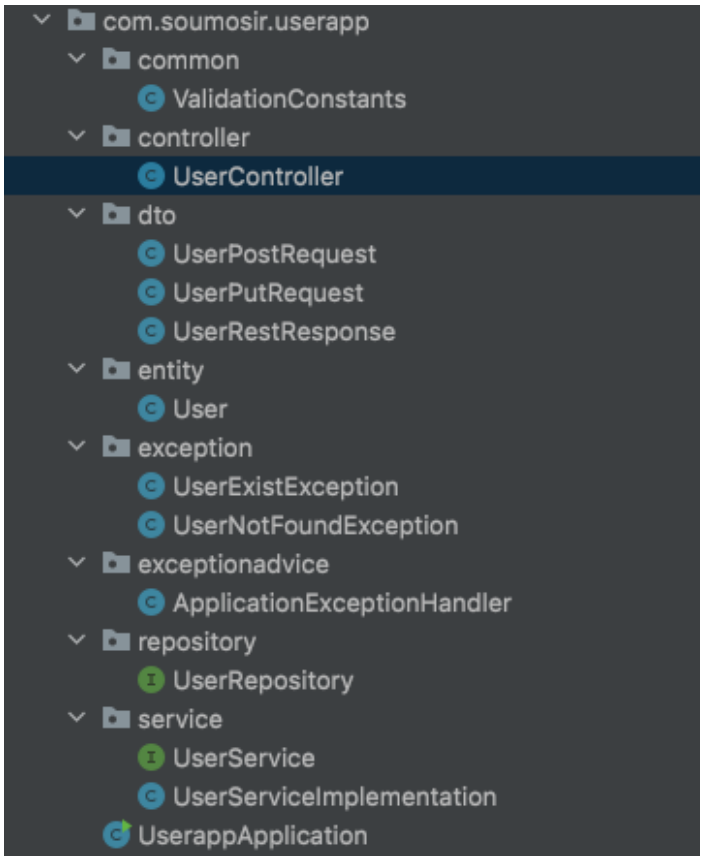
Question:- Backend user app

Answer:-
(Installation instructions at the end of pdf)

I have developed the application in Java Spring (Java version - openjdk 11.0.12 2021-07-20)
This is what the project structure looks like :-



I have segregated the Application in

1. **Entity** - which is the User object

2. Also, there are **Data Transfer Objects** for REST API call for PUT, POST, and GET

3. There is a **Controller** which is the REST API Controller for the User and it maps to UserService

4. For the **Service** layer , there is UserService interface has a concrete implementation called UserServiceImplementation

5. There is **ExceptionHandler** - ApplicationExceptionHandler which handles the exceptions and displayed appropriate messages to the user in the REST API response.

