Week 2: Spring and JPA

Prerequisite: Collection, Spring, JPA, Exception handling

|  |  |
| --- | --- |
| **Complexity** | Medium |
| **Technology** | Java |
| **Framework** | Spring |
| **Database** | MYSQL|Oracle |
| **Repository support** | JPA |

Use case:

Create a payment wallet application The payment wallet application consists of below functionalities,

* creating a wallet account
* adding amount to the wallet account
* show account balance
* transfer fund

Consider the below entities for the use case. Add additional attributes or entities if required.

|  |  |
| --- | --- |
| User | Datatype |
| id | int(pk) |
| username | string |
| password | string |
| email id | string |

|  |  |
| --- | --- |
| Transaction | Datatype |
| id | int(pk) |
| wallet\_id | int(fk) |
| Status | string |
| Amount | double |
| Date | LocalDate |
| Time | LocalTime |

|  |  |
| --- | --- |
| Wallet | Datatype |
| id | int(pk) |
| user\_id | int(fk) |
| currency\_id | int(fk) |
| balance | double |

|  |  |
| --- | --- |
| Currency | Datatype |
| id | int(pk) |
| name | string |
| abbreviation | string |

* Task 1: Create a maven project with the required dependency for Spring and JPA
* Task 2: Create a layered architecture. Create package with respect to different layers like, dao, service etc.
* Task 3: Create entity classes use the JPA annotation and establish relationship among the entities.
* Task 4: User story 1:

**Create payment wallet account:**

1.Given the user details, the data entered should be validated before registering the user and creating an account for the same.

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.

Sample input 1

Enter user details:

Username: John

Password: \*\*\*\*\*\*\*

Email id: [John@gmail.com](mailto:John@gmail.com)

Output:

User with the given email id already exits

Sample input 2

Enter user details:

Username: John

Password: \*\*\*\*\*\*\*

Email id: JohnRobert@gmail.com

Output:

User registered successfully.

* Task5: User story 2:

**Adding amount to the payment wallet account**

**Note: Post creation of user ideally the wallet should have zero balance.**

1. Given the account holding user, the user is able to add desired amount to his/her respective wallet

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.

Sample input 1

Enter user id/ email id : [JohnRobert@gmail.com](mailto:JohnRobert@gmail.com)

Output

There is no user registered with above email id

Sample input 2

Enter user id/ email id : [JohnRobert@gmail.com](mailto:JohnRobert@gmail.com)

Enter the amount you want to add to the wallet: 1000

Output

Amount added successfully

* Task 6: User story 3:

**Show payment wallet account balance**

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

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

Sample input1

Enter the user id/ email id to view balance: [JohnRobert@gmail.com](mailto:JohnRobert@gmail.com)

Output:

Available balance: Rs. 1000

* Task 7: User story 4:

**Transfer funds from one account to another**

As a payment wallet account holder, I want to be able to transfer funds from my account to another payment wallet account.

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

Sample input1:

Enter the wallet id from which transfer needs to be made: 11778

Enter the wallet id to which transfer needs to be made: 9899

Enter amount to transfer: 2000

Output:

Insufficient balance. You have Rs 1000 balance in your wallet

Sample input2:

Enter the wallet id from which transfer needs to be made: 11778

Enter the wallet id to which transfer needs to be made: 9899

Enter amount to transfer: 200

Output:

Transfer successful. Available balance is Rs.800

Task estimation

|  |  |
| --- | --- |
| Task | Day |
| Task 1, Task2,Task 3 | Day 1(week2) |
| Task 4 | Day 2(week2) |
| Task 5 | Day 3(week2) |
| Task 6 | Day 4(week2) |
| Task 7 | Day 5(week2) |