Programming Assignment : Transfer Service

As per the requirement, I have developed a Transfer Service Rest Api to simulate one of the core features of e-payment system – transferring money from one account to another account.

 # Technology Stack :

1) Spring boot 1.5.10 RELEASE
2) Java 1.8
3) Spring JPA
4) Derby (embedded database)

 # API endpoints : TransferController.java

1) POST - /transfer-api/createAccount – To create monetary accounts
2) PUT - /transfer-api/transfer – To transfer money from one account to another account.
3) GET -/transfer-api/getAllAccount – To get list of all the accounts from database
4) GET -/transfer-api/getAllTransactions – To get list of all the transactions from database

1) **/transfer-api/createAccount :**
   It allows user to create a new account with monetary balance. It accepts a JSON object in the below format and performs validation, e.g. If an account already exists, It returns a message to the user that account already exists, otherwise it will create a new account.
   #Input

```
{
        "accountNumber" : "443232366",
        "balance" : 800,
        "accountHolderName" : "Bhadauria"
}
```

2) **/transfer-api/transfer :**
   It allows user to transfer money from one account to another account. It also accept a JSON Object in the below format and performs various validation prior to execute the transaction. It also throws some custom exceptions as per the business rules.

   - **Insufficient Funds Exception** : when user tries to transfer an amount greater than available amount in the account.
   - **Negative Balance :** When a user tries to transfer a negative amount.

```
{
        "senderAccountNumber" : "443232366",
        "receiverAccountNumber" : "557684573",
        "amount" : 500
}
```

 # Deployment Instructions :

This application made use of Spring boot features and therefore there are many easy ways you can run this application. It needs Maven 3.3.9 or more and Java 1.8 to be able to run successfully.

1) **Using Eclipse :**
   - Clone the application in the local environment from github.com
   - Import the project into workspace.
   - find for "**TransferServiceBootApplication**.java".
   - Right click on TransferServiceBootApplication.java and run as Java Application.

   Spring boot will start the embedded tomcat server and application should be accessible at default port 8080.

   Try url : http://localhost:8080/transfer-api/getAllAccounts

2) **Using Command Line Interface :**
   - Clone the application in the local environment from github.com.
   - Open Command prompt.
   - Navigate to the root folder of the project "**TransferService**", where pom.xml is available.
   - Run command : `mvn package spring-boot:repackage`
   - Run command `: cd target`
   - Run command `: java –jar TransferService-0.0.1-SNAPSHOT.jar`

   Spring boot will start the embedded tomcat server and application should be accessible at default port 8080.

   Try url : http://localhost:8080/transfer-api/getAllAccounts

Note : This application by default creates some dummy accounts and transaction to play at the time of bootstrap.