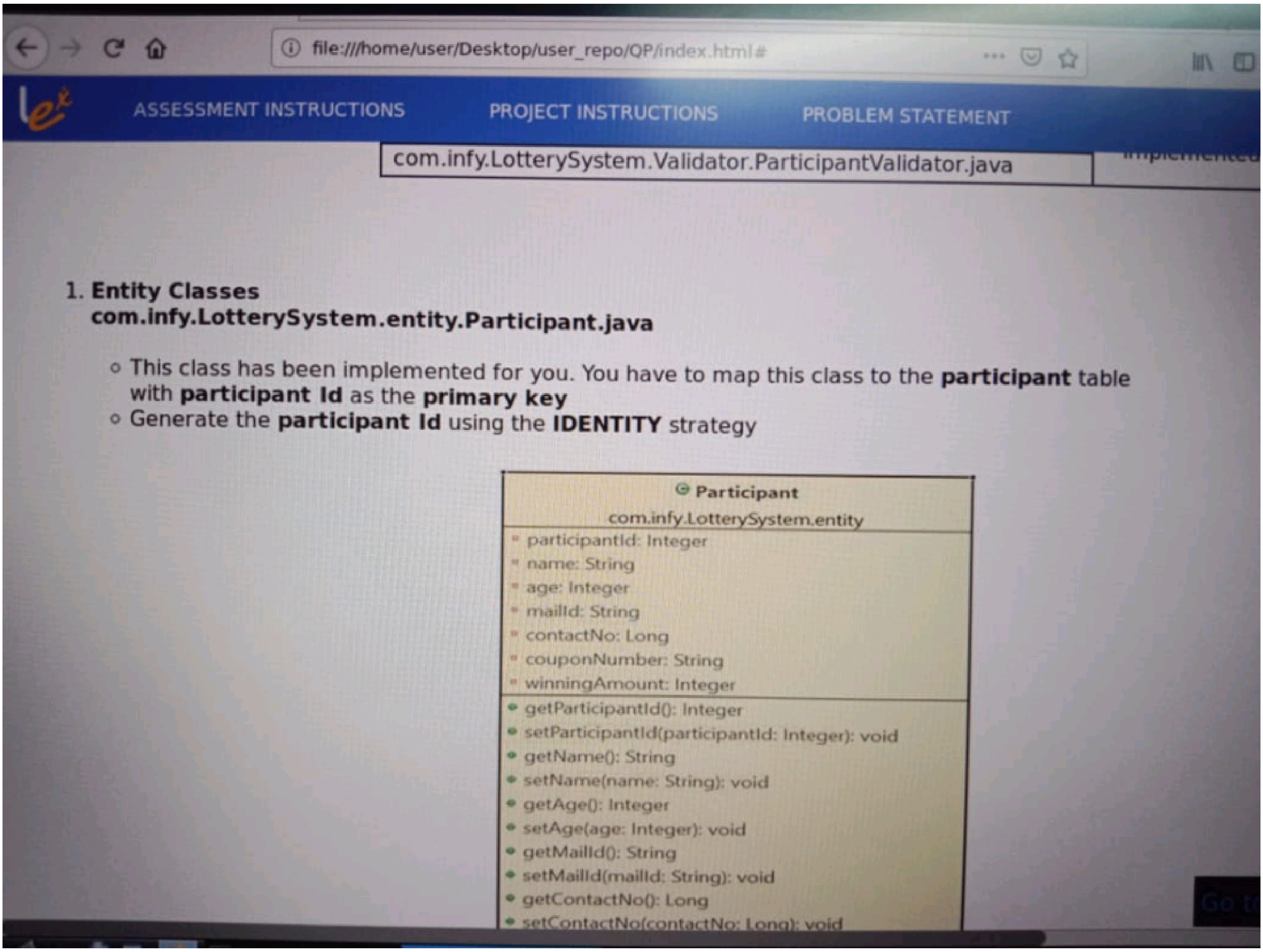
