Table of Contents

1. What is E-wallet?

A digital wallet, also known as a digital wallet, can go a long way towards facilitating frictionless purchases. It is a combination of software and data that allows consumers to make quick and easy purchases using something called near-field communications technology.

2. Test Case For E-Wallet

The user can add a max 10k INR amount to the wallet.
The user can add money to wallet using Internet banking / Credit card / Debit Card
The user is able to pay the bill to merchant account using the wallet
Only registered users can avail the digital wallet facility. The guest user cannot use it.
E-wallet offers should work e.g. pay using the wallet and get 100 rs cashback.
E-wallet money can be transferred to the bank with a 4% deduction means transferring 100 Rs from wallet to a bank account will transfer 96 Rs to a linked bank account.
The user should be able to pay all bills and e-commerce orders available on site using wallet money.
There should be no expiry date for wallet money, unlike promotional offers that have valid expiry.
In case of returns, digital wallet money should be refunded back to the E-wallet account within 24 hours.
Email should be triggered to the user when money is debited from digital wallet.
Email should be triggered to the user when money is credited/added to E-wallet.
Email should be triggered to the user in case of errors like transaction failure when the amount is paid using digital wallet.
E-wallet money should be automatically deducted from the final amount to be paid by the user for any kind of payments offered by the site.
Try changing the payment gateway while the request is in progress.
What happens when we give request with the connection is on and we close the connection immediately.

3. Test Case For Credit Card

Check the valid card numberCheck whether the Credit Card is valid or not.
Check for expiry month and expiry year
Check for 3 digit CVC pin is valid or not
Check with an invalid card number
Check with invalid CVC number
Check for the valid user(number).
Check for his/her last login (whether that falls within the quota, the date after his bill was sent)
Check for the balance amount that he can access from the allowed one(say Rs.10,000 or 50,000 and the amount that he/she is in need of the current transaction.
Check for the Account Holders Name on the Credit Card
Check for the expiration date on the Credit card.
Check with past expiration month and year
Check card field accepts characters
The amount will be detected only when all the details are correct
If any card details fail, it should throw an error message
The transaction will be declined when a card does not have enough balance in their account
Checking with a valid card and valid pin.
Checking with a valid card and invalid pin
Checking with invalid card and pin
Check how much is the balance on the Credit Card
Check Credit Card Verification number on the card
Check for the Credit Card Payment Due Date

4. Positive test scenarios for banking applications

1) Verify if the customer details are available in the application before creating an account.

2) Verify all mandatory fields like customer ID, currency, product code, etc. are entered to open an account

3) When opening savings or current banking account verify the product code is selected appropriately.

For example, when you are opening a current account select product code appropriate to current account and when you are opening a saving bank account select a product code appropriate to the savings account.

4) When opening an account verify the currency is selected appropriately i.e., either local or foreign currency.

For example, if you are opening a local savings account then select currency as INR and if you are opening a foreign savings account then select a foreign currency.

5) While transferring amount from one account to another account verify all the mandatory fields are entered, such as debit amount, debit account number, credit amount, credit account number, etc.

6) For telegraphic transfer verify debit currency and credit currency is different.

7) For teller transactions like cash deposit or withdrawal verify credit amount or debit amount is entered correctly. Also, verify all the denominations entered are matched with total debit or credit amount.

8) Verify check details are captured in the retail banking application by entering check number, amount, customer, etc.

9) Verify inward clearing is done by providing details like debit amount, the debit account and credit account.

10) Verify outward clearing is done by providing details like credit amount, credit account and debit account.

11) Verify all the card details like card number, valid through, customer, etc. are encrypted and stored in the system as it is very sensitive data.

12) Verify if a payee can be added through channel banking by providing all the mandatory details like account name, Account number, etc.,

13) Verify if the account transfer is done successfully through channel banking by providing all the mandatory details like transaction type, amount and from an account for an existing payee.

14) Verify whether messages are received once a transaction is done through channel banking.

15) Verify collateral details are given while creating a secured loan.

16) Verify collateral details are not given while creating an unsecured loan.

5. Negative test scenarios:

1) Verify creation of an account with invalid customer details.

2) Verify creation of an account by not selecting a product code or any of the mandatory fields.

3) Verify creating a savings banking account by entering product code as current account and vice versa.

4) Verify creating a local currency account by entering foreign currency amount and vice versa.

