**Akshitha Katta(U38519838)**

**Testing Portfolio**

**Test Plan Identifier**

**Document ID**: PhonePe\_TestPlan\_2024  
**Version**: 1.0  
**Date**: October 2024  
**Prepared by**: Akshitha Katta

**2. Introduction**

The purpose of this test plan is to define the scope, objectives, and approach for testing the **PhonePe** mobile application. The test will focus on verifying key functionalities like login, UPI payments, recharge & bill payments, transaction history, and security features. This test plan will ensure that the application meets the expected standards for performance, usability, and security, and will identify any defects or issues.

**3. Objectives**

The main objectives of this test plan are:

* **Functional Validation**: Ensure the key modules of PhonePe (login, UPI payments, recharge, transaction history) function correctly under various input conditions.
* **Reliability Testing**: Verify that PhonePe consistently performs its expected functions without failure.
* **Error Handling**: Test how well the application handles unexpected input, errors, and edge cases.
* **Security Testing**: Validate the security mechanisms, especially OTP validation and UPI PIN authentication, to prevent unauthorized access and fraud.
* **Automated Testing**: Implement and demonstrate automated testing for key functions to improve efficiency.

**4. Scope of Testing**

The scope of testing will focus on the following modules of PhonePe:

1. **Login and Registration**: Testing the registration and login process with valid and invalid credentials, OTP validation, and session handling.
2. **UPI Payments**: Testing payment flows including successful and failed transactions, UPI PIN validation, and retry mechanisms.
3. **Recharge and Bill Payments**: Ensuring smooth and error-free mobile recharges and bill payments.
4. **Transaction History**: Verifying the accuracy of recorded transactions and their accessibility in the history tab.
5. **Security Features**: Testing for security-related vulnerabilities, OTP verification, and transaction authorization.

**5. Assumptions**

* The PhonePe application is stable and free from any major known defects before testing begins.
* Test data will include valid and invalid UPI IDs, transaction amounts, and user credentials.
* Test environment is correctly set up to mimic real-world usage conditions, including network connectivity fluctuations.

**6. Test Items**

The key modules to be tested are:

* **Login Module**: Tests related to user authentication, OTP handling, and session management.
* **UPI Payment Module**: Tests for payment success, failures, retries, and transaction history updates.
* **Recharge and Bill Payment Module**: Tests for successful recharge and bill payment flows.
* **Transaction History Module**: Tests to verify the correct recording and display of transactions.
* **Security Features**: Tests for validating security aspects such as OTP handling and UPI PIN validation.

**7. Features to be Tested**

The following features will be tested:

1. **User Authentication**:
   1. Login with valid and invalid credentials.
   2. OTP validation.
   3. Session timeout and logout.
2. **UPI Payment Flow**:
   1. Successful UPI payment transaction.
   2. Payment failures (due to insufficient balance, invalid UPI ID).
   3. Payment retries and correct handling of failed transactions.
3. **Recharge & Bill Payment**:
   1. Recharge flow with valid inputs.
   2. Handling failed recharges (e.g., invalid operator).
   3. Bill payments to various service providers.
4. **Transaction History**:
   1. Verifying successful transactions are correctly logged in history.
   2. Checking accuracy of details such as amount, date, and recipient.
5. **Security Testing**:
   1. OTP verification during login.
   2. UPI PIN validation for payments.
   3. Resistance to unauthorized access attempts.

**8. Features Not to Be Tested**

The following features will not be tested as part of this plan:

* **Third-Party Integrations**: Any functionality relying on third-party services or APIs will be excluded unless it is critical for UPI payments.
* **Promotions and Offers**: Promotional deals and cash-back offers are not part of the core testing.
* **Advanced Analytics**: Usage tracking, customer insights, or performance metrics based on user data will not be covered.

**9. Test Strategy**

The testing strategy will cover functional, structural, and automated testing methods.

**1. Functional Testing**  
Functional testing will include five key techniques:

* **Boundary Value Analysis**: Testing minimum and maximum values (e.g., password length, transaction limits).
* **Equivalence Partitioning**: Grouping inputs into valid and invalid sets (e.g., correct/incorrect UPI IDs).
* **Decision Table Testing**: Decision table to handle different outcomes based on valid/invalid combinations of inputs.
* **State Transition Testing**: Testing the transitions between different states (e.g., payment status transitions from "pending" to "success" or "failure").
* **Error Guessing**: Predicting common user errors (e.g., wrong UPI PIN, invalid OTP).