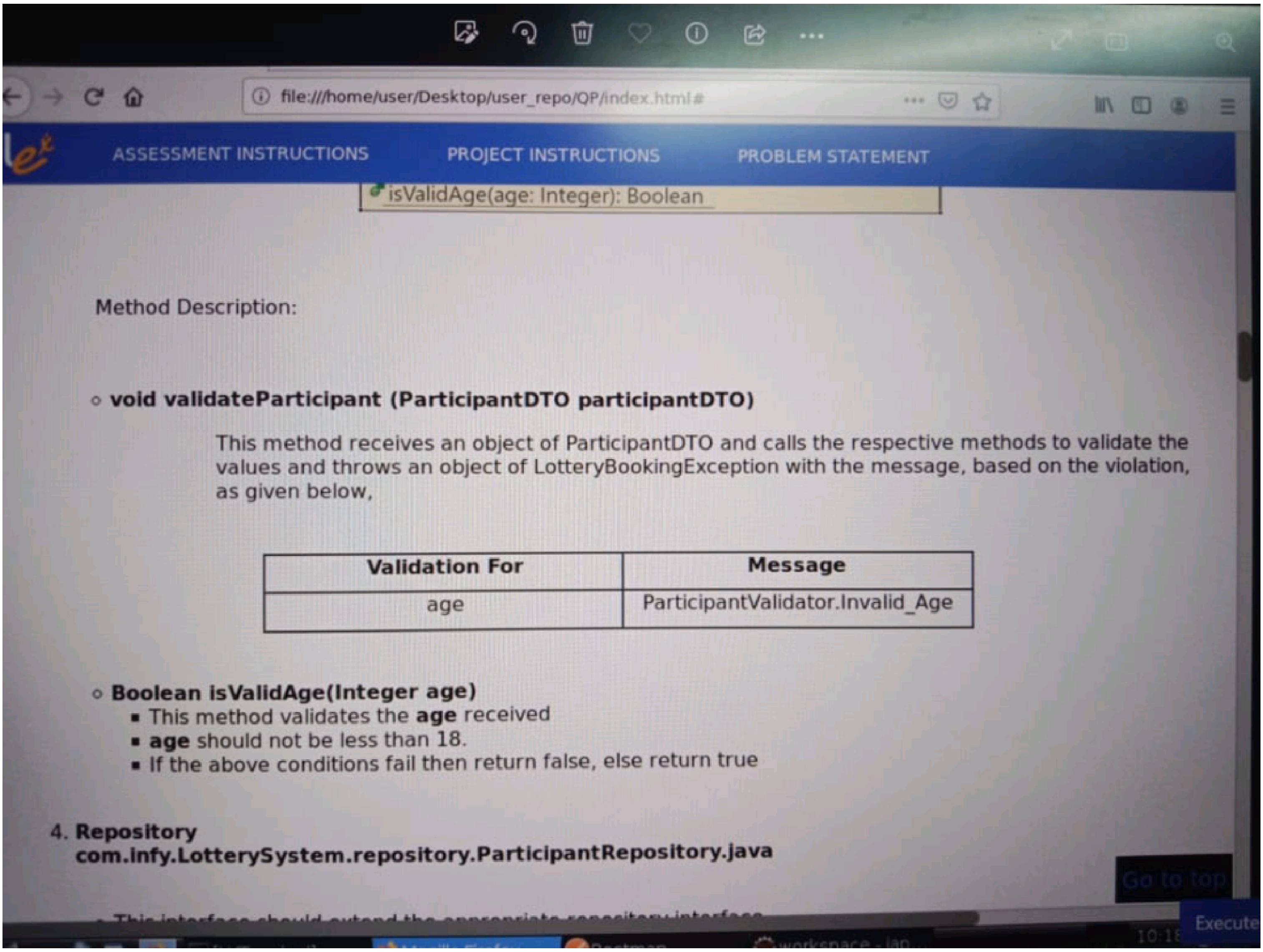


