Credit Card Payment Application

Team Members

1. Amerendra Kumar
2. Ashutosh
3. Bharti Chahal
4. Gaurav Kumar Singh
5. Hemant Jaiswal
6. Isha Swaroop
7. kunal upadhyay

Problem Statement

1. To Create an Application for Credit card bill payment
2. Customer should be able to view credit card payment due date and amount.
3. Customer should be able to pay the bill due amount as well as different amounts as per requirement.
4. Customer should be able to view statement history.
5. Customer should be able to view payment history.

Tools and Technology Used

Tools Used

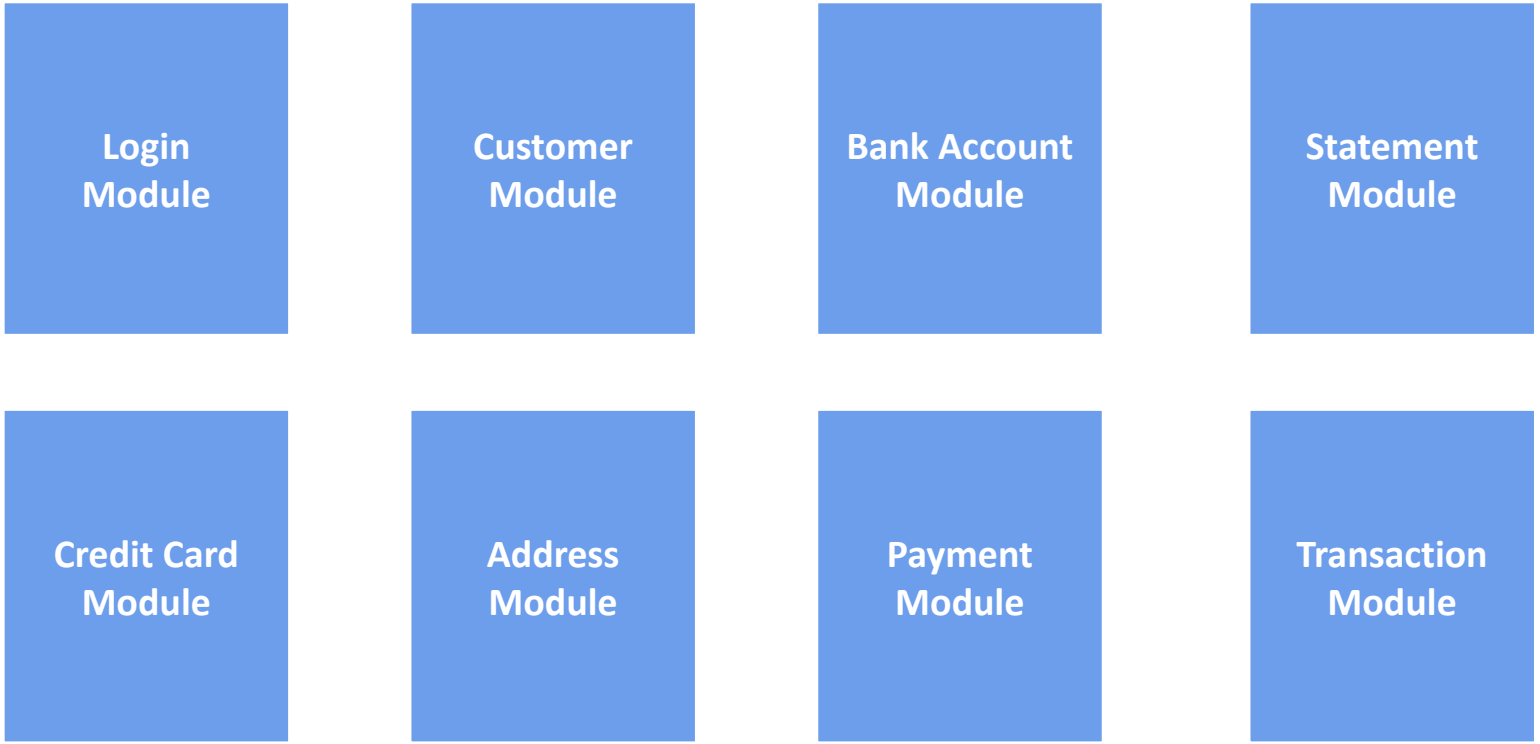