**2. Structural Testing**  
If the source code is accessible, structural testing will include:

* **Code Coverage**: Ensure all key paths, loops, and branches in the login, UPI payment, and transaction history functions are tested.
* **Control Flow Testing**: Validate that the control flow during various stages of UPI payments operates as expected.

**10. Entry and Exit Criteria**

**Entry Criteria**:

* The test environment is fully set up, including test data and user credentials.
* The PhonePe application build is stable.
* All necessary test cases and test scripts are ready for execution.

**Exit Criteria**:

* All planned test cases have been executed.
* All high-severity defects have been resolved or tracked for future releases.
* Test results have been documented, and the overall quality of the application is assessed.

**11. Test Deliverables**

The deliverables for the testing process include:

* **Test Cases Document**: Detailed functional test cases for each module.
* **Defect Reports**: Comprehensive documentation of any bugs or defects encountered during testing.
* **Test Execution Report**: A report summarizing the results of all test cases (pass/fail status).
* **Automated Test Scripts**: Selenium scripts for automated tests.
* **Final Portfolio Report**: A final report summarizing the entire testing process, findings, and conclusions.

**12. Environmental Needs**

* **Test Environment**: A simulated user environment with access to a test version of PhonePe.
* **Devices**: Testing will be conducted on Android and iOS devices to ensure compatibility.
* **Tools**: Testing tools like Selenium for automation and Jira for defect tracking.
* **Data**: Valid and invalid UPI IDs, payment amounts, recharge values, and credentials.

**13. Resources and Responsibilities**

* **Test Lead**: Akshitha Katta
* **Automation Engineer**: Responsible for automating key test cases.
* **Manual Testers**: Responsible for executing functional and structural tests.
* **Test Environment Engineer**: Responsible for setting up and maintaining the test environment.

**15. Risks**

* **Risk**: Delays in environment setup or device issues.
  + **Mitigation**: Ensure early preparation of the test environment and have backup devices.
* **Risk**: High number of defects identified.
  + **Mitigation**: Prioritize critical defects for immediate fixing and log the lower-severity ones for future updates.

**Boundary Value Analysis (BVA):**

**Test Case 1:**

* **Test Case ID** : B001
* **Component**: Login Module
* **Purpose of Test Case**: Verify login with a password length below the minimum limit.
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: Password of 5 characters (min length is 6).
* **Expected Output**: The system should reject the password with an appropriate error message.
* **Post-Conditions**: User remains on the login screen.
* **Execution History**:

**Date**: 09-25-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 2:**

* **Test Case ID**: B002
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with an amount at the maximum allowable limit
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: User is logged in, and UPI payment option is available.
* **Inputs**: Payment amount of 1,00,000 INR (maximum limit).
* **Expected Output**: Payment should be processed successfully.
* **Post-Conditions**: The transaction should reflect in the transaction history.
* **Execution History**:

**Date**: 09-25-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 3:**

* **Test Case ID**: B003
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Verify UPI payment below the minimum transaction amount.
* **Functional Test Type** Boundary Value Analysis
* **Pre-Conditions**: User is logged in, and UPI payment option is available.
* **Inputs**: Payment amount of 0.10 INR (below minimum of 1 INR).
* **Expected Output**: System should reject the payment and display an error message.
* **Post-Conditions**: No payment should be processed.
* **Execution History**:

**Date**: 09-25-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 4:**

* **Test Case ID**: B004
* **Component**: Registration Module
* **Purpose of Test Case**: Test phone number validation with an invalid number of digits .
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions** User is on the registration screen.
* **Inputs**: A phone number with 9 digits.
* **Expected Output**: The system should reject the input and display a warning.
* **Post-Conditions**: User is not registered.
* **Execution History**:

**Date**: 09-25-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 5:**

* **Test Case ID**: B005
* **Component**: OTP Verification
* **Purpose of Test Case**: Validate OTP verification with minimum boundary.
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: OTP screen is displayed.
* **Inputs**: 4-digit OTP (valid length is 6 digits).
* **Expected Output**: The system should reject the OTP as invalid.
* **Post-Conditions**: OTP verification fails.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 6:**

* **Test Case ID**: B006
* **Component**: Password Reset
* **Purpose of Test Case**: Validate password change with a password at maximum length.
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: User is on the password reset screen.
* **Inputs**: 16-character password (maximum limit).
* **Expected Output**: The system accepts the password and updates successfully.
* **Post-Conditions**: Password is changed.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 7:**

* **Test Case ID**: B007
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with an amount slightly above the maximum limit.
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: User is logged in, and UPI payment option is available.
* **Inputs**: Payment amount of 1,00,001 INR.
* **Expected Output**: System rejects the payment with an error message.
* **Post-Conditions**: No transaction is processed.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 8:**

* **Test Case ID**: B008
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Test UPI payment at minimum amount limit.
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: User is logged in.
* **Inputs**: Payment amount of 1 INR.
* **Expected Output**: Payment is processed successfully.
* **Post-Conditions**: Transaction is logged in history.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 9:**

* **Test Case ID**: B009
* **Component**: Login Module
* **Purpose of Test Case**: Test login with the maximum length password.
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: User is on the login screen.
* **Input**: 20-character password.
* **Expected Output**: The system logs in the user successfully.
* **Post-Conditions**: User is logged in.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 10:**

* **Test Case ID**: B010
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with an amount slightly below the minimum allowable limit.
* **Functional Test Type**: Boundary Value Analysis
* **Pre-Conditions**: User is logged in.
* **Inputs**: Payment amount of 0.99 INR.
* **Expected Output**: System should reject the payment and display an error message.
* **Post-Conditions**: No transaction is processed.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Equivalence Partitioning**

**Test Case 1:**

* **Test Case ID**: E001
* **Component**: Login Module
* **Purpose of Test Case**: Validate login with a valid phone number.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: A valid 10-digit phone number.
* **Expected Output**: System accepts the phone number and proceeds to OTP verification.
* **Post-Conditions**: User proceeds to OTP verification.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 2:**

* **Test Case ID**: E002
* **Component**: Login Module
* **Purpose of Test Case**: Validate login with an invalid phone number (too few digits).
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: A 9-digit phone number.
* **Expected Output**: System rejects the phone number and displays an error message.
* **Post-Conditions:**  User remains on the login screen.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 3:**

* **Test Case ID**: E003
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with a valid UPI ID.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is logged in and on the UPI payment screen.
* **Inputs**: Valid UPI ID (e.g., name@bank).
* **Expected Output**: The system processes the payment and shows a success message.
* **Post-Conditions**: Payment is processed, and a receipt is shown.
* **Execution History**:

**Date**: 10-21-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 4:**

* **Test Case ID**: E004
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with an invalid UPI ID.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is logged in and on the UPI payment screen.
* **Inputs**: Invalid UPI ID (e.g., invalid@wrong).
* **Expected Output**: The system rejects the UPI ID and shows an error message.
* **Post-Conditions**: No payment is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 5:**

* **Test Case ID**: E005
* **Component**: OTP Verification
* **Purpose of Test Case**: Validate OTP entry with a valid 6-digit code.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is on the OTP entry screen.
* **Inputs**: Valid 6-digit OTP.
* **Expected Output**: The system accepts the OTP and logs the user in.
* **Post-Conditions**: User is logged in.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 6:**

* **Test Case ID**: E006
* **Component**: OTP Verification
* **Purpose of Test Case**: Validate OTP entry with an invalid OTP.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is on the OTP entry screen.
* **Inputs**: Invalid 6-digit OTP.
* **Expected Output**: The system rejects the OTP and displays an error message.
* **Post-Conditions**: User remains on the OTP entry screen.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 7:**

* **Test Case ID**: E007
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with insufficient balance.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is logged in, UPI is set up, and balance is insufficient.
* **Inputs**: Valid UPI ID, valid PIN, payment amount exceeding balance.
* **Expected Output**: The system rejects the payment and shows an "Insufficient Balance" error.
* **Post-Conditions**: No payment is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 8:**