com.infy.LotterySystem.Validator.ParticipantValidator.java

1. **Entity Classes**
   com.infy.LotterySystem.entity.Participant.java

   - This class has been implemented for you. You have to map this class to the **participant** table with **participant Id** as the **primary key**
   - Generate the **participant Id** using the **IDENTITY** strategy

```
Participant
com.infy.LotterySystem.entity
participantId: Integer
name: String
age: Integer
mailId: String
contactNo: Long
couponNumber: String
winningAmount: Integer
getParticipantId(): Integer
setParticipantId(participantId: Integer): void
getName(): String
setName(name: String): void
getAge(): Integer
setAge(age: Integer): void
getMailId(): String
setMailId(mailId: String): void
getContactNo(): Long
setContactNo(contactNo: Long): void
```



isValidAge(age: Integer): Boolean

Method Description:

- **void validateParticipant (ParticipantDTO participantDTO)**

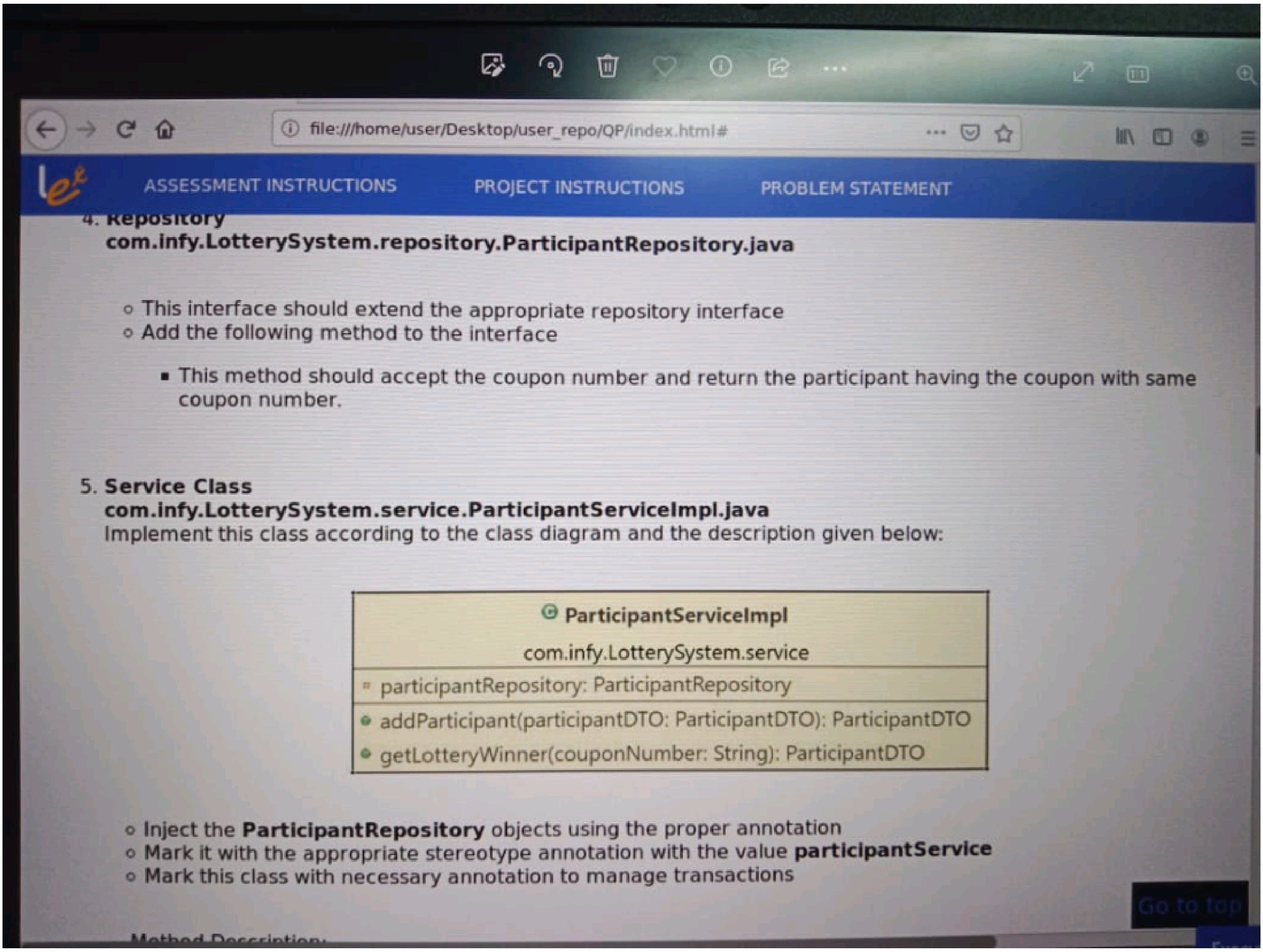
  This method receives an object of ParticipantDTO and calls the respective methods to validate the values and throws an object of LotteryBookingException with the message, based on the violation, as given below,

