Business Requirement Mapping

|  |  |  |
| --- | --- | --- |
| Epic 1 | FS\_TX 3 | Details |
| 1 | [3.1.1](#_Order_ID_Generation) | Transaction number Generation |
| 2 | [3.1.2](#_Order_State_Management) | Payment status Management |
| 3 | [3.1.4](#_Merchant_Order_Retry) | Payment Retry Functionality |
| 4 | [3.1.5](#_Operations) | Operations |

## Epic Name: Transaction Processing Module for SBI ePay

### **Description**

This Epic covers the implementation of Transaction ID (ATRN) creation and Payment Processing module for SBI ePay. It handles entire Payment cycle starting from Payment initiation, processing and status transitions (Creation, Paid, Expired, Failed, Refund, Settled).

### Actors

1. Merchant
2. Aggregator
3. Customer

### Objective

* **Payment initiation:**

An ATRN number will be generated for every transaction linked to an order, ensuring that each transaction can be uniquely identified across the merchant’s system.

* **Payment Processing:** Manage the execution of Payment considering various components of Payment such as ATRN, Payment mode, PG, Amount, charges, Order info, Merchant and customer info.

### Functional Details

#### Transaction ID Generation (ATRN)

* **Description:** Once Customer enter payment details on ePay page, Aggregator generates ATRN **(Aggregator Transaction Reference Number)** for payment processing.
* This ATRN will serve as a unique identifier for tracking and reconciling transactions within the system, ensuring that each transaction is distinct and easily traceable.

**Technical Considerations:**

* + Ensure uniqueness at the database level (apply unique constraints).
  + Design a sequence generation mechanism that can handle a high volume of transactions without collisions.
  + Customer should fill required details to generate ATRN
  + The ATRN number's length will be a minimum of 20 characters to a maximum of 30 characters. This length will be changed in incremental way once ATRN is generated.
  + The ATRN number will be an alphanumeric value, which may include a combination of letters (A-Z, a-z) and numbers (0-9).

#### Transaction State Management

* Defines and manages the states of an Transaction:
  + **Paid**
  + **Refund**
  + **Settled**
  + **Failed**
  + **Cancel**
* Final transaction state should be maintained in DB

#### Paid

* **Description:** Final state indicating that the payment has been successfully processed.
* **Transition Conditions:**
  + Once payment is captured by ePay, the transaction transitions to "Paid."
* **Technical Considerations:**
  + Ensure finality and immutability after payment is completed.
  + Trigger downstream processes if there is anything

#### Failed

* **Description:** Indicates that the payment was unsuccessful after multiple attempts.
* **Transition Conditions:**
  + Transition occurs from "Booked" to “Failed” after exceeding the maximum number of retries, or if the payment fails repeatedly.
  + If transaction token is expired and no successful payment attempt is captured, all the ATRN(s) status will be marked as failed
* **Technical Considerations:**
  + Clear error reporting to merchants and users.
  + Wrong response received from Merchant that includes field level validation, Encryption
  + Ensure proper logging and tracking of payment attempts, including failure reasons.

#### Settled

**Description:** Indicates that the payment is successful and ePay received confirmation from PG so that status changes from “Paid” to “settled”

**Transition Conditions:**

* + - * Transition occurs from "Paid" to “Settled” when payment confirmation received from Issuer’s bank via PG.
      * If there is no discrepancy in the Payment details entered by customer and actual details with the issuer bank, the status Payment transitions to “settled”.

**Technical Considerations:**

* Clear error reporting to merchants and users.
* Wrong response received from Merchant that includes field level validation, Encryption

#### Refund

* **Description:** Merchant can initiate refund if transaction reaches its end state (Paid/ Settled)
* **Transition Conditions:**
  + The transaction can move from “Booked” to "Refund" status, if transaction is already in “Paid/Settled” state and Merchant initiates refund.
  + It also depends on Merchant’s configurations
  + Weather Full or Partial Refund depends of amount refunded by Merchant and this is initiated from Merchant portal. But in this the final status is “Refunded”

#### Cancel

* **Description:** Merchant can initiate Cancel if transaction(ATRN) status is Paid
* **Transition Conditions:**
  + The transaction can move to "Cancel" status, if transaction is already in “Paid” state and Merchant initiates refund.
  + It also depends on Merchant’s configurations
* **Technical Considerations:**
* Clear error reporting to merchants and users.
* Wrong response received from Merchant that includes field level validation, Encryption

#### Payment Processing

#### **Pay-mode**

The Transaction Payment can be done by different methods through which a customer can make a payment for a transaction on the merchant's platform. Each payment mode represents a different channel or method of payment that the system can process.

While creating order we check the Merchant approved paymode and same is displayed on ePay page. Similarly we check if there is any downtime set for any these paymode, to be displayed on ePay page.

The downtime schedule is managed and configured through the SBIePay Admin interface.

* + 1. **Credit/ Debit Card**

• Payment is made using a credit card issued by a bank or financial institution.

• Requires entry of card details like card number, expiry date, and CVV.

• Payment Gateway need to be used PayU (Wibmo)

* + 1. **Net Banking – SBI / Channel Banks**

**•** Payment is processed through the customer’s online banking portal.

• The customer is redirected to their bank's online banking page to complete the payment.

• For a successful corporate transaction status will be in the Pending state, the checker user has to complete the transaction within 24Hours, otherwise transaction will be marked as expired at SBIePay end. Post which if Checker will authorize the transaction it will treat as expired only.

• SBIePay application will run the scheduler for offline pulling with corporate bank which will fetch the status till 24 hours and update accordingly the transaction if response will received from bank as Success/Failed.

