### **Automation Test Cases: Payment Module**

**Objective:** To provide a detailed set of implementable test cases for each of the automated test suites defined in the Test Automation Strategy. These cases are designed for execution within the CI/CD pipeline to provide rapid feedback on build quality and regression status.

#### **1. API Smoke Test Suite**

* **Purpose:** A rapid, critical-path verification suite. A failure in this suite will fail the Jenkins build and prevent deployment.
* **Execution:** Automatically triggered by Jenkins after every successful deployment to the SIT environment.

| **Test Case ID** | **Severity** | **Test Scenario** | **API Endpoint & Method** | **Prerequisites** | **Test Steps** | **Key Assertions** | **Traceability** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AUTO-SMOKE-01** | Blocker | **User Authentication** | POST /user/auth/login | - CONE simulator is running.<br>- paymentMaker user exists. | 1. Send a POST request with valid paymentMaker credentials. | - Assert HTTP status is 200 OK.<br>- Assert response body contains a userToken.<br>- Store the token for subsequent tests. | TDD 4.1.4 |
| **AUTO-SMOKE-02** | Blocker | **Get Standing Orders List** | GET /StandingOrder | - A valid JWT token is available. | 1. Send a GET request with the user token in the user-token header. | - Assert HTTP status is 200 OK.<br>- Assert response body is a valid JSON array. | Postman Collection |
| **AUTO-SMOKE-03** | Blocker | **Get Transactions List** | GET /Transaction | - A valid JWT token is available. | 1. Send a GET request with the user token. | - Assert HTTP status is 200 OK.<br>- Assert response body is a valid JSON array. | Postman Collection |
| **AUTO-SMOKE-04** | Blocker | **SYB Inquiry Connectivity** | POST /externals/check-billing | - A valid JWT token is available.<br>- SYB simulator is running. | 1. Send a POST request with a valid BillingId known to the simulator. | - Assert HTTP status is 200 OK.<br>- Assert response body contains the expected customerName. | SYB Spec |
| **AUTO-SMOKE-05** | Blocker | **DWH Connectivity** | POST /AccountNumber | - A valid JWT token is available.<br>- DWH simulator is running. | 1. Send a POST request with a valid accountNumber known to the simulator. | - Assert HTTP status is 200 OK.<br>- Assert response body contains the expected accountName and accountStatus. | TDD 4.1.4.6 |

#### **2. Full API Regression Suite**

* **Purpose:** An exhaustive validation of all API endpoints, business logic, and error handling.
* **Execution:** Automatically triggered by a nightly Jenkins job.

##### **2.1. Standing Order Lifecycle**

| **Test Case ID** | **Severity** | **Test Scenario** | **API Endpoint & Method** | **Prerequisites** | **Test Steps** | **Key Assertions** | **Traceability** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AUTO-API-REG-01** | Critical | **Create & Approve SO (Happy Path)** | POST /StandingOrder<br>PUT /StandingOrder/:id/approved | - Maker and Checker JWT tokens.<br>- OVS-BAS sim ready for successful fee debit. | 1. **(Maker)** Create a new SO.<br>2. Extract the id from the response.<br>3. **(Checker)** Approve the SO using the extracted id. | - Assert SO creation returns 200 OK.<br>- Assert approval returns 200 OK.<br>- Assert the SO status in the DB is now "Approved". | Payment Flow: A |
| **AUTO-API-REG-02** | Major | **Create SO with Invalid Billing ID** | POST /StandingOrder | - Maker JWT token.<br>- SYB sim configured to reject the Billing ID. | 1. **(Maker)** Attempt to create an SO with a billingId the simulator will reject. | - Assert HTTP status is 400 Bad Request.<br>- Assert response contains "Billing ID not found". | SYB Spec |
| **AUTO-API-REG-03** | Major | **Approve SO with Insufficient Funds for Fee** | PUT /StandingOrder/:id/approved | - Checker JWT token.<br>- OVS-BAS sim configured to fail the fee debit. | 1. **(Checker)** Attempt to approve an SO linked to an account with no funds. | - Assert HTTP status is 400 Bad Request.<br>- Assert response contains "Insufficient funds for registration fee". | FDD 7.1.1 |
| **AUTO-API-REG-04** | Critical | **Amend & Approve an SO** | PUT /StandingOrder/:id<br>PUT /StandingOrder/:id/approved | - Maker and Checker tokens.<br>- An existing "Approved" SO. | 1. **(Maker)** Update the accountNumberDebit on the approved SO.<br>2. **(Checker)** Approve the amendment. | - Assert the PUT call changes approvalStatus to "Submitted" and applicationStatus to "Amendment".<br>- Assert the approval call changes approvalStatus back to "Approved". | UI Design Spec |
| **AUTO-API-REG-05** | Critical | **Terminate & Approve an SO** | PUT /StandingOrder/:id<br>PUT /StandingOrder/:id/approved | - Maker and Checker tokens.<br>- An existing "Approved" SO. | 1. **(Maker)** Update the SO with applicationStatus: "Terminate" and a finalDebit date.<br>2. **(Checker)** Approve the termination. | - Assert the PUT call changes approvalStatus to "Submitted" and applicationStatus to "Terminate".<br>- Assert the approval call changes approvalStatus to "Approved". | UI Design Spec |