Validation For	Message
age	ParticipantValidator.Invalid_Age

- **Boolean isValidAge(Integer age)**
  - This method validates the **age** received
  - **age** should not be less than 18.
  - If the above conditions fail then return false, else return true

4. **Repository**
   com.infy.LotterySystem.repository.ParticipantRepository.java

   - This interface should extend the appropriate repository interface



4. **Repository**
com.infy.LotterySystem.repository.ParticipantRepository.java

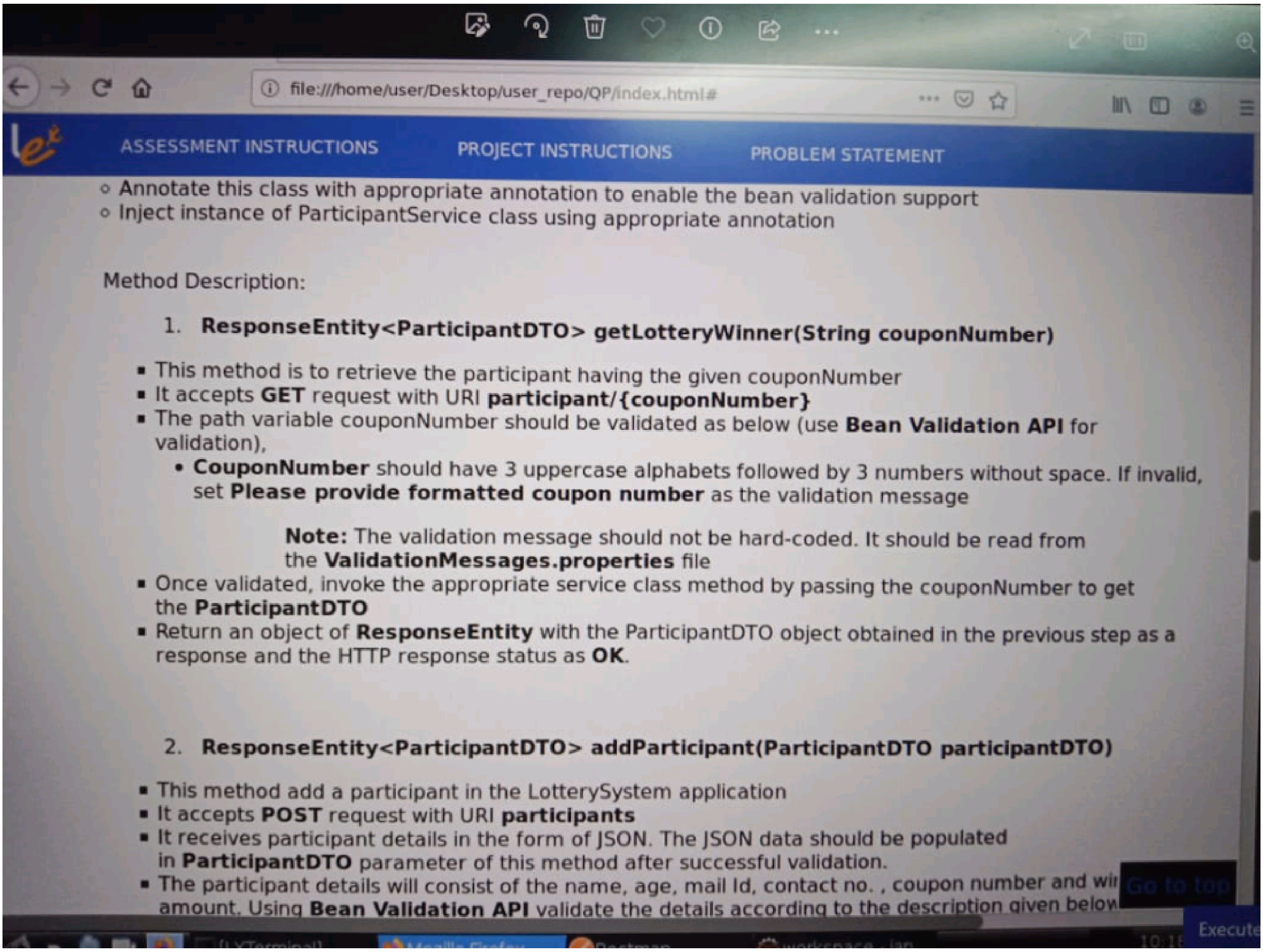
- This interface should extend the appropriate repository interface
- Add the following method to the interface
  - This method should accept the coupon number and return the participant having the coupon with same coupon number.