* **Test Case ID**: E008
* **Component**: Recharge Module
* **Purpose of Test Case**: Validate recharge with valid mobile number and operator.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is logged in and on the recharge screen.
* **Inputs**: Valid mobile number and valid operator.
* **Expected Output**: System completes the recharge successfully.
* **Post-Conditions**: Recharge is processed and confirmed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 9:**

* **Test Case ID**: E009
* **Component**: Recharge Module
* **Purpose of Test Case**: Validate recharge with an invalid mobile number.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is logged in and on the recharge screen.
* **Inputs**: Invalid mobile number.
* **Expected Output**: The system rejects the input and shows an error message.
* **Post-Conditions**: No recharge is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 10:**

* **Test Case ID**: E010
* **Component**: Recharge Module
* **Purpose of Test Case**: Validate recharge with an invalid operator selection.
* **Functional Test Type**: Equivalence Partitioning
* **Pre-Conditions**: User is logged in and on the recharge screen.
* **Inputs**: Valid mobile number, invalid operator.
* **Expected Output**: The system rejects the operator selection and shows an error message.
* **Post-Conditions**: No recharge is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Decision Table Testing :**

**Test Case 1:**

* **Test Case ID**: DT001
* **Component**: Login Module
* **Purpose of Test Case**: Validate login with valid username and valid password.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: Valid username, valid password.
* **Expected Output**: System logs the user in successfully.
* **Post-Conditions**: User is logged in.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 2:**

* **Test Case ID**: DT002
* **Component**: Login Module
* **Purpose of Test Case**: Validate login with valid username and invalid password.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: Valid username, invalid password.
* **Expected Output**: System rejects the login and shows an error message.
* **Post-Conditions**: User remains on the login screen.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 3:**

* **Test Case ID**: DT003
* **Component**: Login Module
* **Purpose of Test Case**: Validate login with invalid username and valid password.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**  User is on the login screen.
* **Inputs**: Invalid username, valid password.
* **Expected Output**: System rejects the login and shows an error message.
* **Post-Conditions**: User remains on the login screen.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 4:**

* **Test Case ID**: DT004
* **Component**: Login Module
* **Purpose of Test Case**: Validate login with invalid username and invalid password.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: Invalid username, invalid password.
* **Expected Output**: System rejects the login and shows an error message.
* **Post-Conditions**: User remains on the login screen.
* **Execution History**:

**Date**: 09-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 5:**

* **Test Case ID**: DT005
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with valid UPI ID and valid UPI PIN.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**  User is logged in and on the UPI payment screen.
* **Inputs**: Valid UPI ID, valid UPI PIN.
* **Expected Output**: System processes the payment successfully.
* **Post-Conditions**: Payment is processed, and transaction receipt is shown.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 6:**

* **Test Case ID**  DT006
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with valid UPI ID and invalid UPI PIN.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is logged in and on the UPI payment screen.
* **Inputs**: Valid UPI ID, invalid UPI PIN.
* **Expected Output**: System rejects the UPI PIN and shows an error message.
* **Post-Conditions**: No payment is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 7:**

* **Test Case ID**: DT007
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with invalid UPI ID and valid UPI PIN.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is logged in and on the UPI payment screen.
* **Inputs**: Invalid UPI ID, valid UPI PIN.
* **Expected Output**: System rejects the UPI ID and shows an error message.
* **Post-Conditions**: No payment is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 8:**

* **Test Case ID**: DT008
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Validate UPI payment with invalid UPI ID and invalid UPI PIN.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is logged in and on the UPI payment screen.
* **Inputs**: Invalid UPI ID, invalid UPI PIN.
* **Expected Output**: System rejects the payment and displays an error message.
* **Post-Conditions**: No payment is processed.
* **Execution History**:

**Date**: 09-26-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 9:**

* **Test Case ID**: DT009
* **Component**: Recharge Module
* **Purpose of Test Case**: Validate recharge with valid mobile number and valid operator.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is on the recharge screen.
* **Inputs**: Valid mobile number and valid operator.
* **Expected Output**: System processes the recharge successfully.
* **Post-Conditions**: Recharge is completed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 10:**

