#### Initiate Payment from Merchant’s Website/App

**User Story**:

**As a User,**

**I want to** initiate a payment request from the merchant's website or app,

**So that** I can be redirected to the Agrani Bank Payment Gateway to continue the payment process securely.

**Acceptance Criteria:**

1. **Display Payment Information on Merchant Page**

* **Given** the customer is on the merchant’s website or application,
* **When** the payment form is displayed,
* **Then** it should show:
  + Customer/Applicant’s Name
  + PIN
  + Application Fee
  + Total Amount
  + Button labeled “আবেদন ফি জমা দিন” (Pay Application Fee)

1. **Redirect to Agrani Bank Payment Gateway**

* **Given** the customer clicks the “আবেদন ফি জমা দিন” button on the merchant page,
* **When** the click event is triggered,
* **Then** the system should seamlessly redirect the customer to the Agrani Bank Payment Gateway page.

1. **Preserve Payment Context During Redirection**

* **Given** the redirection occurs,
* **When** the Agrani Bank Payment Gateway loads,
* **Then** the previously submitted payment context (amount, name, etc.) should be preserved and displayed accurately.

#### Payment Summary

**User Story**:

**As a User,**

**I want to** view the payment summary and all available payment channel methods after redirection to the Bank Payment Gateway (BPG),

**So that** I can select my preferred payment method.

**Acceptance Criteria:**

1. **Display Payment Summary Details**

* **Given** the customer is redirected from the merchant website to the **Bank Payment Gateway**,
* **When** the page loads,
* **Then** the customer should see the following details clearly displayed:
  + Total payment amount (without any applicable charges at this stage)
  + Customer name
  + Customer mobile number
  + Bank reference number
  + Merchant reference number
  + Merchant name

1. **Display Available Payment Channels**

* **Given** the customer is redirected from the merchant website to the **Bank Payment Gateway**,
* **When** the page loads,
* **Then** the customer views the available payment options,:
  + **Agrani Smart Banking** (no charges apply)
  + **Mobile Banking and Card** via Shurjopay (charges to be shown later in the payment flow)

1. **Redirect to Selected Payment Flow**

* **Given** the customer selects a payment channel from the available options,
* **When** the selection is made,
* **Then** the system should redirect the customer to the corresponding payment flow:
  + Redirect to **Shurjopay flow** if "Mobile Banking & Card" is selected, where applicable charges will be displayed
  + Redirect to **Agrani Smart Banking IAM login flow** if "Agrani Smart Banking" is selected, with no additional charges shown

#### Pay Using Shurjopay

**User Story**:

**As a User,**

**I want to** complete payment via **Mobile Banking or Card payment** (Shurjopay),

**So that**  I can pay conveniently using cards or mobile wallets.

**Acceptance Criteria:**

1. **Redirect to Shurjopay Payment Gateway**

* **Given** the user is on the Payment Channel selection page,
* **When** the user selects the **Mobile Banking & Card** option,
* **Then** the system should:
  + Navigate to a new screen displaying:
    - **Main Amount**
    - **Processing Fee**
    - **Total Amount**
  + Include “**Back”** and **“Confirm**” buttons.
* **And** when the user clicks **Confirm** button,
* **Then** the system should redirect the user to the external **Shurjopay** gateway along with:
  + **Transaction metadata** (order ID, amount, merchant info)

1. **Complete Shurjopay Payment Flow**

* **Given** the customer is redirected to Shurjopay,
* **When** the customer completes the payment steps (entering required details, authentication, etc.),
* **Then** the payment should be processed through Shurjopay’s secure flow.

1. **Post-Shurjopay Redirection (After Payment Completion)**

* **Given** the user completes the payment on **Shurjopay** (successfully or unsuccessfully),
* **When** Shurjopay redirects the user back,
* **Then** the system should:
  + Redirect the user to the **payment confirmation page**
  + Show a **transaction success or failure status** based on the Shurjopay response
  + Include:
    - Application PIN
    - Applicant Name
    - Transaction ID / Reference
    - Paid Amount

1. **Security and Integrity**

* **Given** a redirection from Shurjopay is received,
* **When** the transaction reference number is processed,
* **Then** the system should verify:
  + That the reference number is valid
  + That the transaction status is not tampered
  + And only then display the success/failure screen