6. Lastly, the Service layer communicates with the **Repository** layer((JPA) for persisting the User's Data.

API examples according to requirements:-

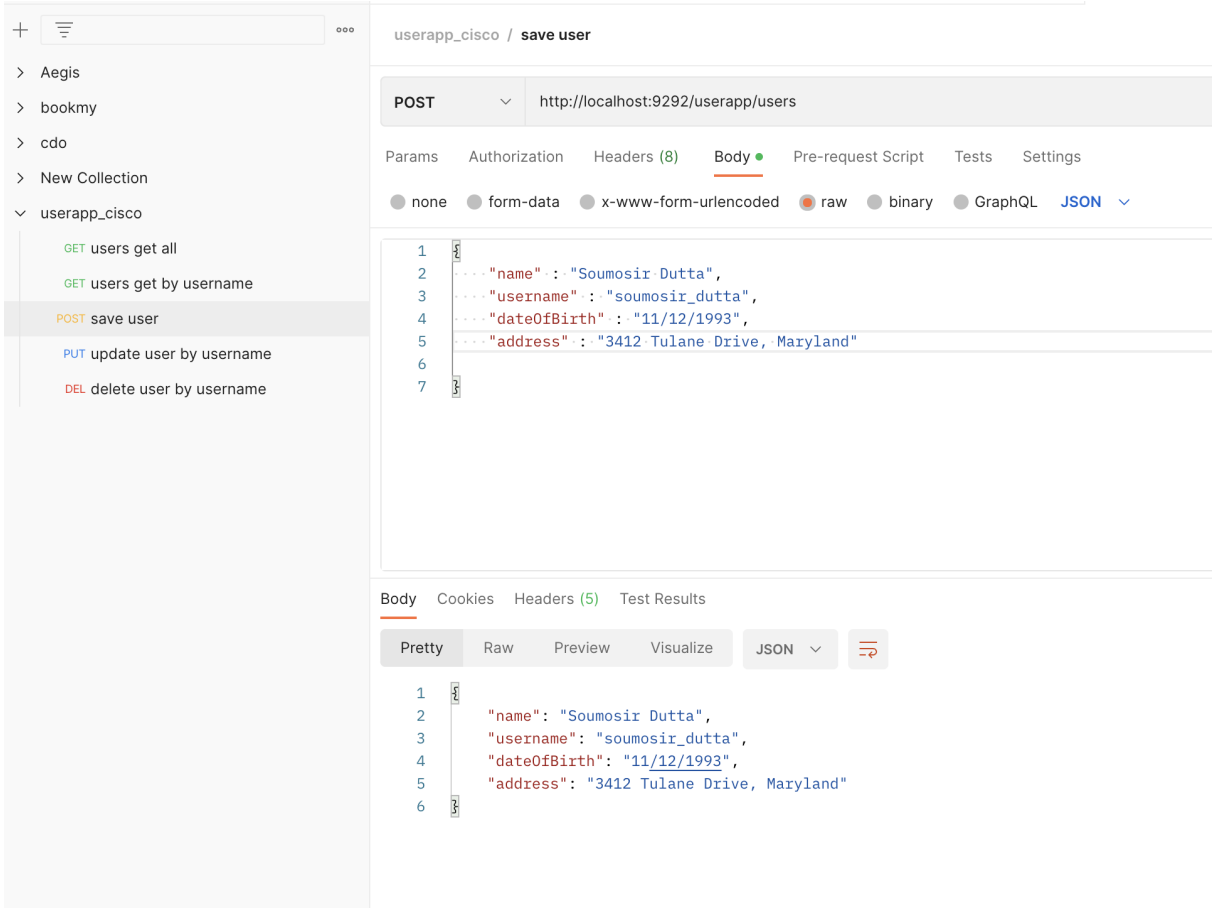
1. POST /userapp/users should create a new user based on the given values.

- name - should be more than 2 characters and less than 20 characters
- address - should be more than 4 characters and less than 200 characters
- dateOfBirth - should be of format MM/dd/yyyy and a past date
- username - should be alphanumeric A-Z, a-z, 0-9 and can have _ and . characters except for beginning and end

The curl request for that is

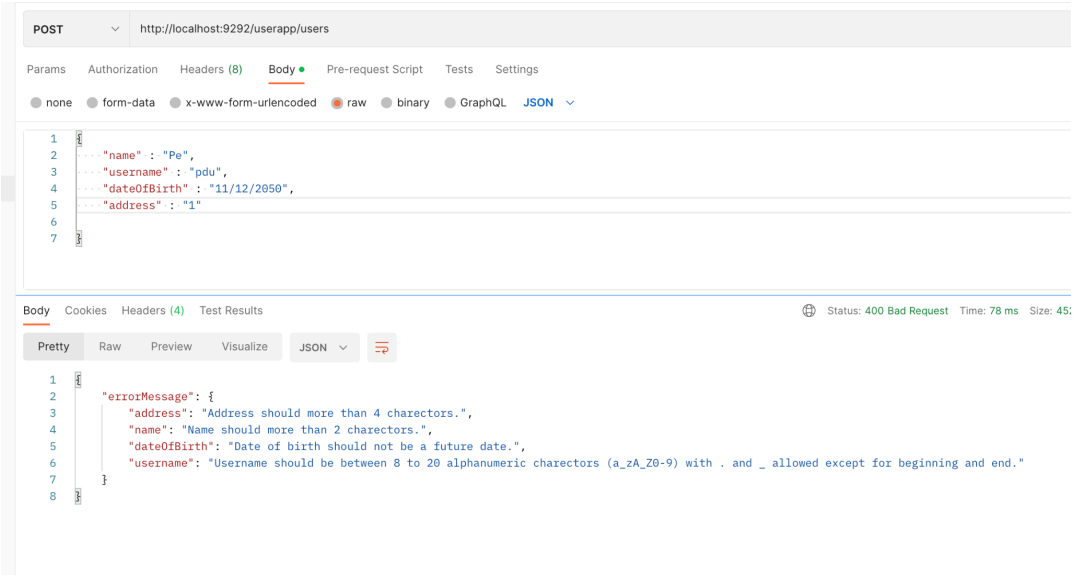
```
curl --location --request POST 'http://localhost:9292/userapp/users' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name" : "Soumosir Dutta",
  "username" : "soumosir_dutta",
  "dateOfBirth" : "11/12/1993",
  "address" : "3412 Tulane Drive, Maryland"
}'
```