* **Test Case ID**: DT010
* **Component**: Recharge Module
* **Purpose of Test Case**: Validate recharge with valid mobile number and invalid operator.
* **Functional Test Type**: Decision Table Testing
* **Pre-Conditions**: User is on the recharge screen.
* **Inputs**: Valid mobile number and invalid operator.
* **Expected Output**  System rejects the operator and shows an error message.
* **Post-Conditions**: No recharge is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**State Transition Testing :**

**Test Case 1:**

* **Test Case ID**: ST001
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Test state transition from "Pending" to "Success" for a successful UPI payment.
* **Functional Test Type**: State Transition Testing
* **Pre-Conditions**: User is on the UPI payment screen with valid payment details.
* **Inputs**: Valid UPI ID, valid UPI PIN, and sufficient balance.
* **Expected Output**: System transitions the transaction from "Pending" to "Success."
* **Post-Conditions**: Payment is marked as successful in transaction history.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 2:**

* **Test Case ID**: ST002
* **Component**: UPI Payment Module
* **Purpose of Test Case**: Test state transition from "Pending" to "Failure" for an unsuccessful UPI payment.
* **Functional Test Type**: State Transition Testing
* **Pre-Conditions**: User is on the UPI payment screen with invalid details or insufficient balance.
* **Inputs**: Invalid UPI PIN or insufficient balance.
* **Expected Output**: System transitions the transaction from "Pending" to "Failure."
* **Post-Conditions**: Payment is marked as failed in transaction history.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 3:**

* **Test Case ID**: ST003
* **Component**: Login Module
* **Purpose of Test Case**: Test state transition from "Logged Out" to "Logged In" after entering valid credentials.
* **Functional Test Type**: State Transition Testing
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: Valid phone number, valid OTP.
* **Expected Output**: System transitions the user from "Logged Out" to "Logged In."
* **Post-Conditions**: User is logged into the app.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 4:**

* **Test Case ID**: ST004
* **Component**: Login Module
* **Purpose of Test Case**: Test state transition from "Logged Out" to "Verification Failed" after entering invalid OTP.
* **Functional Test Type**: State Transition Testing
* **Pre-Conditions**: User is on the OTP entry screen.
* **Inputs**: Invalid OTP.
* **Expected Output**: System transitions to "Verification Failed."
* **Post-Conditions**: User remains logged out, and an error message is displayed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 5:**

* **Test Case ID**: ST005
* **Component**: Transaction History Module
* **Purpose of Test Case**: Test state transition from "Pending" to "Logged" after a successful payment.
* **Functional Test Type**: State Transition Testing
* **Pre-Conditions**: User completes a successful UPI payment.
* **Inputs**: N/A (System event).
* **Expected Output**: The transaction status changes to "Logged" in transaction history.
* **Post-Conditions**: The transaction appears in the history.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Error Guessing :**

**Test Case 1:**

* **Test Case ID**: EG001
* **Component**: Login Module
* **Purpose of Test Case**: Test system behavior when entering an invalid phone number (letters instead of digits).
* **Functional Test Type**: Error Guessing
* **Pre-Conditions**: User is on the login screen.
* **Inputs**: Phone number with letters (e.g., abcdefghij).
* **Expected Output**: System rejects the input and displays an error message.
* **Post-Conditions**: User remains on the login screen.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 2:**

* **Test Case ID** EG002
* **Component**  UPI Payment Module
* **Purpose of Test Case**: Test system behavior when making a UPI payment with an incorrect UPI ID.
* **Functional Test Type**: Error Guessing
* **Pre-Conditions**: User is logged in and on the UPI payment screen.
* **Inputs**: UPI ID: wrongupi@bank.
* **Expected Output**: System rejects the UPI ID and shows an error.
* **Post-Conditions**: No payment is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 3:**

* **Test Case ID**: EG003
* **Component**: Recharge Module
* **Purpose of Test Case**: Test system behavior when entering an invalid recharge amount (negative value).
* **Functional Test Type**: Error Guessing
* **Pre-Conditions**: User is on the recharge screen.
* **Inputs**: Recharge amount: -100 INR.
* **Expected Output**: System rejects the input and shows an error.
* **Post-Conditions**  No recharge is processed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 4:**