#### 

#### 

#### Pay Using Agrani Smart Banking

**User Story**:

**As a User,**

**I want to** pay using my Agrani Smart Banking

**So that** I can securely complete the payment using my bank accounts registered with Agrani Bank.

**Acceptance Criteria:**

1. **Redirect to Payment Summary**

* **Given** the customer selects **Agrani Smart Banking option** on the payment channel selection page,
* **When** the customer is redirected,
* **Then** the payment summary page should display:
  + Main payment amount
  + Processing Fee (৳0.00)
  + Total Amount
  + “Back” and “Confirm” buttons

1. **Confirm Payment Summary and Navigate to Login**

* **Given** the payment summary page is displayed,
* **When** the user clicks the **Confirm** button,
* **Then** the system should redirect to the **Agrani Bank Identity Provider Login page**
* **And** show:
  + **User ID input field**
  + **Password field**
  + **“Login” button**

1. **Login Authentication**

* **Given** the customer is on the **Login page**,
* **When** the customer enters valid credentials and clicks Login,
* **Then** the system should authenticate the customer,
* **And** navigate to the Account Selection screen.

1. **Account Selection**

* **Given** the customer is on the **Account Selection** screen,
* **When** the customer selects an account from the dropdown,
* **Then** the system should show:
  + Account type(savings/current)
  + Account number (masked except last 5 digits, e.g., \*\*\*00457)
  + Available account balance
  + Edit icon for account selection
* **And** the **Next** button should become enabled once an account is selected

1. **OTP Verification Screen**

* **Given** the customer clicks Next after selecting an account,
* **When** the OTP Verification screen appears,
* **Then** the system should:
  + Display 6 input fields accepting only digits for OTP
  + Show a 2-minute countdown timer
  + Display Cancel and Confirm buttons (Confirm disabled until all 6 digits entered)

1. **Resend OTP Functionality**

* **Given** the timer expires,
* **When** the user clicks the **Resend** button,
* **Then** the system should:
  + Send a new OTP to the registered mobile number
  + Invalidate the previous OTP
  + Restart the 2-minute countdown

1. **OTP Validation and Confirmation**

* **Given** the user enters a valid 6-digit OTP and clicks **Confirm**,
* **When** the OTP is verified,
* **Then** the system should:
  + Complete the transaction
  + Navigate to the **Payment Completion Confirmation screen**

1. **Payment Completion Confirmation Screen**

* **Given** the payment is completed successfully,
* **When** the Payment Completion screen is displayed,
* **Then** it should show:
  + Message: “Payment Completed”
  + Amount Paid (e.g., BDT 521.15)
  + Reference ID
  + Info message: auto-redirect to merchant in 10 seconds
  + A clickable fallback link (e.g., “click here”)

1. **Redirect Back to Merchant Site**

* **Given** the confirmation screen is shown,
* **When** 10 seconds pass or the user clicks the fallback link,
* **Then** the system should redirect back to the merchant site
* **And** show a post-payment screen including:
  + PIN
  + Applicant Name
  + Transaction ID
  + Paid Amount

1. **Navigation Controls**
2. **Back Button on Payment Summary Page**

* **Given** the customer is on the **Agrani Smart Banking Payment Summary** page,
* **When** the customer clicks the **Back** button,
* **Then** the system should navigate the customer back to the **payment channel selection** screen.

1. **Cancel Button on OTP Verification Page**

* **Given** the customer is on the OTP Verification page,
* **When** the customer clicks the Cancel button,
* **Then** the payment process will be terminated and the customer would be redirected back to the merchant site.

#### 

#### See Charge/Fee Details for Payment

**User Story**:

**As a User,**

**I want to** see the charge/fee details for my payment,

**so that** I understand the total cost based on my selected payment method (Shurjopay or Agrani Smart Banking).

**Acceptance Criteria:**

1. **Fee Breakdown for Shurjopay**

* **Given** the customer selects **Mobile Banking & Card (Shurjopay)** as the payment method,
* **When** the payment summary page is displayed,
* **Then** the system should show:
  + Original payment amount (as received from the merchant)
  + Shurjopay’s **processing fee** (based on a flat fee or percentage)
  + **Total payable amount** = Original payment amount + Processing fee
  + "Back" and "Confirm" buttons for navigation

1. **Fee Breakdown for Agrani Smart Banking**

* **Given** the customer selects **Agrani Smart Banking** as the payment method,
* **When** the payment summary page is displayed,
* **Then** the system should show:
  + Original payment amount (from merchant)
  + **Processing fee** as ৳0.00 (if no charge applies based on bank’s profile)
  + **Total payable amount** = Original payment amount
  + "Back" and "Confirm" buttons for navigation