Example

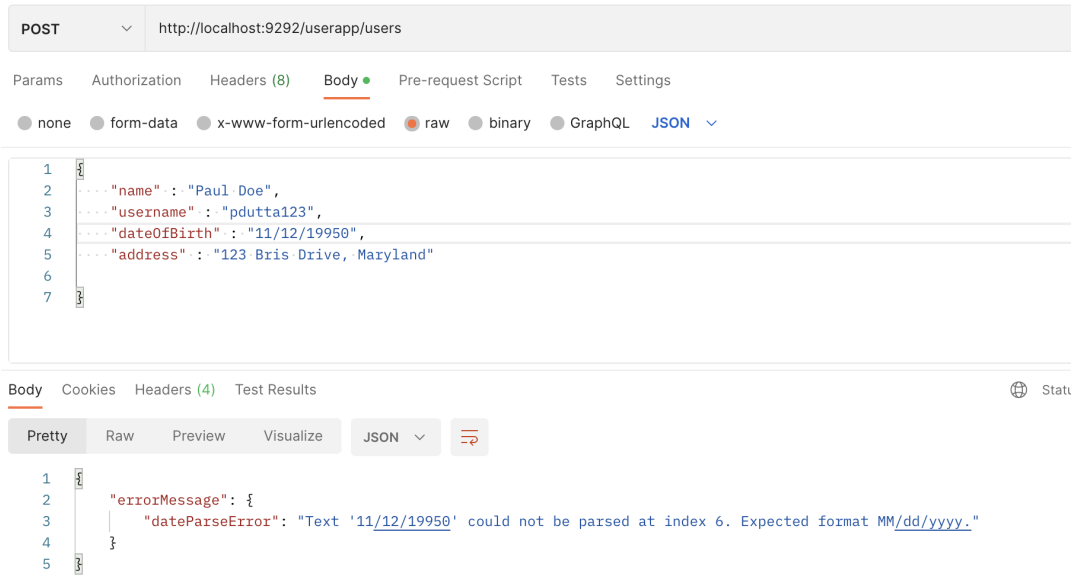


Validations for the fields which is given by a json with 400 status code
{
  errorMessage : { field1 : <reason> }