* **Test Case ID**: EG004
* **Component**: Transaction History Module
* **Purpose of Test Case**: Test system behavior when viewing transaction history during network failure.
* **Functional Test Type**: Error Guessing
* **Pre-Conditions**: User tries to access transaction history with no internet connection.
* **Inputs**: N/A (network failure).
* **Expected Output**: System displays an error about the network issue.
* **Post-Conditions**: Transaction history is not displayed.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 5:**

* **Test Case ID**: EG005
* **Component**: OTP Verification
* **Purpose of Test Case**: Test system behavior when entering a valid OTP after it has expired.
* **Functional Test Type**: Error Guessing
* **Pre-Conditions**: User is on the OTP entry screen.
* **Input** Expired OTP.
* **Expected Output**: System rejects the OTP and displays a message about expiry.
* **Post-Conditions**: User is not logged in.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Structural Testing Test Case 1: Code Coverage for UPI Payment Processing**

**Test Case ID: STC001:**

* **Component**: UPI Payment Module
* **Purpose of Test Case**: Ensure that all branches and conditions in the UPI payment flow are executed and tested.
* **Functional Test Type**: Code Coverage Testing
* **Pre-Conditions**: User is logged into the PhonePe app and has sufficient balance.
* **Inputs**:

**Scenario 1**: Valid UPI ID and valid UPI PIN.

**Scenario 2**: Valid UPI ID and invalid UPI PIN.

**Scenario 3**: Invalid UPI ID and valid UPI PIN.

* **Expected Output**:

**Scenario 1**: Payment should be processed successfully.

**Scenario 2**: The system should reject the payment due to an invalid UPI PIN.

**Scenario 3**: The system should reject the payment due to an invalid UPI ID.

* **Post-Conditions**:

**Scenario 1**: Payment is successful and logged in transaction history.

**Scenario 2**: No payment is processed.

* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 2: Control Flow Testing for Transaction Logging**

**Test Case ID: STC002**

* **Component**: Transaction History Module
* **Purpose of Test Case**: Validate control flow to ensure that all transactions are correctly logged when payments succeed or fail.
* **Functional Test Type**: Control Flow Testing
* **Pre-Conditions**: User has made a UPI payment attempt (successful or failed).
* **Inputs**:

**Scenario 1**: Successful payment with valid UPI ID and UPI PIN.

**Scenario 2**: Failed payment due to an invalid UPI PIN.

**Scenario 3**: Failed payment due to an invalid UPI ID.

* **Expected Output**:

**Scenario 1**: Transaction is successfully logged in the history with a "Success" status.

**Scenario 2**: Transaction is logged with a "Failure" status and proper error details.

**Scenario 3**: Transaction is logged with a "Failure" status and proper error details.

* **Post-Conditions**: All transactions (successful or failed) should be visible in the transaction history with accurate status.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Test Case 3: Data Flow Testing for Payment Retry**

**Test Case ID: STC003**

* **Component**: UPI Payment Retry Module
* **Purpose of Test Case**: Ensure that data such as UPI IDs and transaction amounts are correctly preserved during retry attempts for a failed payment.
* **Functional Test Type**: Data Flow Testing
* **Pre-Conditions**: User has initiated a UPI payment, and the payment failed (e.g., due to network issues or invalid UPI PIN).
* **Inputs**:

**Scenario 1**: Valid UPI ID and correct payment amount on retry after initial network failure.

**Scenario 2**: Valid UPI ID and correct payment amount on retry after initial failure due to invalid UPI PIN.

* **Expected Output**:

**Scenario 1**: The system retains the UPI ID and transaction amount and processes the retry successfully after network issues are resolved.

**Scenario 2**: The system retains the UPI ID and transaction amount and processes the retry successfully after the correct UPI PIN is entered.

* **Post-Conditions**: The original transaction data (UPI ID and amount) is retained, and successful retries are logged in transaction history.
* **Execution History**:

**Date**: 10-22-2024

**Result**: Pass

**Tester**: Akshitha Katta

**Automated Testing:**

**Test Case 1: Validate Login with Invalid Phone Number**

* **Test Case ID**: AT001
* **Test Description**: Verify login fails with an invalid phone number format.
* **Pre-conditions**: The user is on the login page.
* **Steps**:
  1. Enter a phone number with letters (e.g., abcdefghij).
  2. Click the "Login" button.
* **Expected Result**: The system displays an error message about the invalid phone number format.