##### **2.2. Payment Execution & Transactional Logic**

| **Test Case ID** | **Severity** | **Test Scenario** | **API Endpoint & Method** | **Prerequisites** | **Test Steps** | **Key Assertions** | **Traceability** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AUTO-API-EXEC-01** | Critical | **Successful Payment Execution** | POST /payments/execute-job | - An approved SO is due for payment.<br>- All simulators configured for success. | 1. Trigger the payment worker job.<br>2. Poll GET /Transaction/:id until status is final. | - Assert the final transaction status is "Paid".<br>- Verify calls were made to SYB Inquiry, OVS Hold, SYB Payment, and OVS Debit. | Payment Flow: B |
| **AUTO-API-EXEC-02** | Major | **Handle "Balance Short" from OVS** | POST /payments/execute-job | - An approved SO is due.<br>- OVS sim configured to fail Hold Amount. | 1. Trigger the payment worker job.<br>2. Poll GET /Transaction/:id. | - Assert the final transaction status is "Balance Short".<br>- Verify no Payment call was made to SYB. | OVS Flow |
| **AUTO-API-EXEC-03** | Major | **Handle "Bill Already Paid" from SYB** | POST /payments/execute-job | - An approved SO is due.<br>- SYB sim configured to return rc: "54". | 1. Trigger the payment worker job.<br>2. Poll GET /Transaction/:id. | - Assert the final transaction status is "Paid by another merchant".<br>- Verify no calls were made to OVS Hold or Payment. | SYB Spec |
| **AUTO-API-EXEC-04** | Critical | **Handle SYB Payment Timeout** | POST /payments/execute-job | - An approved SO is due.<br>- SYB sim configured to timeout on the /payment call. | 1. Trigger the payment worker job.<br>2. Poll GET /Transaction/:id. | - Assert the final transaction status is "Payment Suspect".<br>- **(Future Test)** Verify the Advice flow is triggered to resolve the status. | SYB Spec |
| **AUTO-API-EXEC-05** | Major | **Handle Dormant Account from OVS** | POST /payments/execute-job | - An approved SO is due.<br>- OVS sim configured to fail Hold Amount with a "Dormant" error. | 1. Trigger the payment worker job.<br>2. Poll GET /Transaction/:id. | - Assert the final transaction status is "Payment Failed".<br>- Assert the logged reason is "Dormant Account". | Your Requirements |

#### **3. UI Regression Suite**

* **Purpose:** A targeted validation of critical end-to-end user workflows to catch regressions in the UI or its integration with the backend.
* **Execution:** Automatically triggered by a nightly Jenkins job.

| Test Case ID | Severity | Test Scenario | User Flow | Prerequisites | Key Assertions | Traceability |

| :--- | :--- | :--- | :--- | :--- | :--- |

| AUTO-UI-REG-01 | Critical | Golden Path: Register & Approve SO | 1. Maker logs in.<br>2. Navigates to "Add Standing Order".<br>3. Fills and submits the form.<br>4. Maker logs out.<br>5. Checker logs in.<br>6. Navigates to Task List and approves the SO. | - Maker and Checker users exist.<br>- All simulators are running. | - Assert a success notification is shown after submission.<br>- Assert the SO appears in the Checker's Task List.<br>- Assert the SO is removed from the Task List after approval.<br>- Assert the SO's status is "Approved" on the main list page. | UI Design Spec |

| AUTO-UI-REG-02 | Major | Golden Path: Manual Repair of Failed Transaction | 1. A transaction with "Balance Short" status exists.<br>2. Maker logs in.<br>3. Navigates to the "Transaction" page, "Need Repair" tab.<br>4. Finds the transaction and clicks "Repair".<br>5. Confirms the repair request.<br>6. Checker logs in and approves the repair request. | - A "Balance Short" transaction exists.<br>- OVS sim is now configured to succeed. | - Assert the repair request appears in the Checker's Task List.<br>- Assert the transaction status eventually becomes "Paid" after the worker job runs. | UI Design Spec |

| AUTO-UI-REG-03 | Minor | Verify Filtering on SO List Page | 1. User logs in.<br>2. Navigates to the "Standing Order" page.<br>3. Opens the filter panel.<br>4. Filters by Approval Status: "Approved".<br>5. Clicks "Apply". | - Assert that all records displayed in the table have "Approved" in the status column.<br>- Assert the total record count matches the expected number of approved SOs. | UI Design Spec |

| AUTO-UI-REG-04 | Minor | Verify Form Validation | 1. Maker logs in.<br>2. Navigates to "Add Standing Order".<br>3. Clicks "Submit" without filling any fields. | - Assert that validation error messages appear below all mandatory fields.<br>- Assert that the form submission is blocked. | UI Design Spec |