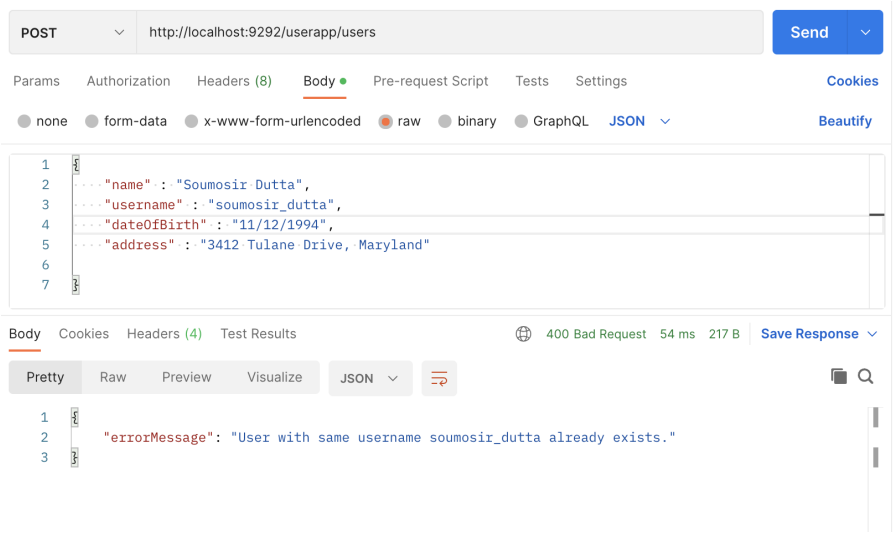
Example:-



Some more validation on error in date format



Validation error if username already exists

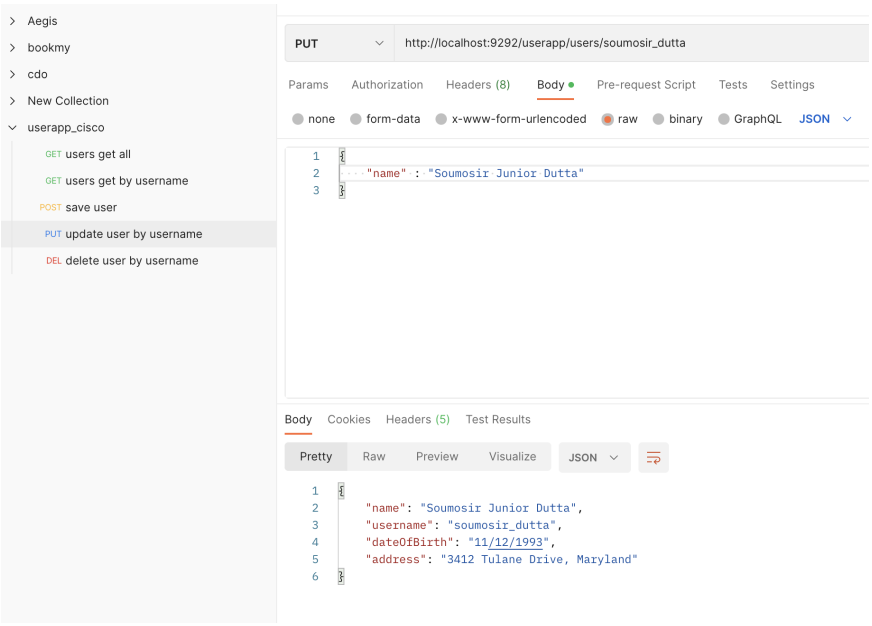


2. PUT /userapp/users/<username> should update the user with the given information.

The curl request for that is

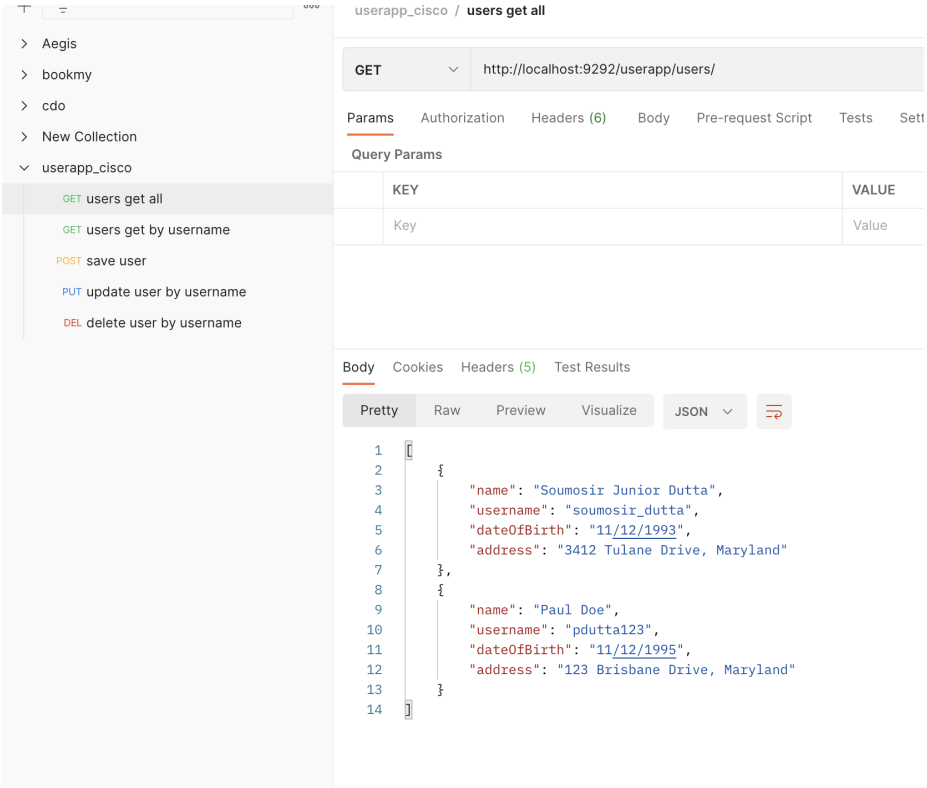
```
curl --location --request PUT 'http://localhost:9292/userapp/users/soumosir_dutta' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name" : "Soumosir Junior Dutta"
}'
```