5. **Service Class**
com.infy.LotterySystem.service.ParticipantServiceImpl.java
Implement this class according to the class diagram and the description given below:

ParticipantServiceImpl com.infy.LotterySystem.service
participantRepository: ParticipantRepository
addParticipant(participantDTO: ParticipantDTO): ParticipantDTO getLotteryWinner(couponNumber: String): ParticipantDTO

- Inject the **ParticipantRepository** objects using the proper annotation
- Mark it with the appropriate stereotype annotation with the value **participantService**
- Mark this class with necessary annotation to manage transactions

Method Description:



- Annotate this class with appropriate annotation to enable the bean validation support
- Inject instance of ParticipantService class using appropriate annotation

Method Description:

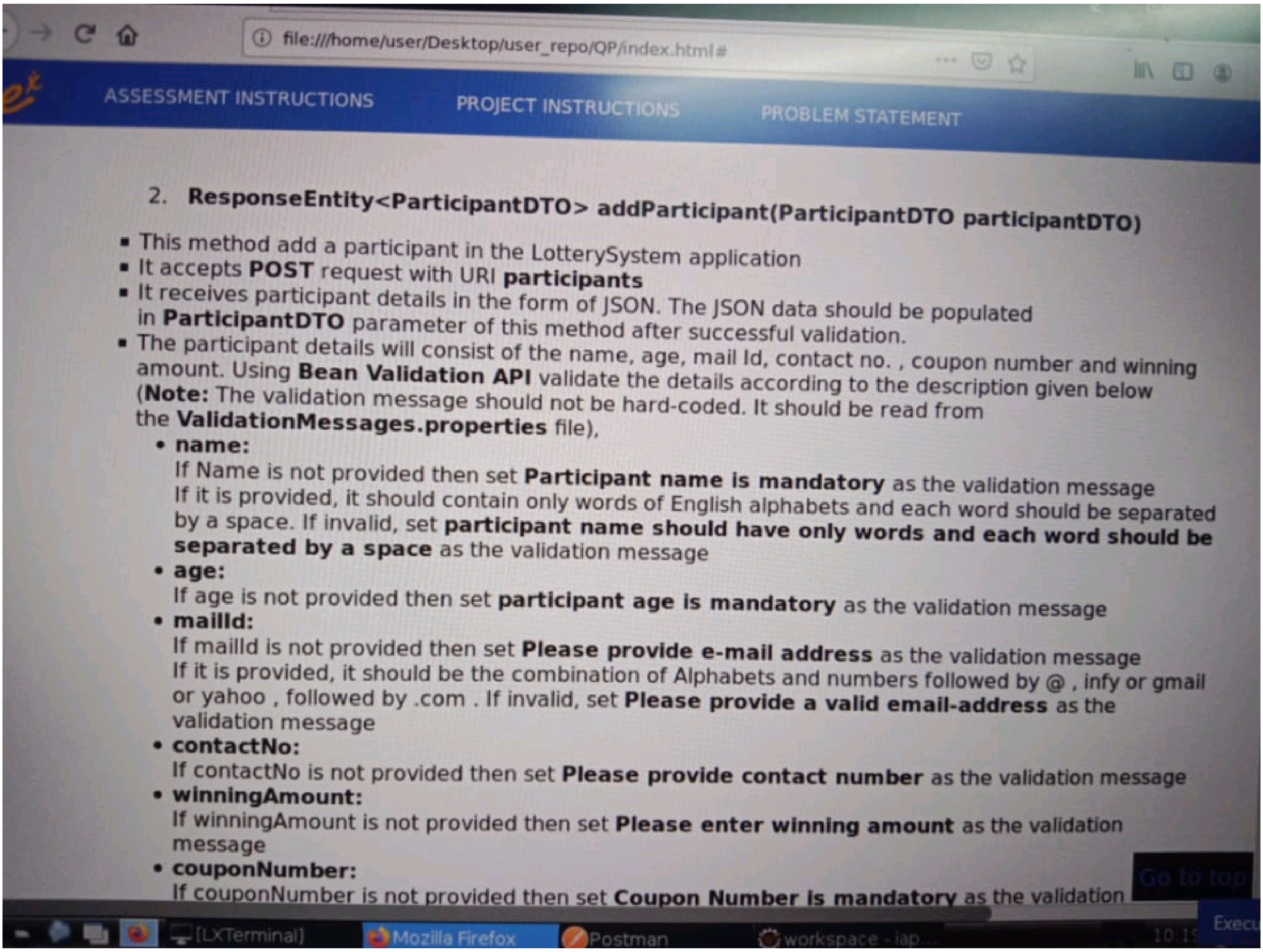
1. **ResponseEntity<ParticipantDTO> getLotteryWinner(String couponNumber)**