5) Verify telegraphic transfer by giving debit currency and credit currency as same. It will be a normal account transfer and not a telegraphic transfer.

6) Verify teller transaction cash withdrawal by entering credit amount and vice versa.

7) Verify teller transaction by providing the invalid denominations.

For example, if credit or debit amount is 150 then provide denomination for 100 as 1 and denomination for 50 as 2. The system should not allow proceeding with the transaction.

8) Verify if a payee can be added through channel banking by providing an account name which is not matched to the account number.

For example, if the account name is Sita and account number is 12345, then for testing this scenario provide Rama as account name and account number as 12345. System should not allow adding a new payee as account name and number are not matching

9) Verify if a payee can be added through channel banking by providing invalid IFSC code.

10) Verify if a payee can be added through channel banking by not providing any of the mandatory fields.

11) Verify if an account transfer through channel banking is done by not providing any of the mandatory fields.

12) Verify if a message is received when a wrong mobile number is provided.

13) Provide collateral details while creating an unsecured loan.

For example, when you’re creating an educational loan, provide mortgage details as collateral. The system should not accept to create an educational loan with mortgage details.

14) While creating a secured loan do not provide collateral details.

6. type of test

Functional testing
Check if new accounts are created correctly with valid data;
Use functional testing type to check how an application behaves if the accounts are created with invalid data;
To make sure your account is secure, check login functionality with invalid data;
It is essential to check that the app functions from a user’s point of view (whether balance updates after withdrawal or crediting, regular payments saved and performed at a specific time);
You must test the app in all of its functions from an admin’s point of view too(whether bulk messaging can be sent and analyzed, whether support requests are correctly handled. Use integration testing for that).

7. Database testing

Check if the data is structured correctly;
Make sure that the field’s data has the correct format;
Check if the values of computed fields are accurately calculated;
Check if each table has all needed constraints: primary keys, foreign keys, and unique indexes;
Check if there is duplicate data in the table;
Don’t be fooled by the null value. Make sure you check if it should exist and not in any other places;
Ensure that all data is written correctly to the table when a user creates or updates their profile;
Check application behaviour when the database server is not working;
Ensure that your data isn’t lost after an operation fails; check if the previous history was saved;
One of the most crucial aspects of maintaining a database is ensuring that you have regular backups. If this process isn’t done on time or at all, your data could be lost forever.

8. Performance testing

Check app performance when lots of users use the same or different functionalities;
Run a test when your device is fully charged, medium charged and when it is about to die;
Use a few types of devices manufactured in different years, for example, 2021 and 2018. You will need this to ensure that the system capabilities are enough not to cause the app’s crash for different devices;
Check app performance when the connection is slow;
You can check your app’s performance to see if it slows down or chokes during a transaction. You should also monitor for this when the Internet speed changes from slow, middling speeds all of a sudden becoming fast-loading and reliable.

9. Accessibility testing

Perceivable Information and user interface
Text alternatives for non-text content
Captions and other alternatives for multimedia
Content can be presented in different ways
Content is easier to see and hear
Operable user interface and navigation
Functionality is available from a keyboard
Users have enough time to read and use the content
Content does not cause seizures and physical reactions
Users can easily navigate, find content, and determine where they are
Users can use different input modalities beyond the keyboard
Understandable information and user interface
Text is readable and understandable
Content appears and operates in predictable ways
Users are helped to avoid and correct mistakes
Robust content and reliable interpretation
Content is compatible with current and future user tools

10. Security testing

Check how the application responds to multiple logins;
Check if the “Forgot password” option allows you to recover the account details quickly;
Check if pass requirements are good enough;
Make sure that IDs and passwords are encrypted;
Check if the application uses a secure protocol, like HTTPS;
Check if the pass hides under dots or other signs when the user enters it in the application;
It is a valuable safety feature to ensure that the user does not remain logged in indefinitely;
Ensure the application response to cache clearance in such types of test cases

11. User acceptance testing

Check if navigation is intuitive and adjustable at the user interface;
While writing a user acceptance testing test case, you should check the font and colour scheme to make sure that all visual elements match;
It is crucial to make sure all the pages on the app are using consistent language, so start with checking for terms like “recurring charges” or regular payments;
Make sure that all of the links and buttons have clear titles;
Error and warning messages should be self-explanatory, so check that they are.
Essential fields have placeholders for important information like tips or explanations on avoiding trouble in the future, so adjust the user interface for that too.