corporate_user5/NoAdmin#2025, environment: CustomerPortal-UAT
Expected: Access is denied and user is redirected or shown an error message.

2. Attempt to modify entitlements via direct URL manipulation.
Test Data: url: https://uat.corpbank.com/admin/entitlements/edit?customer_id=CorpClient2025, user_credentials: corporate_user5/NoAdmin#2025
Expected: System blocks the action and logs the unauthorized attempt.

3. Check audit logs for the unauthorized access attempt.
Test Data: audit_log_query: SELECT * FROM audit_logs WHERE user_id='corporate_user5' AND action='entitlement_modify_attempt'
Expected: Audit log records the unauthorized attempt with timestamp and user details.

## Test Case 24: Usability test for dynamic field display based on country entitlement

**Test ID:** TC_038

**Module:** Corporate Module - Payment Initiation UI

**Priority:** Medium | **Category:** Usability

**Description:** Test that Tax Collection fields are displayed only when the customer's GCIF country entitlement is set to Egypt.

**Objective:** Ensure dynamic UI behavior based on entitlement configuration.

**Test Steps:**
1. Navigate to the Governmental Payment initiation page.
Test Data: navigation_path: Dashboard > Payments > Governmental Payments, user_credentials: CorpClientEG/Egypt2025, environment: UAT4
Expected: Tax Collection fields are visible on the form.

2. Change GCIF country entitlement to 'UAE' for the same customer.
Test Data: customer_id: CorpClientEG, new_country: UAE, admin_credentials: admin_user4/AdminUAE#2025
Expected: Tax Collection fields are hidden on the payment initiation form.

3. Revert GCIF country entitlement to 'Egypt' and refresh the page.
Test Data: customer_id: CorpClientEG, new_country: Egypt, admin_credentials: admin_user4/AdminUAE#2025
Expected: Tax Collection fields are visible again.

## Test Case 25: Error handling test for invalid date format in payment form

**Test ID:** TC_039

**Module:** Corporate Module - Payment Data Entry

**Priority:** Medium | **Category:** Error Handling

**Description:** Test that the system validates date fields and rejects invalid date formats in the payment form.

**Objective:** Ensure robust error handling for date input.

**Test Steps:**
1. Enter an invalid date format (YYYY/MM/DD) in the 'Date' field and submit.
Test Data: amount: 2000.00, beneficiary: BEN20250805, payment_type: Bill Collection, date: 2025/08/03, environment: UAT5, user_credentials: corporate_user6/Pass2025!, additional_params: browser=Chrome 126
Expected: System displays an error for invalid date format.

2. Enter a non-date string ('August 3rd, 2025') in the 'Date' field and submit.
Test Data: amount: 2000.00, beneficiary: BEN20250805, payment_type: Bill Collection, date: August 3rd, 2025
Expected: System displays an error for invalid date format.

3. Enter a valid date (03-08-2025) and submit.
Test Data: amount: 2000.00, beneficiary: BEN20250805, payment_type: Bill Collection, date: 03-08-2025
Expected: Form is submitted successfully.

## Test Case 26: Field Format and SWIFT Compliance Validation

**Test ID:** TC_043

**Module:** Corporate Module - Payment Entry

**Priority:** Critical | **Category:** Compliance

**Description:** Ensure that free-format fields only accept SWIFT-compliant characters and reject invalid input.

**Objective:** Prevent non-compliant data entry in payment forms.

**Test Steps:**
1. Navigate to governmental payment entry form.
Test Data: navigation_path: Customer Portal > Payments > Governmental Payments > New Payment, environment: https://customer.corpbank.test, user_credentials: corp_user02/securePass!9
Expected: Payment entry form is displayed.

2. Enter valid SWIFT-compliant free-format data.
Test Data: free_format_field: Invoice#2024/EGY-001, allowed_characters: A-Z, 0-9, /, #, -, environment: SWIFT Validator v3.2
Expected: Field accepts input without error.

3. Enter invalid data with non-compliant characters.
Test Data: free_format_field: Invoice@2024$EGY*001, invalid_characters: @, $, *, environment: SWIFT Validator v3.2
Expected: Field displays validation error and rejects input.