- This method is to retrieve the participant having the given couponNumber
- It accepts **GET** request with URI **participant/{couponNumber}**
- The path variable couponNumber should be validated as below (use **Bean Validation API** for validation),
  - **CouponNumber** should have 3 uppercase alphabets followed by 3 numbers without space. If invalid, set **Please provide formatted coupon number** as the validation message

    Note: The validation message should not be hard-coded. It should be read from the **ValidationMessages.properties** file
- Once validated, invoke the appropriate service class method by passing the couponNumber to get the **ParticipantDTO**
- Return an object of **ResponseEntity** with the ParticipantDTO object obtained in the previous step as a response and the HTTP response status as **OK**.

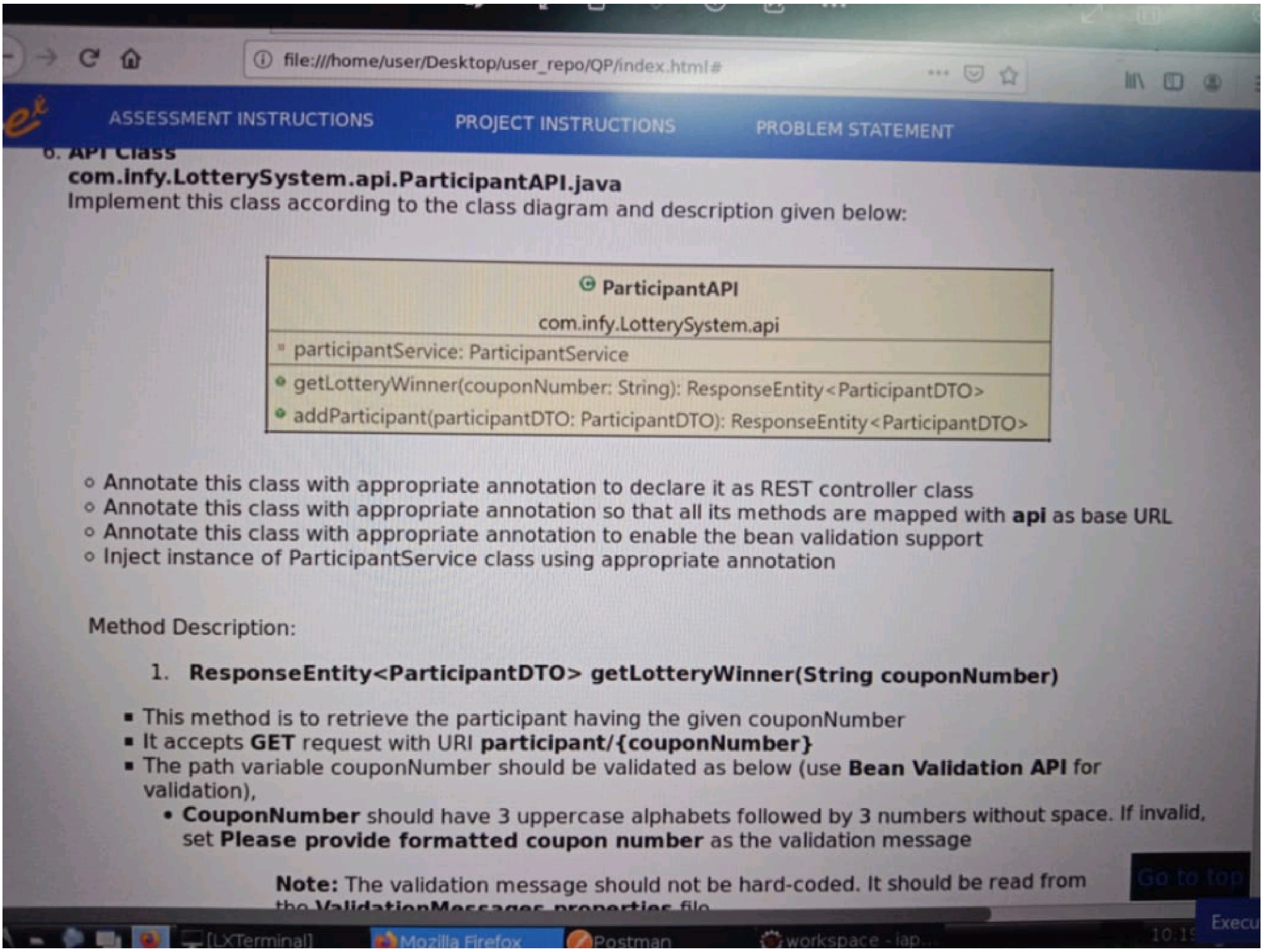
2. **ResponseEntity<ParticipantDTO> addParticipant(ParticipantDTO participantDTO)**