Here we are updating only the name field



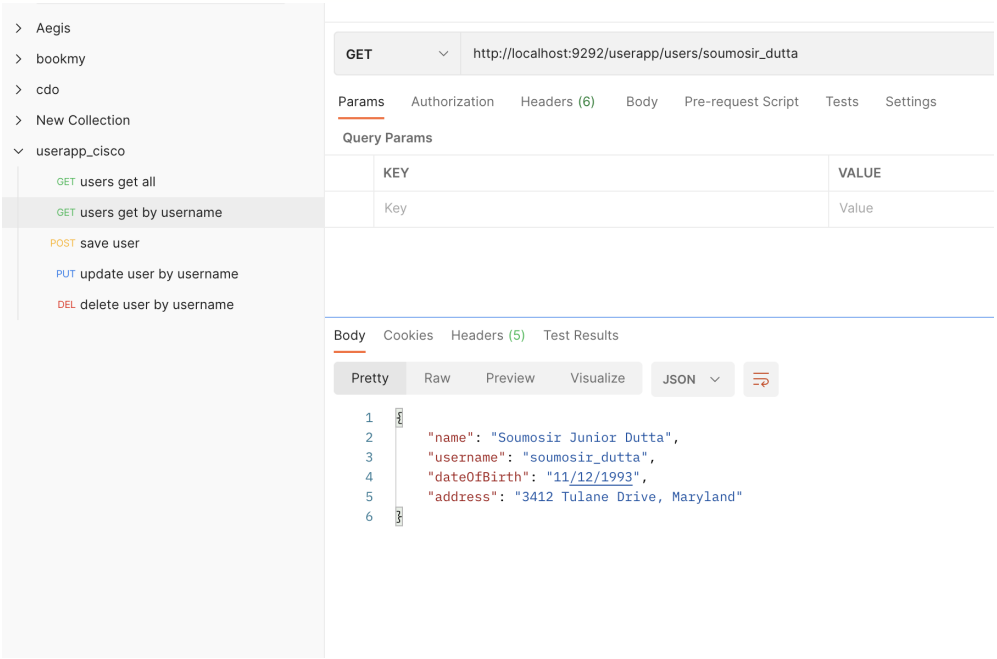
3. GET /userapp/users should return a list of users.

curl --location --request GET 'http://localhost:9292/userapp/users'



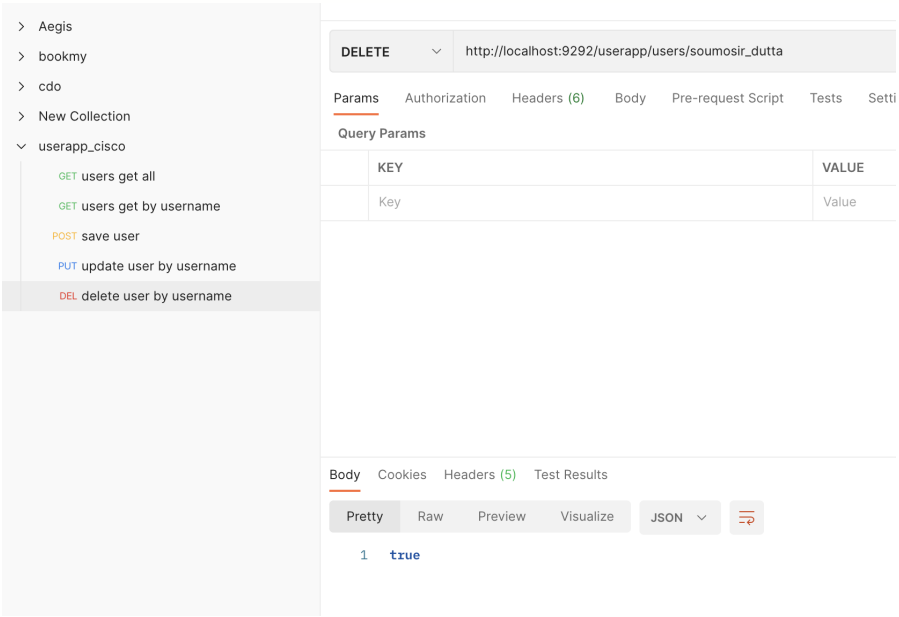
4. GET /userapp/users/<username> should return details of a specific user.

curl --location --request GET 'http://localhost:9292/userapp/users/soumosir_dutta'

