

# SB201: C4 Evaluation

## Note:

- All the questions are compulsory.
- The duration of the test is 3 hours.
- Don't seek help from any person/resource during the test.

## Marks Distribution is as follows:

- Question1 : 6 Marks
- Question2 : 4 marks

## Q1/- Consider the following Entity class:

```
class Customer{
    customerId: Integer;
    customerName: String;
    address: String;
    age: int
    email: String;
    mobile: String;
    password: String;
}
```

Generate the Rest API for the following Services of the **CustomerService** interface using **Spring Data JPA**:

```
public interface CustomerService{

    public Customer registerCustomer(Customer Customer)throws CustomerException;

    public Customer getCustomerById(Integer customerId)throws CustomerException;

    public Customer loginCustomer(String email, String password)throws Customer Exception;

    public Customer updateCustomerPassword(String username, String oldPassword,String newPassword)throws CustomerException ;

    public List<Customer> getAllCustomerDetails()throws CustomerException;

    public Customer deleteCustomerById(Integer customerId)throws CustomerException;

    public List<Customer> getCustomerDetailsByAddress(String address)throws CustomerException;

    public Customer updateCustomer(Customer customer)throws CustomerException;
```

```

    public List<Customer> getCustomersBetweenAge(Integer start_age Integer end_age)throws CustomerException

    public List<CustomerDTO> getNameAddressAgeOfAllCustomers()throws CustomerException;
}

```

```

class CustomerDTO{
    name String;
    address String;
    age Integer;
}

```

- Apply the following Validation rules of the User Request Data:
  - All the fields are mandatory.
  - Customer name should be min 3 and max 10 character length.
  - Customer age should be greater than 18
  - Address should be min 3 and max 15 character length
  - Email should be in proper format
  - Should be 10 digit length started with either (6,7,8,9)
  - Password should be min 8 and max 15 character length.
- Handle all the Exceptions in proper Exception Handler.

## Q2/- Consider the following Entity class:

```

class Account{
    accountNo Integer;
    accountholder name String;
    address String;
    email String;
    balance Integer;
    mobile String;
}

```

## Generate the Rest API for the following AccountService interface using Spring Data JPA.

```

public interface AccountService
{

    public Account openAccount(Account acc)throws AccountException;
}

```

```

public Account closeAccount(Integer accno)throws AccountException;

public Account depositAmount(Integer accno, Integer amount)throws AccountException;
    //after deposit amount return the updated Account object.

public Account withdrawAmount(int accno, Integer amount)throws AccountException,InsufficientFundException;

    //after withdraw amount returns the updated Account object.
//if the insufficient amount is there throw the InsufficientFundException with proper message
//if Invalid accno is there then throw the AccountException with proper message

public String transferAmount(AccountDTO dto)throws AccountException, InsufficientFundException.
    //after successful transfer, returns the Success message.
    //if Invalid srcAccno and desAccno is there then throw the AccountException with proper message
    //if the insufficient amount is there in destination account throw the InsufficientFundException with proper message
}

```

```

class AccountDTO {
    Integer srcAccno;
    Integer desAccno;
    Integer amount;
}

```

- Apply the following Validation rules of the User Request Data:
  - **All the fields are mandatory.**
  - **Opening balance of account should be minimum 10000**
- Handle all the Exceptions in proper Exception Handler.

Sample of the **application.properties** file: modify it accordingly

```

#changing the server port
server.port=8888

#db specific properties
spring.datasource.url=jdbc:mysql://localhost:3306/sb201db
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.username=root
spring.datasource.password=root

#ORM s/w specific properties
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

```