**Test Case 2: Check Logout Functionality**

* **Test Case ID**: AT002
* **Test Description**: Validate that the user can successfully log out of the application.
* **Pre-conditions**: The user is logged in.
* **Steps**:
  1. Click on the user profile icon.
  2. Click on "Logout."
* **Expected Result**: The user is logged out and redirected to the login page.

**Test Case 3: Validate UPI Payment Retry after Network Failure**

* **Test Case ID**: AT003
* **Test Description**: Verify UPI payment retry functionality when there is a network issue.
* **Pre-conditions**: The user is logged in, and the first payment attempt failed due to network issues.
* **Steps**:
  1. Initiate a UPI payment.
  2. Simulate a network failure during the transaction.
  3. Retry the payment.
* **Expected Result**: The system successfully processes the payment on retry.

**Test Case 4: Validate Bill Payment Functionality with Invalid Details**

* **Test Case ID**: AT004
* **Test Description**: Ensure the system rejects bill payments with invalid details.
* **Pre-conditions**: The user is logged in.
* **Steps**:
  1. Go to the "Bill Payments" section.
  2. Enter an invalid consumer number for the electricity bill.
  3. Click on "Pay."
* **Expected Result**: The system rejects the payment and displays an error message.

**Test Case 5: Verify Search Functionality in Transaction History**

* **Test Case ID**: AT005
* **Test Description**: Validate that the user can search for transactions in their history using specific keywords.
* **Pre-conditions**: The user is logged in with existing transaction history.
* **Steps**:
  1. Go to "Transaction History."
  2. Enter a keyword or recipient name in the search bar.
  3. Click on "Search."
* **Expected Result**: The system filters and displays relevant transactions.

**Test Case 6: Check OTP Expiration Handling**

* **Test Case ID**: AT006
* **Test Description**: Ensure that expired OTPs are rejected.
* **Pre-conditions**: The user is on the OTP entry screen.
* **Steps**:
  1. Wait for the OTP to expire.
  2. Enter the expired OTP.
  3. Click "Verify."
* **Expected Result**: The system rejects the OTP and displays an expiration message.

**Test Case 7: Test Password Reset with Valid Details**

* **Test Case ID**: AT007
* **Test Description**: Verify that users can successfully reset their password with valid details.
* **Pre-conditions**: The user is on the password reset page.
* **Steps**:
  1. Enter a valid phone number.
  2. Enter the OTP received.
  3. Enter a new valid password.
  4. Confirm the password.
  5. Click "Submit."
* **Expected Result**: The system resets the password and shows a success message.

**Test Case 8: Adding a New Bank Account**

* **Test Case ID**: AT008
* **Test Description**: Validate that users can successfully add a new bank account.
* **Pre-conditions**: The user is logged in.
* **Steps**:
  1. Navigate to the "Bank Accounts" section.
  2. Click "Add Bank Account."
  3. Enter valid bank details.
  4. Click "Submit."
* **Expected Result**: The bank account is successfully added and linked.

**Test Case 9: Test Transaction History Export**

* **Test Case ID**: AT009
* **Test Description**: Verify that the user can export their transaction history as a PDF or Excel file.
* **Pre-conditions**: The user is logged in with transaction history available.
* **Steps**:
  1. Go to the "Transaction History" section.
  2. Click "Export" and select the format (PDF/Excel).
* **Expected Result**: The system generates the file, and the user is able to download it.

**Test Case 10: Validate Referral Code Usage**

* **Test Case ID**: AT010
* **Test Description**: Check if the referral code is applied correctly during registration.
* **Pre-conditions**: The user is on the registration page with a valid referral code.
* **Steps**:
  1. Enter the referral code during registration.
  2. Complete the registration process.
* **Expected Result**: The referral code is applied, and the user receives the corresponding rewards.