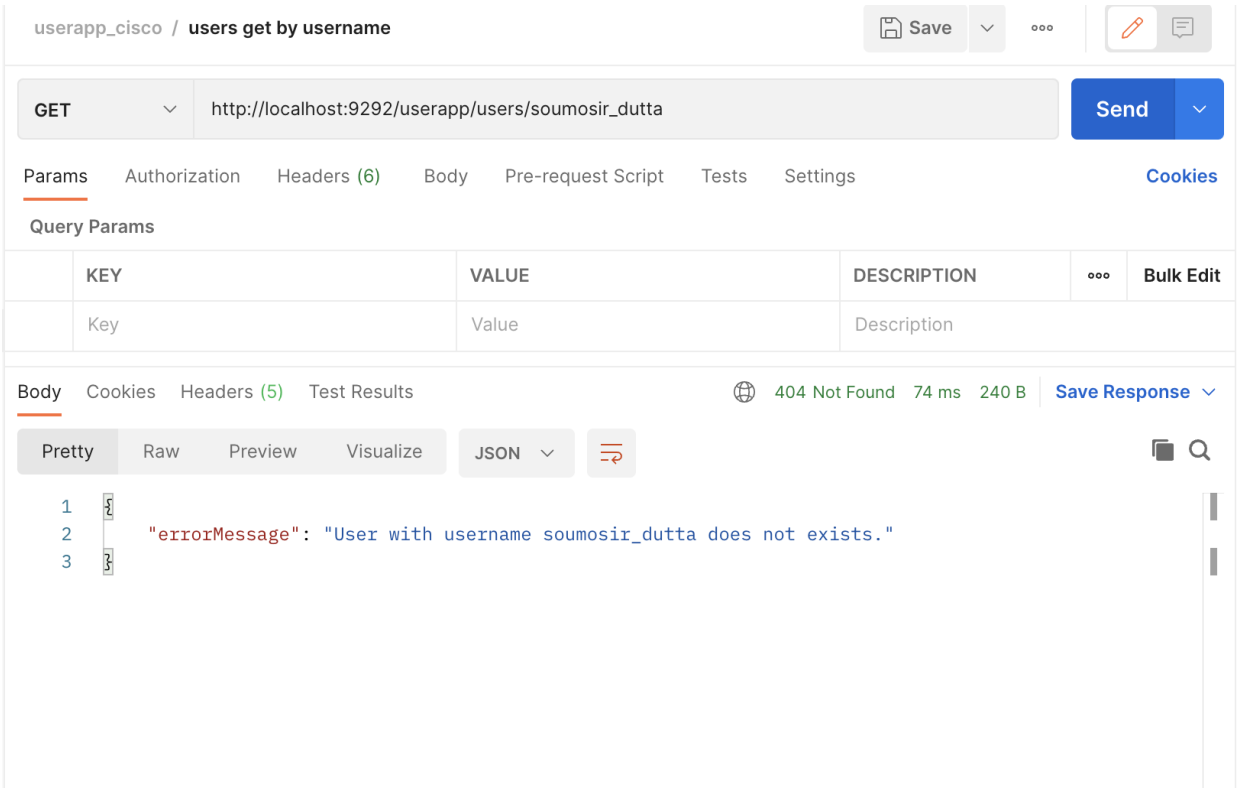


5. DELETE /userapp/users/<username> should delete the specified user.

curl --location --request DELETE 'http://localhost:9292/userapp/users/soumosir_dutta'



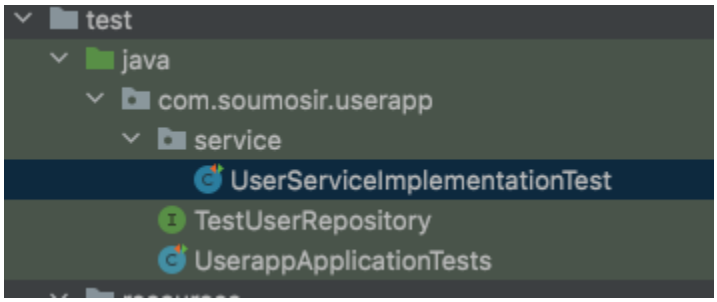
Get call to verify it is actually deleted

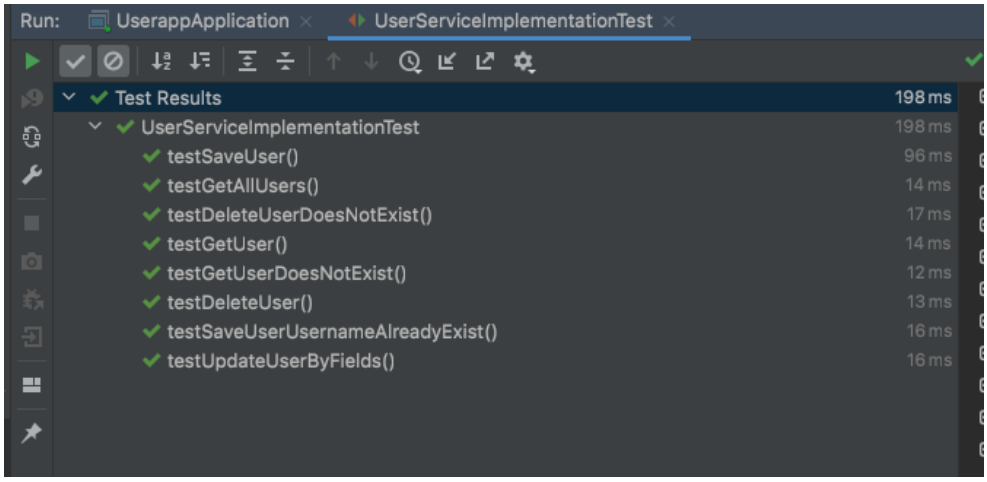


Also, there are validations in place if there is no user with a username.