1. **Customer Review and Action**

* **Given** the payment summary and fee details are displayed,
* **When** the customer reviews the charges,
* **Then** the customer should have the ability to:
  + Click **Confirm** to proceed with payment
  + Click **Back** to return to the previous payment method selection page
  + (For Shurjopay only) be redirected to the Shurjopay gateway upon confirmation

#### Confirm Payment via Channel Selection

**User Story**:

**As a User,**

**I want to** confirm payment via channel selection,

**so that** I can pay using my preferred gateway option with the updated payment amount including applicable charges.

**Acceptance Criteria:**

1. **Channel Selection Updates Payment Details**

* **Given** the user is on the **Payment Channel Selection** page,
* **When** the user selects a payment channel (e.g., **Agrani Smart Banking** or **Mobile Banking & Card [Shurjopay]**),
* **Then** the system should:
  + Navigate to the corresponding channel-specific payment summary screen
  + Dynamically calculate and display any applicable charges for that channel
  + Show the updated total payable amount (Original Amount + Channel Fees)

1. **Display Updated Payment Amount**

* **Given** the selected payment method includes additional charges,
* **When** the updated total is calculated,
* **Then** the screen should:
  + Clearly display:
    - Main Amount
    - Processing Fee
    - Total Payable Amount
* **And** the user can review the final cost before proceeding

1. **Confirmation Action Enabled After Selection**

* **Given** the updated payment details are displayed on the payment summary page,
* **When** the page loads (regardless of additional input),
* **Then** the **"Confirm"** button should be **enabled by default**
* **And** clicking the button should:
  + Redirect to the Agrani Smart Banking login flow if that channel was selected
  + Or redirect to the Shurjopay gateway if that channel was selected

1. **Cancellation Option Available**

* **Given** the user is reviewing the payment details on the summary screen,
* **When** the user decides not to proceed,
* **Then** the user should be able to:
  + Click the **"Back"** button to return to the previous screen **(Payment Channel Selection Page**)
  + Without initiating the payment process

#### 

#### Authenticate via Agrani Smart Banking Payment

**User Story**:

**As a user,**

**I want to** authenticate myself using my Agrani Smart Banking credentials,

**so that** my identity and transaction are securely verified before proceeding with the payment.

**Acceptance Criteria:**

1. **Request User ID and Password**

* **Given** the user selects **Agrani Smart Banking** as the payment method and reaches the login page,
* **When** the login screen is displayed,
* **Then** the system should show:
  + Input fields for **User ID** and **Password**
  + Password field should mask the input characters
  + The **Login button** should remain **disabled** until both fields are filled

1. **Validate User Role (IBUSER Role Check)**

* **Given** the user submits valid credentials,
* **When** the system verifies login permissions,
* **Then** it should confirm the user has the **IBUSER** role required for payment authorization

1. **Check User Block Status**

* **Given** the user has a valid role,
* **When** the system checks the account status,
* **Then** it should ensure the user is **not blocked** from using Smart Banking

1. **Verify Temporary Login Suspension**

* **Given** the user is not permanently blocked,
* **When** the system checks temporary login restrictions,
* **Then** it should prevent access if the user is **temporarily suspended** (e.g., due to multiple failed attempts)

1. **Enforce Geographic Login Restrictions**

* **Given** all previous validations pass,
* **When** the user attempts login,
* **Then** the system should apply any **geographic restrictions** based on IP or region, and block access from unauthorized locations

1. **Authentication Failure Handling**

* **Given** the login attempt fails due to:
  + - Invalid credentials
    - Blocked status
    - Suspension
    - Role mismatch
    - Geographic restriction
* **When** the system detects any of these issues,
* **Then** the user should:
  + - Remain on the login page
    - See a **clear and specific error message**
    - Be prevented from proceeding to the next step

1. **Successful Authentication**

* **Given** the user provides valid credentials, and all validations pass,
* **When** authentication is completed,
* **Then** the system should:
  + - Allow the user to proceed to the **Account Selection page**
    - **Pre-fill** any known account details if available

#### Preprocess Payment Request

**User Story**:

**As a user,**

**I want to** select my preferred bank account number,

#### so that I can complete the payment securely and accurately using that account.

Acceptance Criteria:

1. **Account Existence and Status Check**

* **Given** the user is selecting a bank account from the dropdown list,
* **When** the system validates the selected account,
* **Then** it must reject any account marked as **Closed**, **Dormant**, **Frozen**, or **Inactive**,
* **And** display a clear message such as: *"Selected account is not active for transactions."*

1. **Account Type Validation**

* **Given** the selected account is active,
* **When** the system checks the account type,
* **Then** only the following types should be allowed:
  + - **Savings or Current accounts**
    - **With product codes 1001, 6001, 6019, or 6020**
* **And** all other types should be rejected with a message like: *"Selected account type is not allowed for this transaction."*

1. **Account Operative & Restrictions Check**

* **Given** the account type is valid,
* **When** the system checks operative flags,
* **Then** it must confirm:
  + - **There are no posting restrictions (e.g., stop payments, liens)**
    - **The account supports debit/outgoing transactions**
    - **Else**, show: "Selected account has transaction restrictions."

1. **Mobile Number Validation for OTP**

* **Given** the user selects a valid account,
* **When** the system checks for a registered mobile number,
* **Then** it must ensure a **valid and active mobile number** exists in CBS for OTP delivery
* **And** if not present or inactive, display an appropriate error.

1. **Transaction Profile Limit Validation**

* **Given** the mobile number is valid,
* **When** the system checks the user’s IB transaction profile,
* **Then** it must:
  + Validate against minimum/maximum transaction limits
  + Validate against daily/monthly cumulative limits
* **And** reject transactions exceeding these limits with informative messages.

1. **Merchant Session Time Validation**

* **Given** a valid session token is attached to the transaction,
* **When** the payment request is received,
* **Then** the system must:
  + - **Check token expiry time and authenticity**
* **And** reject transactions if the session is invalid or expired with an appropriate error message like "Your payment session has expired. Please restart the process."

1. **OTP Authorization Initiation**

* **Given** all validations pass successfully,
* **When** the user clicks **Next**,
* **Then** the system should:
  + **Proceed to the OTP Verification screen**
  + **Generate and send a 6-digit OTP to the registered mobile**
  + **Start a 2-minute countdown timer (configurable)**
  + **Ensure OTP is single-use and expires on timeout**

1. **Error Handling**

* Given any of the validations fail at any stage,
* When the failure is triggered,
* Then the system must:
  + **Display an informative error message near the account dropdown or in a modal**
  + **Prevent navigation to the OTP screen**

1. **UI and Button Behavior**

* **Given** no account is selected from the dropdown,
* **When** the account selection screen is displayed,
* **Then** the **"Next"** button should remain **disabled**.
* **Given** the user selects a **valid account** from the dropdown,
* **When** the selection is made,
* **Then** the **"Next"** button should become **enabled**,
* **And** a **✓ checkmark** should appear beside the selected account.

#### Authorize Payment by OTP

**User Story**:

**As a user,**

**I want to** authorize my payment by entering an OTP

**so that** my transaction is securely completed.

**Acceptance Criteria:**

1. **OTP Input & UI Validation**

* **Given** the user is on the OTP input screen,
* **When** the OTP screen loads,
* **Then** the system should display:
  + Six OTP input boxes (accepting numeric input only)
  + A countdown timer (e.g., 2 minutes)
  + A Cancel button
  + A Confirm button (initially disabled until all 6 digits are entered)

1. **OTP Resend and Timer Control**

* **Given** the OTP timer has expired,
* **When** the user clicks the **Resend** button,
* **Then** the system should:
  + Send a new OTP to the registered mobile number
  + Reset the timer (e.g., back to 2 minutes)
  + Allow OTP entry again from scratch

1. **Incorrect OTP Entry**

* **Given** the user enters an incorrect OTP,
* **When** the OTP is submitted,
* **Then** the system should display the message:  
   **“Invalid OTP, please try again.”**

1. **Expired OTP Submission**

* **Given** the OTP has expired before confirmation,
* **When** the user attempts to submit the expired OTP,
* **Then** the system should display the message:  
   **“OTP expired, please request a new one.”**

1. **OTP Attempt Limit and Transaction Blocking**

* **Given** the user enters incorrect OTP **three times consecutively**,
* **When** the third incorrect attempt is submitted,
* **Then** the system should:
  + Block the transaction
  + Show an error message:  
     **“Transaction blocked due to too many incorrect attempts.”**