* + 1. **UPI**

• Payment is initiated using a UPI ID or Mobile number or QR code.

• Payment Gateway need to be used as BHIM-UPI

#### **Payment Retry**

* Retry is a new functionality, which will allow customers to retry transactions using other paymodes, if previous transaction is failed and transaction token time is not expired.
* For Each transaction attempt a unique ATRN will be generated.
* Maximum Five retry attempts will be allowed to customer as default.
* Retry attempt will be configurable merchant to merchant.
* Each retry attempt generates a new ATRN if the previous attempt is genuine failure from channel bank / payment channel.
* This new ATRN uniquely identifies each retry attempt, allowing for clear tracking and differentiation from previous attempts.
* No retry functionality will be available for Offline payment modes
* If transaction token is expired and no successful payment attempt is captured, all the ATRN(s) status will be marked as failed, and OrderID status should be changed to ‘expired’ from ‘attempted’ status.
* If the Transaction token is valid and number of payments to be attempted count has been exhausted, then orderID status will be moved from ‘attempted’ to ‘failed’.

#### **Bank double verification**

* Once the web response of payment is successful, a direct server-to-server call happens between SBIePay server and the customer bank to validate the payment status, in a positive scenario the Order status is marked as Paid.
* If SBIePay if not received the response from payment channel, for those transactions SBIePay will run offline pulling (DV) to get the status. Once response received from payment channel it will mark as success or failure for the same.
* If no response received from bank end, it will be in queue to get the status till 5 attempts for Booked to fail (Merchant time interval-Gatewaytimeinterval to EpayTransactionDVExpiryTime) scheduler.
* Once transaction is marked as success/fail from DV and failed from Book to fail scheduler, PUSH response will send to merchant if URL is configured at SBIePay end.

#### Operations

#### Get transaction

**Description**: This operation retrieves the transaction based on its ATRN or other identifiers. The system decrypts the sensitive fields before sending the response to the client.

**Process Flow:-**

**Receive Get transaction Request:** The merchant requests for transaction details using **ATRN** and any other mandatory parameter in encrypted format.

**Fetch transaction from Database:** Retrieve the transaction details by the identifier provided.

**Decrypt Sensitive Data:** Decrypt fields such as ATRN, Period, Customer info, order info, Payment Details using **AES-256 GCM** with the appropriate key.

**Return Response:** Return the full transaction details, including decrypted sensitive fields.

**Technical Considerations**

**Decryption:** Ensure the integrity of decrypted data using **GCM's authentication tag** to prevent tampering.

**Error Handling:** Handle cases where decryption fails (e.g., due to key rotation or corrupted data).

Error Messages should be defined based on type of errors that can occur during data extraction service.

**Token validity:** Check if token is valid to call the service else Merchant will requests for new token

#### Get all transaction for a Merchant

* + **Description:** This operation fetches all transactions associated with a specific merchant as per duration(Date/time). Sensitive data will be encrypted in the database but decrypted before sending the response.
  + **Process Flow**
    - **Receive Get Orders Request:** Merchant sends an encrypted request to fetch all transactions for a particular duration of transaction associated with their account.
    - **Fetch Orders from Database:** Retrieve all transactions associated with the **Merchant ID**.
    - **Decrypt Sensitive Data:** Decrypt sensitive fields like ATRN, Period, Customer info, order info, Payment Details using **AES-256 GCM** with the appropriate key
    - **Return Response:** Return the list of orders, with decrypted sensitive data and any relevant order metadata (e.g., order status, timestamp).
    - By default transactions will be retrieved based on Latest first.
  + **Technical Considerations**
    - **Pagination:** Implement performance pagination, especially when dealing with many transactions.
    - **Decryption:** Ensure proper decryption for each transaction in the list without affecting performance.
    - **Error Handling:** Handle cases where decryption fails (e.g., due to key rotation or corrupted data).
    - Error Messages should be defined based on type of errors that can occur during data extraction service.
    - **Token validity:** Check if token is valid to call the service else Merchant will requests for new token

#### Filter transaction/ Payments

* + **Description -** This operation allows fetching transaction based on various filters like ATRN Number/Order Number/ Customer Id/ Customer Name/ Customer Phone Number/ Customer Email/Tx Status/ Refund/Cancellation Status/ Payout Status. Data will be decrypted before sending the response.
  + **Process Flow**
    - **Receive Filtered Get Order Request:** Merchant provides one or more filters to search orders (e.g., Order Number, Customer ID, Customer Email, etc.).
    - **Fetch Orders from Database:** Query the database using the provided filter criteria.
    - **Decrypt Sensitive Data:** Decrypt fields containing ATRN Number/Order Number/ Customer Id/ Customer Name/ Customer Phone Number/ Customer Email/Tx Status/ Refund/Cancellation Status/ Payout Status for the transaction captured in ePay DB.
    - **Return Response:** Return the list of transactions that match the provided criteria, with decrypted sensitive fields.
      * By default transactions will be retrieved based on Latest first.
  + **Technical Considerations**
    - **Complex Queries:** Support for complex filtering, such as combining multiple criteria (e.g., transaction status and Order status).
    - **Decryption Performance:** Ensure the decryption process scales efficiently, especially when retrieving large data sets.
    - **Error Handling:** Handle cases where decryption fails (e.g., due to key rotation or corrupted data).
      * Error Messages should be defined based on type of errors that can occur during data extraction service.
    - **Token validity:** Check if token is valid to call the service else Merchant will requests for new token