UNIT TEST CASES

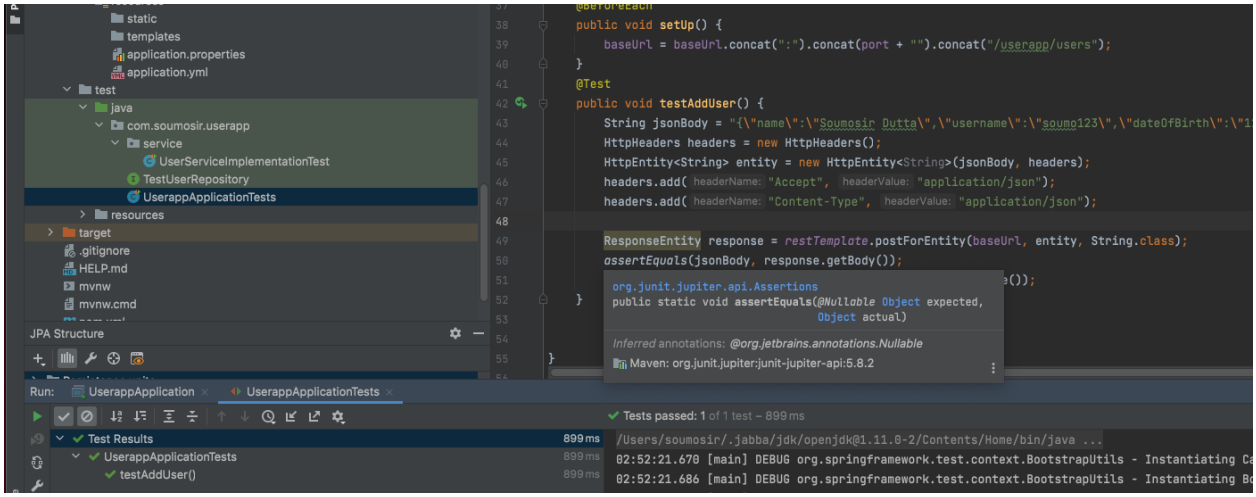
I have written some unit test cases for UserServiceImplementation





src/test/java/com/soumosir/userapp/service/UserServiceImplementationTest.java

There is a simple integration test written for add user.



src/test/java/com/soumosir/userapp/UserappApplicationTests.java

Logging

There is some logging functionality added with `@Slf4j`

The logs are stored in **customlog.txt** in the root directory.

The logs for tests are stored in customlogTest.txt in root

PTO

Test Case Scenarios:-

Note: Separate Test cases can be created for all edge conditions:-

INVALID USER POST/PUT FIELDS

1. Name < 3 characters
2. Name > 200 characters
3. Address < 5 characters
4. Address > 5 characters
5. Date of birth - future date
6. Date of birth - wrong format other than MM/dd/yyyy
7. Username - less than 8 characters
8. Username - more than 20 characters
9. Username - non alphanumeric character with . or _ beginning
   Example of invalid username “abc 123445” , “__ab2222c”,”.qwerty123”

Consodilated

Sl_no	Description
1	Save user with invalid fields - name < 3 and name > 20 - address < 5 and address>200 - date of birth - future date - username - not alphanumeric number and < 8 or > 20

Input

```
POST 'http://localhost:9292/userapp/users' \
'{
  "name" : "Pa",
  "username" : "pdu w",
  "dateOfBirth" : "11/12/2050",
  "address" : "12"
}'
```

Expected Output

```
{
  "errorMessage": {
    "address": "Address should more than 4 charectors.",
    "name": "Name should more than 2 charectors.",
    "dateOfBirth": "Date of birth should not be a future date.",
    "username": "Username should be between 8 to 20 alphanumeric charectors (a_zA_Z0-9) with . and _ allowed except for beginning and end."
  }
}
```