## Test Case 27: Boundary Test for Daily Single Limit Field

**Test ID:** TC_044

**Module:** Corporate Module - Entitlement Matrix

**Priority:** High | **Category:** Boundary

**Description:** Verify that the daily single limit field enforces minimum and maximum allowed values.

**Objective:** Ensure correct handling of boundary values for entitlement parameters.

**Test Steps:**
1. Enter minimum allowed value in daily single limit field.
Test Data: daily_single_limit: 0.01, field_type: AMT(12,2), environment: Entitlement Matrix UI, user_credentials: admin_user01
Expected: Field accepts minimum value.

2. Enter maximum allowed value in daily single limit field.
Test Data: daily_single_limit: 99999999.99, field_type: AMT(12,2), environment: Entitlement Matrix UI
Expected: Field accepts maximum value.

3. Enter value exceeding maximum allowed.
Test Data: daily_single_limit: 100000000.00, field_type: AMT(12,2), environment: Entitlement Matrix UI
Expected: Field displays error and rejects input.

## Test Case 28: Role-Based Access Control for Entitlement Assignment

**Test ID:** TC_045

**Module:** Corporate Module - Entitlement Management

**Priority:** Critical | **Category:** Security

**Description:** Test that only users with admin privileges can assign or modify governmental payment entitlements.

**Objective:** Prevent unauthorized entitlement changes.

**Test Steps:**
1. Login as non-admin user and attempt to access entitlement assignment.
Test Data: user_credentials: corp_user03/userPass!7, navigation_path: Admin Portal > Entitlement Matrix, environment: https://admin.corpbank.test
Expected: Access denied error is displayed.

2. Login as admin user and access entitlement assignment.
Test Data: user_credentials: admin_user01/password@123, navigation_path: Admin Portal > Entitlement Matrix, environment: https://admin.corpbank.test
Expected: Entitlement assignment page is accessible.

3. Assign new entitlement to a corporate customer.
Test Data: customer_gcif: 10023456, entitlement_module: Governmental Payments, sub_product: Universal Collection, environment: Entitlement Matrix UI
Expected: Entitlement assigned successfully.

## Test Case 29: Dynamic Field Display Based on Country Entitlement

**Test ID:** TC_046

**Module:** Corporate Module - Payment Entry

**Priority:** Medium | **Category:** Functional

**Description:** Test that Tax Collection fields are only displayed when the customer's GCIF country is Egypt.

**Objective:** Ensure correct dynamic field rendering based on entitlement.

**Test Steps:**
1. Login as Egyptian corporate user and access payment entry.
Test Data: user_credentials: corp_user_egypt/egyptPass!1, customer_gcif: 10023456, country: Egypt, navigation_path: Customer Portal > Payments > Governmental Payments, environment: https://customer.corpbank.test
Expected: Tax Collection fields are displayed.

2. Login as UAE corporate user and access payment entry.
Test Data: user_credentials: corp_user_uae/uaePass!2, customer_gcif: 10098765, country: UAE, navigation_path: Customer Portal > Payments > Governmental Payments, environment: https://customer.corpbank.test
Expected: Tax Collection fields are NOT displayed.

## Test Case 30: Audit Trail Verification for Entitlement Changes

**Test ID:** TC_047

**Module:** Corporate Module - Audit and Logging

**Priority:** High | **Category:** Security

**Description:** Ensure all entitlement changes are logged with correct details in the audit trail.

**Objective:** Maintain traceability and compliance for entitlement changes.

**Test Steps:**
1. Modify entitlement for a corporate customer.
Test Data: user_credentials: admin_user01/password@123, customer_gcif: 10023456, entitlement_module: Governmental Payments, action_type: Add Sub-product: Universal Collection, environment: Entitlement Matrix UI
Expected: Entitlement modification is successful.

2. Access audit log and search for the modification entry.
Test Data: audit_log_id: AUD-20240601-0456, search_criteria: GCIF: 10023456, Action: Add Sub-product, environment: Audit Log Portal
Expected: Audit log entry with correct details is found.