**Defect Reporting Forms:**

**Defect Report 1 :**

* **Defect ID**: DEF006
* **Defect Title**: Search in Transaction History Shows Incorrect Results
* **Module**: Transaction History
* **Priority**: Medium
* **Severity**: Moderate
* **Environment**: Chrome on Windows 10, PhonePe Web version 5.1
* **Reported By**: Akshitha Katta
* **Date**: 10-23-2024
* **Steps to Reproduce**:
  1. Log in and go to the transaction history page.
  2. Search for a transaction using a valid recipient name.
  3. Observe the search results.
* **Expected Result**: The system should show all relevant transactions matching the search criteria.
* **Actual Result**: The system shows incorrect or incomplete transactions.
* **Status**: Open
* **Assigned To**: Development Team

**Defect Report 2 :**

* **Defect ID**: DEF007
* **Defect Title**: Incorrect Payment Failure Message Displayed for Insufficient Balance
* **Module**: UPI Payment
* **Priority**: High
* **Severity**: Major
* **Environment**: Safari on iOS 14, PhonePe version 5.1.0
* **Reported By**: Akshitha Katta
* **Date**: 10-23-2024
* **Steps to Reproduce**:
  1. Open PhonePe and log in.
  2. Initiate a UPI payment with an amount greater than the account balance.
  3. Submit the payment.
* **Expected Result**: The system should display an error message about insufficient balance.
* **Actual Result**: A generic "Payment Failed" message is displayed without mentioning the reason (insufficient balance).
* **Status**: Open
* **Assigned To**: Development Team

**Defect Report 3:**

**Defect ID**: DEF001

**Defect Title**: Invalid OTPs are not correctly rejected.

**Module**: OTP Verification

**Priority**: High

**Severity**: Major

**Environment**: Android 12, PhonePe version 5.1.0

**Reported By**: Akshitha Katta

**Date**: 10-23-2024

**Steps to Reproduce**:

* 1. Launch the PhonePe application.
  2. Enter a valid phone number for login.
  3. Enter an invalid 6-digit OTP on the OTP verification screen.
  4. Click the "Verify" button.

**Expected Result**: The system should reject the invalid OTP and display an appropriate error message.

**Actual Result**: The system accepts the invalid OTP and proceeds to the home screen.

**Status**: Open

**Assigned To**: Development Team

**Defect Report Form 4:**

**Defect ID**: DEF002

**Defect Title**: UPI payment failure is not properly logged in transaction history.

**Module**: UPI Payment

**Priority**: Medium

**Severity**: Moderate

**Environment**: iOS 14, PhonePe version 5.1.0

**Reported By**: Akshitha Katta

**Date**: 10-23-2024

**Steps to Reproduce**:

* 1. Open the PhonePe app and navigate to the UPI payment section.
  2. Enter a valid UPI ID and invalid UPI PIN.
  3. Attempt the payment.
  4. Go to the transaction history screen.

**Expected Result**: The failed UPI payment should be logged in the transaction history with a "Failed" status and error details.

**Actual Result**: The failed payment is not recorded in the transaction history.

**Status**: Open

**Assigned To**: Development Team

**Defect Report 5:**

**Defect ID**: DEF003

**Defect Title**: Payment Failure Message Shown Without Proper Error Details

**Module**: UPI Payment Module

**Priority**: High

**Severity**: Major

**Environment**: Android 12, PhonePe version 5.1.0

**Reported By**: Akshitha Katta

**Date**: 10-23-2024

**Steps to Reproduce**:

* 1. Initiate a UPI payment to a recipient (e.g., ₹65,000 to MR VISHAL VIJAY).
  2. Attempt the transaction with valid UPI ID and UPI PIN.
  3. Payment fails with a generic error message shown, "Payment Failed."

**Expected Result**:

* 1. The system should provide a more detailed error message specifying the exact reason for the payment failure (e.g., network issue, insufficient balance, invalid UPI PIN).
  2. Additionally, users should be informed of any next steps or alternative solutions.

**Actual Result**:

* 1. A generic "Payment Failed" message is shown without specifying the reason for the failure.
  2. No further action guidance is provided other than retrying.

**Status**: Open

**Assigned To**: Development Team

**Attachments**:

A black screen with a red circle

Description automatically generated