Pre-Condition: Create a user with username - soumosir_dutta

Sl_no	Description
2	Save another user with username - soumosir_dutta

Input

```
curl --location --request POST 'http://localhost:9292/userapp/users' \
--header 'Content-Type: application/json' \
--data-raw '{
   "name" : "Soumosir Dutta",
   "username" : "soumosir_dutta",
   "dateOfBirth" : "11/12/1993",
   "address" : "3412 Tulane Drive, Maryland"
}'
curl --location --request POST 'http://localhost:9292/userapp/users' \
--header 'Content-Type: application/json' \
--data-raw '{
   "name" : "Soumosir Dutta",
   "username" : "soumosir_dutta",
   "dateOfBirth" : "11/12/1993",
   "address" : "3412 Tulane Drive, Maryland"
}'
```

Expected Output

```
{
   "errorMessage": "User with same username soumosir_dutta already exists." }
```

Pre-Condition: Create a user with username - soumosir_dutta

Sl_no	Description
2	Update user with username - soumosir_dutta , name to be “Soumosir Junior Dutta”

Input

```
curl --location --request POST 'http://localhost:9292/userapp/users' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name" : "Soumosir Dutta",
  "username" : "soumosir_dutta",
  "dateOfBirth" : "11/12/1993",
  "address" : "3412 Tulane Drive, Maryland"
}'
curl --location --request PUT 'http://localhost:9292/userapp/users/soumosir_dutta' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name" : "Soumosir Junior Dutta"
}'
```

Expected Output (Name is successfully updated)


```
{
  "name": "Soumosir Junior Dutta",
  "username": "soumosir_dutta",
  "dateOfBirth": "11/12/1993",
  "address": "3412 Tulane Drive, Maryland"
}
```

uri = /userapp/users/

sl_no	description	input	Expected output
3	Pre-Condition : create user with username soumosir_dutta Delete an user which exists	DELETE uri/<username> curl --location --request DELETE 'http://localhost:9292/userapp/users/soumosir_dutta'	true

4	Delete an user which doesnot exist	Delete uri/<username>
		curl --location --request DELETE 'http://localhost:9292/userapp/users/soumosir_dutta3'
			404 {
			"errorMessage": "User with username soumosir_dutta3 does not exists."
			}

5	Get all users
	
	Should return list of users
		GET uri
		curl --location --request GET 'http://localhost:9292/userapp/users'
			[
			{
			"name": "Paul Doe",
			"username": "pdutta123",
			"dateOfBirth": "11/12/1995",
			"address": "123 Brisbane Drive, Maryland"
			},
			{
			"name": "Soumosir Junior Dutta",
			"username": "soumosir_dutta",
			"dateOfBirth": "11/12/1993",
			"address": "3412 Tulane Drive, Maryland"
			}
			]

6	Pre-condition :create User with pdutta123
	
	Get user by username
	Should return a user in json
		GET URI/<username>
		curl --location --request GET 'http://localhost:9292/userapp/users/pdutta123'
			{
			"name": "Paul Doe",
			"username": "pdutta123",
			"dateOfBirth": "11/12/1995",
			"address": "123 Brisbane Drive, Maryland"
			}

7	Get user by username which doesnt exists	GET URI/<username>
		curl --location --request GET 'http://localhost:9292/userapp/users/pdutta12377'
			404
			{
			"errorMessage": "User with username pdutta12377 does not exists."
			}

Additional work to be done:-

1. Adding more unit test cases for userController and userServiceImplementation class
2. Adding unit test cases for Exception handler
3. Adding integration test for the remaining flow

Steps to run the application: -

1. Download java version 11

   java --version
   openjdk 11.0.12 2021-07-20
   OpenJDK Runtime Environment Microsoft-25199 (build 11.0.12+7)
   OpenJDK 64-Bit Server VM Microsoft-25199 (build 11.0.12+7, mixed mode)

2. **Unzip userapp.zip**

3. **Run the jar file directly**

   java -jar target/userapp-0.0.1-SNAPSHOT.jar

   Or

   Run in development mode

   ```
   mvn --version
   Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
   Maven home: /opt/homebrew/Cellar/maven/3.8.6/libexec
   Java version: 18.0.2, vendor: Homebrew, runtime: /opt/homebrew/Cellar/openjdk/18.0.2/libexec/openjdk.jdk/Contents/Home
   Default locale: en_US, platform encoding: UTF-8
   OS name: "mac os x", version: "13.0.1", arch: "aarch64", family: "mac"
   ```

   cd userapp
   mvn install
   mvn spring-boot:run

   Or

   Open the Spring application in an ide
   Install dependencies - mvn install

   And run the
   src/main/java/com/soumosir/userapp/UserappApplication.java file

Please contact soumosirdutta@gmail.com for any setup help!