STS **Postman** **Swagger** **Mysql**

Technology Used

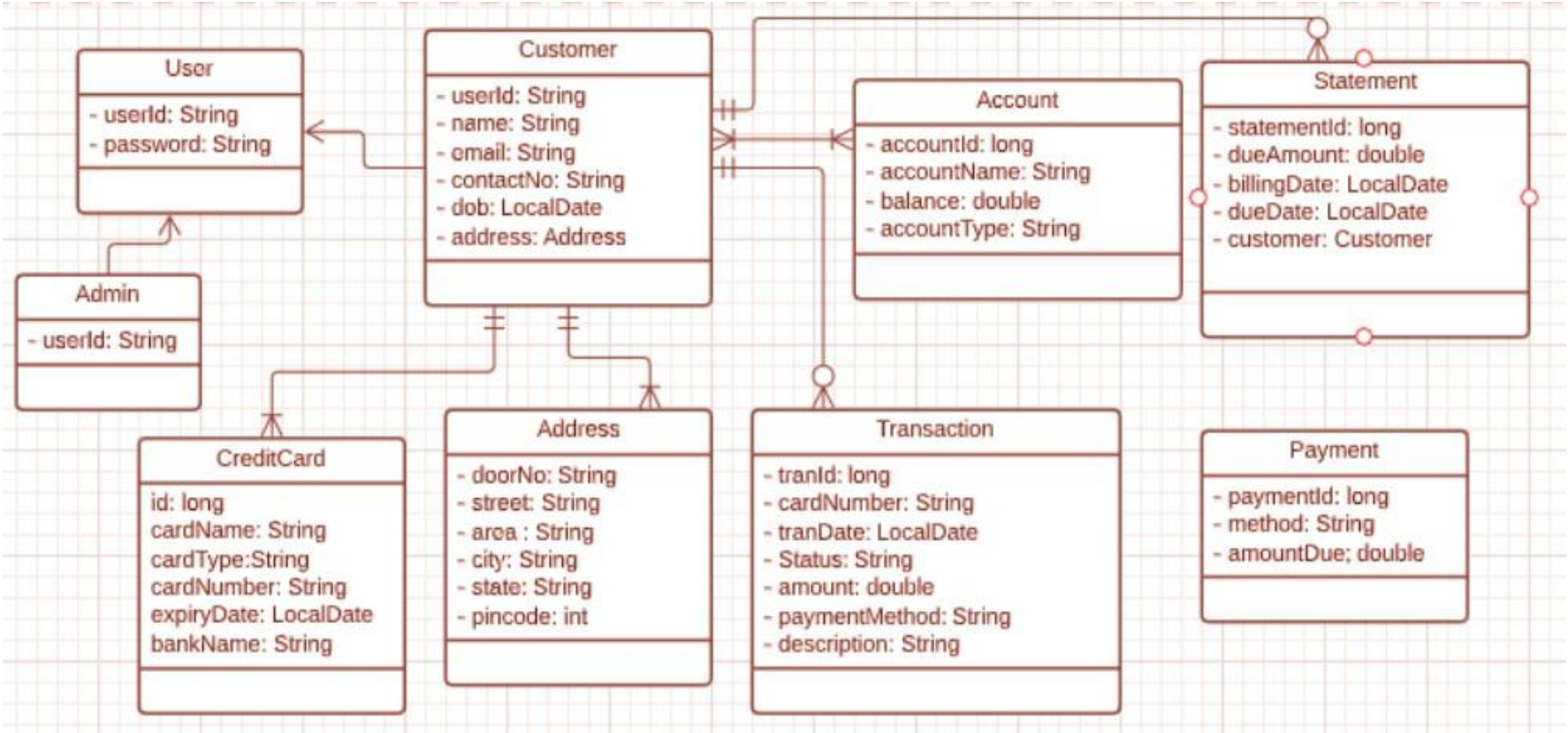
Spring Boot **Junit + Mockito**

Modules

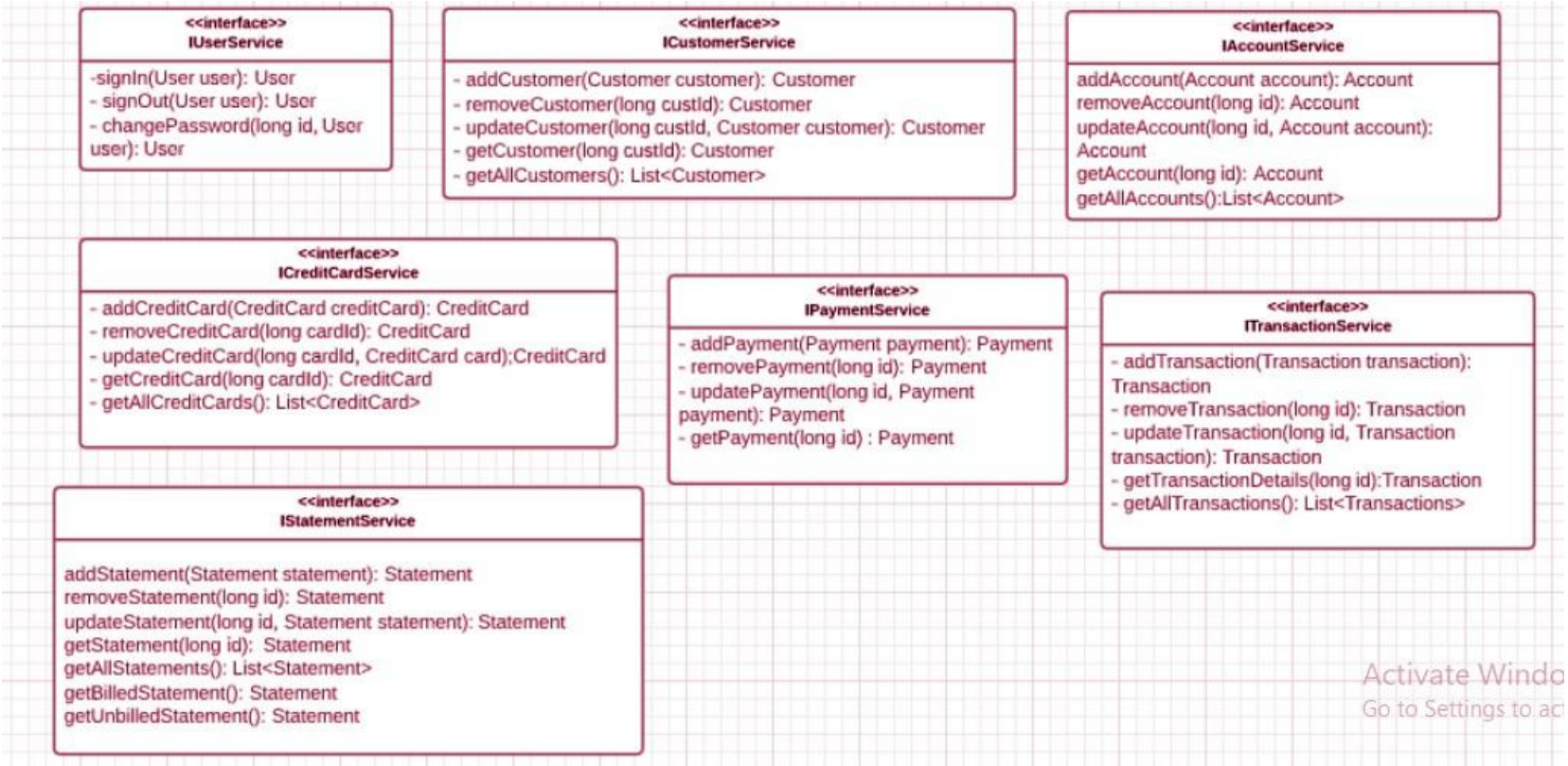
Modules


Login Module
Customer Module
Bank Account Module
Statement Module
Credit Card Module
Address Module
Payment Module
Transaction Module