1. **Successful OTP Verification**

* **Given** the user enters a valid 6-digit OTP within the time and attempt limits,
* **When** the OTP is verified,
* **Then** the system should:
  + Re-confirm the account balance including charges
  + Enforce transaction profile limits (max/min/payment frequency)
  + Check merchant session validity
  + Proceed with transaction completion

1. **Tamper-Proof Validation and Rollback**

* **Given** the OTP is valid and verification succeeds,
* **When** the transaction fails due to a backend or balance/limit issue,
* **Then** the system should:
  + Trigger rollback to revert any partial changes
  + Show a message: **“Payment failed.”**

#### Request a New OTP

**User Story**:

**As a user,**

**I want to** request a new OTP if I didn’t receive the original one or if it expired,

**so that I can co**ntinue and complete my payment securely.

**Acceptance Criteria:**

1. **Availability of Resend OTP Option**

* **Given** the user is on the OTP verification screen,
* **When** the countdown timer reaches **00:00** or the previous OTP is invalid/expired,
* **Then** the system should **display a "Resend" button** clearly and accessibly.

1. **Generation and Delivery of New OTP**

* **Given** the user clicks the "Resend" button,
* **When** the system processes the request,
* **Then** it should generate a **new unique 6-digit OTP**
* **And** send it via **SMS to the registered mobile number**.

1. **Invalidation of Previous OTP**

* **Given** a new OTP is generated,
* **When** the previous OTP was still within its validity period,
* **Then** the system should immediately **invalidate the old OTP** to ensure security.

1. **OTP Expiration Timer Reset**

* **Given** a new OTP is sent to the user,
* **When** the resend is successful,
* **Then** the system should **reset and restart the countdown timer** (e.g., 2 minutes or 5 minutes, as per system config).

1. **Limit on Number of Resend Attempts**

* **Given** the user requests to resend OTP multiple times,
* **When** the number of resend attempts reaches **the configured limit (e.g., 3 times)**,
* **Then** the system should:
  + Block further resend attempts,
  + Cancel the transaction,
  + And display the message:  
     "Too many OTP resend attempts. This transaction has been blocked."

1. **User Confirmation on Resend Request**

* **Given** the system successfully sends the new OTP,
* **When** the process completes,
* **Then** the user should see a confirmation message:
  + "A new OTP has been sent to your registered mobile number."

1. **Prevent Abuse and Security Checks**

* **Given** there are excessive OTP resend attempts across multiple sessions or users,
* **When** security thresholds are breached,
* **Then** the system should **log the activity and prevent abuse**,  
   possibly **flagging the transaction/session for manual review**.

1. **UI/UX Timer and Button State**

* **Given** the OTP timer is active (counting down),
* **When** the timer is still running,
* **Then** the "Resend OTP" button should be **disabled or hidden** until the countdown reaches zero.

#### 

#### 

#### Cancel a Payment

**User Story:**

**As a User,**

**I want to** cancel a payment before confirmation,

**so that** I can stop an unintended transaction.

**Acceptance Criteria**:

1. **Cancel Transaction Action**

* **Given** the user is on **OTP verification screen**,
* **When** the user clicks the **"Cancel"** button,
* **Then** the system should **immediately abort the transaction**,
* **And** redirect the user back to the **merchant’s website**.

1. **No Payment Processing**

* **Given** the transaction is canceled before final confirmation (OTP or Shurjopay submission),
* **When** the cancellation occurs,
* **Then** **no banking or payment transaction** (including debit, authorization, or charge) should be executed.

1. **Agrani Smart Banking Specific Flow**

* **Given** the user selected **Agrani Smart Banking** as the payment method,
* **When** the user cancels the transaction at any step before OTP confirmation,
* **Then** the system should:
  + Stop any further processing
  + Not initiate account verification or debit
  + Redirect the user back to the merchant site.

1. **Shurjopay Flow Specific Handling**

* **Given** the user selected **Shurjopay (Mobile Banking or Card)**,
* **When** the user cancels before confirming payment on the Shurjopay gateway,
* **Then** the system should:
  + Stop redirecting to the Shurjopay interface (if canceled before it loads)
  + Or exit the payment session if already inside the Shurjopay UI
  + **Ensure no wallet or card transaction is triggered**
  + Redirect the user back to the merchant site.