- This method add a participant in the LotterySystem application
- It accepts **POST** request with URI **participants**
- It receives participant details in the form of JSON. The JSON data should be populated in **ParticipantDTO** parameter of this method after successful validation.
- The participant details will consist of the name, age, mail Id, contact no. , coupon number and wi[illegible] amount. Using **Bean Validation API** validate the details according to the description given below



2. **ResponseEntity<ParticipantDTO> addParticipant(ParticipantDTO participantDTO)**

- This method add a participant in the LotterySystem application
- It accepts **POST** request with URI **participants**
- It receives participant details in the form of JSON. The JSON data should be populated in **ParticipantDTO** parameter of this method after successful validation.
- The participant details will consist of the name, age, mail Id, contact no. , coupon number and winning amount. Using **Bean Validation API** validate the details according to the description given below (**Note:** The validation message should not be hard-coded. It should be read from the **ValidationMessages.properties** file),
  - **name:**
    If Name is not provided then set **Participant name is mandatory** as the validation message
    If it is provided, it should contain only words of English alphabets and each word should be separated by a space. If invalid, set **participant name should have only words and each word should be separated by a space** as the validation message
  - **age:**
    If age is not provided then set **participant age is mandatory** as the validation message
  - **mailId:**
    If mailId is not provided then set **Please provide e-mail address** as the validation message
    If it is provided, it should be the combination of Alphabets and numbers followed by @ , infy or gmail or yahoo , followed by .com . If invalid, set **Please provide a valid email-address** as the validation message
  - **contactNo:**
    If contactNo is not provided then set **Please provide contact number** as the validation message
  - **winningAmount:**
    If winningAmount is not provided then set **Please enter winning amount** as the validation message
  - **couponNumber:**
    If couponNumber is not provided then set **Coupon Number is mandatory** as the validation



6. API Class
com.infy.LotterySystem.api.ParticipantAPI.java
Implement this class according to the class diagram and description given below:

ParticipantAPI com.infy.LotterySystem.api
participantService: ParticipantService
getLotteryWinner(couponNumber: String): ResponseEntity<ParticipantDTO> addParticipant(participantDTO: ParticipantDTO): ResponseEntity<ParticipantDTO>

- Annotate this class with appropriate annotation to declare it as REST controller class
- Annotate this class with appropriate annotation so that all its methods are mapped with **api** as base URL
- Annotate this class with appropriate annotation to enable the bean validation support
- Inject instance of ParticipantService class using appropriate annotation

Method Description:

1. **ResponseEntity<ParticipantDTO> getLotteryWinner(String couponNumber)**

- This method is to retrieve the participant having the given couponNumber
- It accepts **GET** request with URI **participant/{couponNumber}**
- The path variable couponNumber should be validated as below (use **Bean Validation API** for validation),
  - **CouponNumber** should have 3 uppercase alphabets followed by 3 numbers without space. If invalid, set **Please provide formatted coupon number** as the validation message

Note: The validation message should not be hard-coded. It should be read from the **ValidationMessages.properties** file