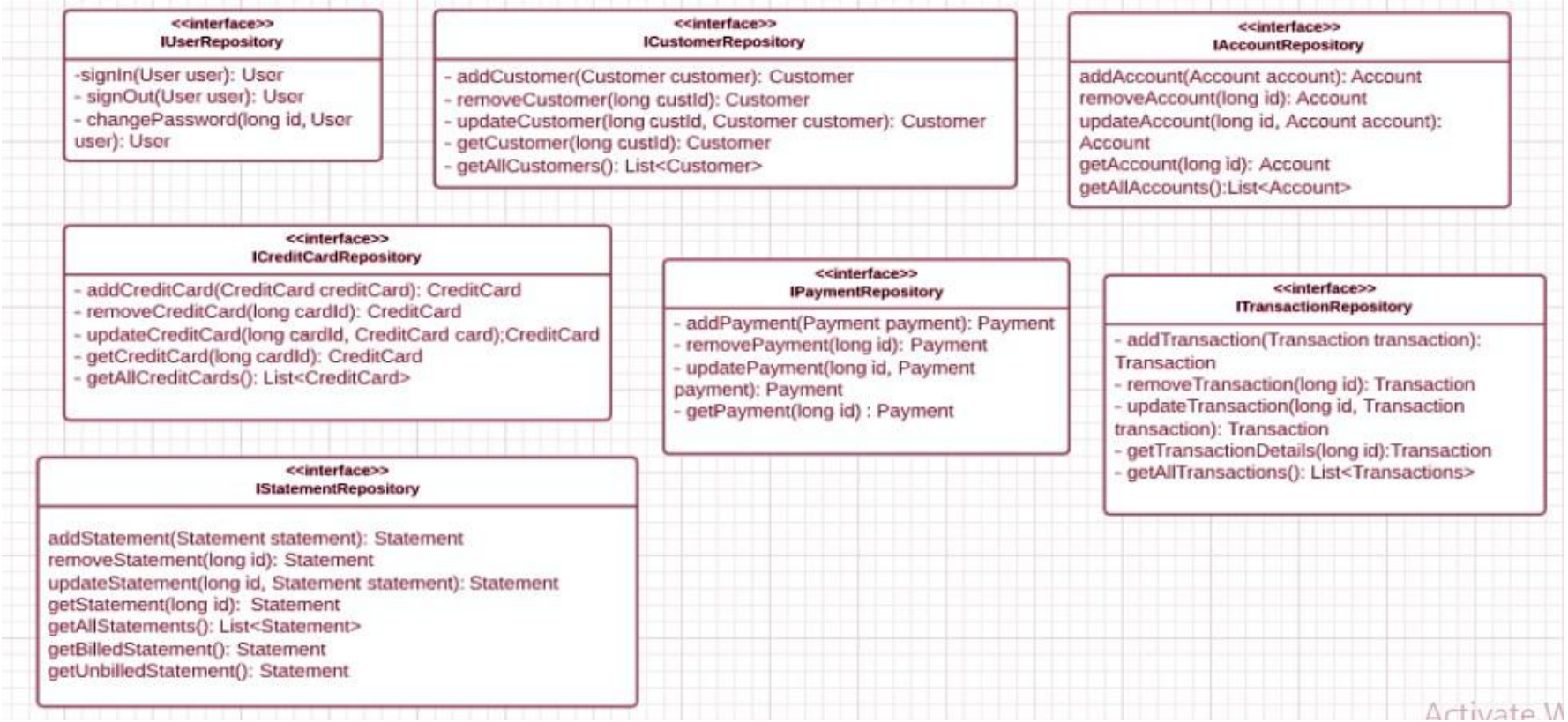
Class Design (Mappings)



Service Interface



Repository


<<interface>>
IUserRepository
-signIn(User user): User
- signOut(User user): User
- changePassword(long id, User user): User
<<interface>>
ICustomerRepository
- addCustomer(Customer customer): Customer
- removeCustomer(long custId): Customer
- updateCustomer(long custId, Customer customer): Customer
- getCustomer(long custId): Customer
- getAllCustomers(): List<Customer>
<<interface>>
IAccountRepository
addAccount(Account account): Account
removeAccount(long id): Account
updateAccount(long id, Account account): Account
getAccount(long id): Account
getAllAccounts():List<Account>
<<interface>>
ICreditCardRepository
- addCreditCard(CreditCard creditCard): CreditCard
- removeCreditCard(long cardId): CreditCard
- updateCreditCard(long cardId, CreditCard card):CreditCard
- getCreditCard(long cardId): CreditCard
- getAllCreditCards(): List<CreditCard>
<<interface>>
IPaymentRepository
- addPayment(Payment payment): Payment
- removePayment(long id): Payment
- updatePayment(long id, Payment payment): Payment
- getPayment(long id) : Payment
<<interface>>
ITransactionRepository
- addTransaction(Transaction transaction): Transaction
- removeTransaction(long id): Transaction
- updateTransaction(long id, Transaction transaction): Transaction
- getTransactionDetails(long id):Transaction
- getAllTransactions(): List<Transactions>
<<interface>>
IStatementRepository
addStatement(Statement statement): Statement
removeStatement(long id): Statement
updateStatement(long id, Statement statement): Statement
getStatement(long id): Statement
getAllStatements(): List<Statement>
getBilledStatement(): Statement
getUnbilledStatement(): Statement

Extra Functionality

1. Generating Pdf for the billed and unbilled statements.
2. Generating Pdf if the customer wants to see the details of all successful transactions.
3. Sending email with an attachment.
4. Performing Junit testing using mockito.
5. Performing global and custom exception handling as per requirements.
6. Performing validation on required fields.
7. Custom Queries has been added to automate and enhance the functionalities of transaction and statement.

Project Links

API Documentation Link: https://documenter.getpostman.com/view/21824992/2s7Z7TsGgQ

Github Repository Link :
https://github.com/Gaurav-Singh-01/Credit-Card-Payment

Thankyou