

INCREMENTAL PROJECT

Project Title: Payment Wallet Application

The payment wallet application consists of four functionalities, creating a wallet account, adding amount to the wallet account, show account balance and transfer funds. All the mentioned functionalities are to be developed as dedicated user stories, which would be part of a particular sprint. The development would be in an incremental manner i.e. with every sprint there would be new additions to the previously accomplished status of the project.

Sprint 1: In this sprint, the user data, account details and other related data must be stored / handled using Collections.

User Story:

US1901 – Create payment wallet account

Description:-

As a prospective payment wallet user, I want to register and create an account on the payment wallet application, so that I can use the features of the wallet to perform my financial / business transactions.

Acceptance Criteria:-

1. Given the user signup / register / create account page, the user is able to enter the necessary data and create an account on the payment wallet application.
 2. Given the user details, the data entered should be validated before registering the user and creating an account for the same.
 3. Given the registered credentials, the user should be able to access the payment wallet.
 4. Given any error or exception causing condition, appropriate validation and exception handling should be available to ensure graceful handling of such scenarios.
 5. Given the developed code, respective unit test cases should be accompanied.
-

User Story:

US1902 – Adding amount to the payment wallet account

Description:-

As a payment wallet account holder, I want to add money to my wallet account, so that I can utilize the funds to perform desired financial / business transactions.

Acceptance Criteria:-

1. Given the account holding user, the user is able to add desired amount to his/her respective account.
 2. Given the addition of amount is successful, the new account balance should be available to the user for utilization at his/her discretion.
 3. Given amount addition request, the amount and account details are validated against user details.
 4. Given any error or exception causing condition, appropriate validation and exception handling should be available to ensure graceful handling of such scenarios.
 5. Given the developed code, respective unit test cases should be accompanied.
-

User Story:

US1903 – Show payment wallet account balance

Description:-

As a payment wallet account holder, I want to be able to view / see / check current available balance in my payment wallet account.

Acceptance Criteria:-

1. Given the account holding user, the user is able to see the available balance in his/her own account.
 2. Given any error or exception causing condition, appropriate validation and exception handling should be available to ensure graceful handling of such scenarios.
 3. Given the developed code, respective unit test cases should be accompanied.
-

User Story:

US1904 – Transfer funds from one account to another

Description:-

As a payment wallet account holder, I want to be able to transfer funds from my account to another payment wallet account on the application as a financial / business transaction or payment.

Acceptance Criteria:-

1. Given the account holding user, the user is able to transfer some or all of his/her funds from their own account to any other account registered on the payment wallet application.
 2. Given the account holding user, the transfer amount, existing balance and the account to which the transfer is made should be subjected to appropriate validations.
 3. Given any error or exception causing condition, appropriate validation and exception handling should be available to ensure graceful handling of such scenarios.
 4. Given the developed code, respective unit test cases should be accompanied.
-

Sprint Goals:

1. **Learn, Understand, Assimilate and Implement Core Java concepts.**
2. **Design, develop and deliver the functionalities defined by the user stories.**
3. **Well versed with handling data using Collections.**