1. **Logging and Audit**

* **Given** the transaction is canceled by the user,
* **When** the cancel button is triggered,
* **Then** the system should:
  + **Log the cancellation event** with user ID (if logged in), reference ID, timestamp, and reason (user-initiated)
  + Ensure the cancellation entry is included in **audit trails and system monitoring logs** for traceability.

#### 

#### Send Confirmation of Transaction to Merchant

**User Story**:

**As a User,**

**I want to** receive confirmation of user payment,

**so that** I know whether the payment was successful.

**Acceptance Criteria**:

1. Display Payment Confirmation Screen

* **Given** the user has successfully confirmed the payment via OTP,
* **When** the transaction is completed,
* **Then** the system should show a confirmation screen containing:
  + A green-highlighted **“Payment Completed”** message
  + The **paid amount** (e.g., BDT 521.15)
  + The **reference ID** (e.g., 7159047574596245)
  + A message: *“***Redirecting to the Merchant Site in 10 seconds…***”*
  + A fallback link: *“***click here”** in case auto-redirect fails

1. **Automatic Redirection to Merchant Site**

* **Given** the confirmation screen is visible,
* **When** 10 seconds pass or the user clicks the manual redirect **fallback link**,
* **Then** the system should redirect the user to the **merchant success page** with transaction reference attached in the URL.

1. Send Payment Confirmation Notification to Merchant

* **Given** the payment reaches a final status (Success or Failure),
* **When** the transaction is processed,
* **Then** the system should send a notification to the merchant using:
  + **Webhook callback** to the merchant server (primary and preferred)
  + **Email** to the registered merchant email address (optional backup)
  + **SMS** to merchant contact number (optional if configured)

1. **Include Key Transaction Data in Notification**

* **Given** the merchant receives the payment notification,
* **When** they parse or view the data,
* **Then** the payload should include:
  + Transaction Status: **Success / Failure**
  + Payment Amount (BDT)
  + Bank Reference Number
  + Merchant Reference Number
  + Transaction Date & Time
  + Customer Name
  + Masked Mobile Number (e.g., 017\*\*\*\*302)
  + Payment Channel (e.g., Agrani Smart Banking, Shurjopay)

1. **Confirmation for Failed Transactions**

* **Given** a payment fails (e.g., insufficient funds, invalid OTP),
* **When** the status is finalized,
* **Then** the merchant should receive a notification indicating:
  + **Failure Status**
  + **Reason Code / Description** if available
  + Same structure as success but marked with failure flag

1. **Security and Privacy Compliance**

* **Given** sensitive payment information is transmitted,
* **When** the system sends confirmation data,
* **Then** it must:
  + Mask customer PII (e.g., phone numbers)
  + Encrypt API/webhook payloads using HTTPS or other secure protocols
  + Avoid storing or transmitting raw OTP or account numbers

#### 

#### Validate Transaction Status by Reference ID

**User Story**:

**As a Merchant,**

**I want to** validate the status of user payment,

**so that** I know whether the payment was successful.

**Acceptance Criteria:**

1. **Accept and Validate Reference ID**

* **Given** the merchant provides a transaction reference ID (e.g., Bank Ref or Merchant Ref),
* **When** the system receives the validation request,
* **Then** the system should:
  + Verify that the reference ID exists
  + Confirm that the transaction is associated with the requesting merchant
  + Return an error if the reference ID is invalid or unauthorized

1. **Return Current Transaction Status**

* **Given** the reference ID is valid and authorized,
* **When** the lookup is successful,
* **Then** the system should return the **current status** of the transaction:
  + **Success**
  + **Failed**
  + **Pending**

1. **Provide Transaction Details on Success**

* **Given** the transaction status is Success,
* **When** the response is generated,
* **Then** it must include the following data:
  + Paid amount (e.g., BDT 520.00)
  + Date and time of payment completion
  + Payment channel used (e.g., Agrani Smart Banking, Shurjopay)
  + Transaction reference ID
  + Merchant reference ID (if applicable)

1. **Secure Access Control**

* **Given** transaction data is sensitive,
* **When** a merchant requests a status check,
* **Then** the system must:  
  + Ensure the reference ID belongs to the requesting merchant
  + Reject unauthorized or mismatched requests with a message like:  
    "Reference ID does not belong to your account"

1. **Error Handling for Invalid Reference**

* **Given** the reference ID is not found, malformed, or expired,
* **When** the system fails to locate a valid transaction,
* **Then** it must return:
  + A clear and user-friendly error message, such as:
  + "Transaction not found. Please check the reference ID and try again."