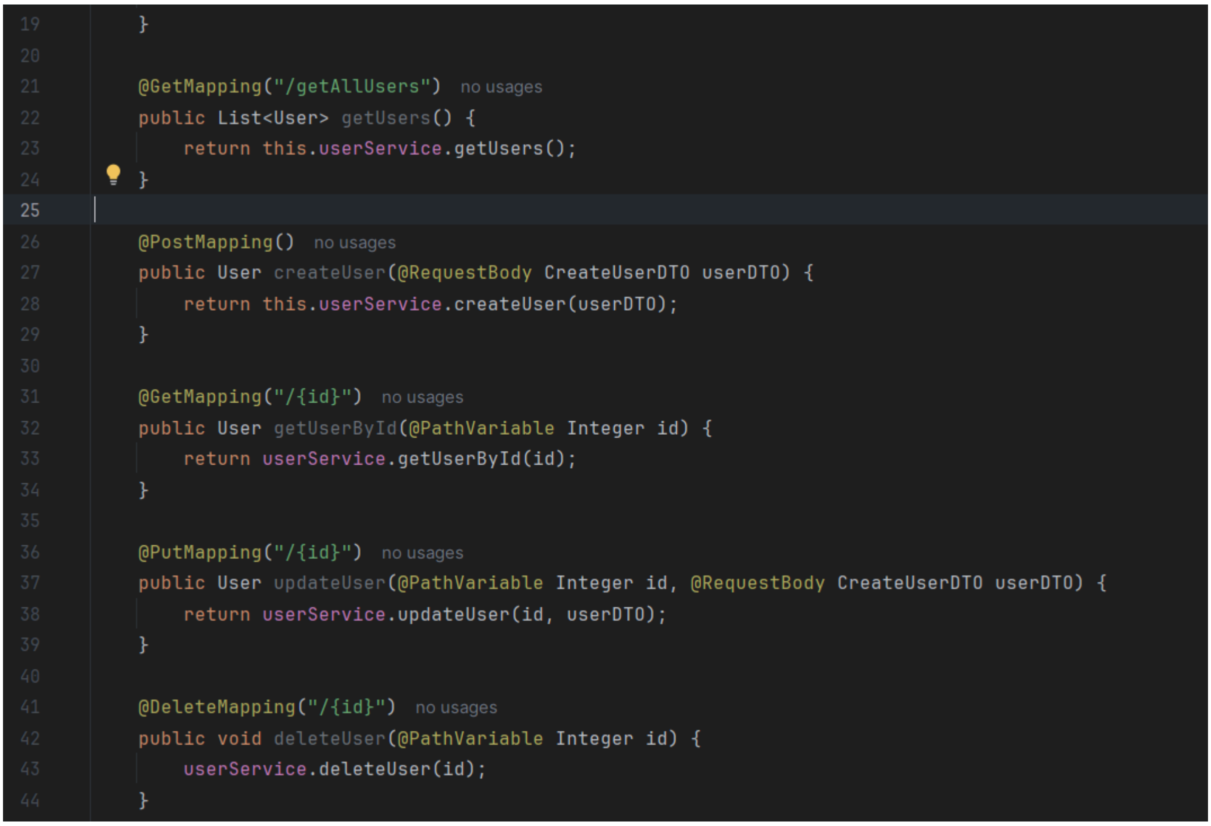
CSE434 Aspect and Service-Oriented Software Systems

Lab 2 Assignment

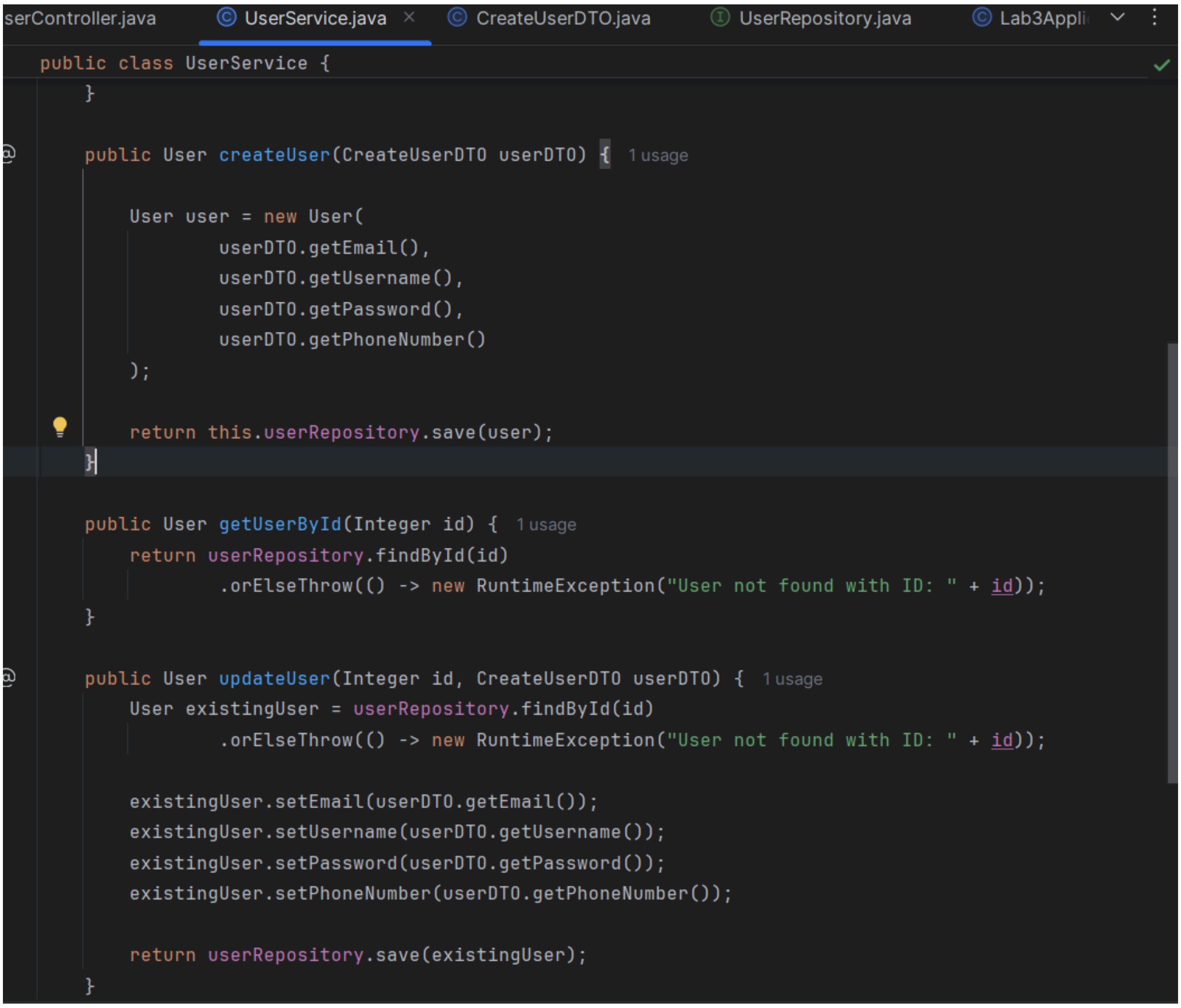
Name	Mohamed Mostafa Bedair ElMaghraby
ID	20P7732

User controller:

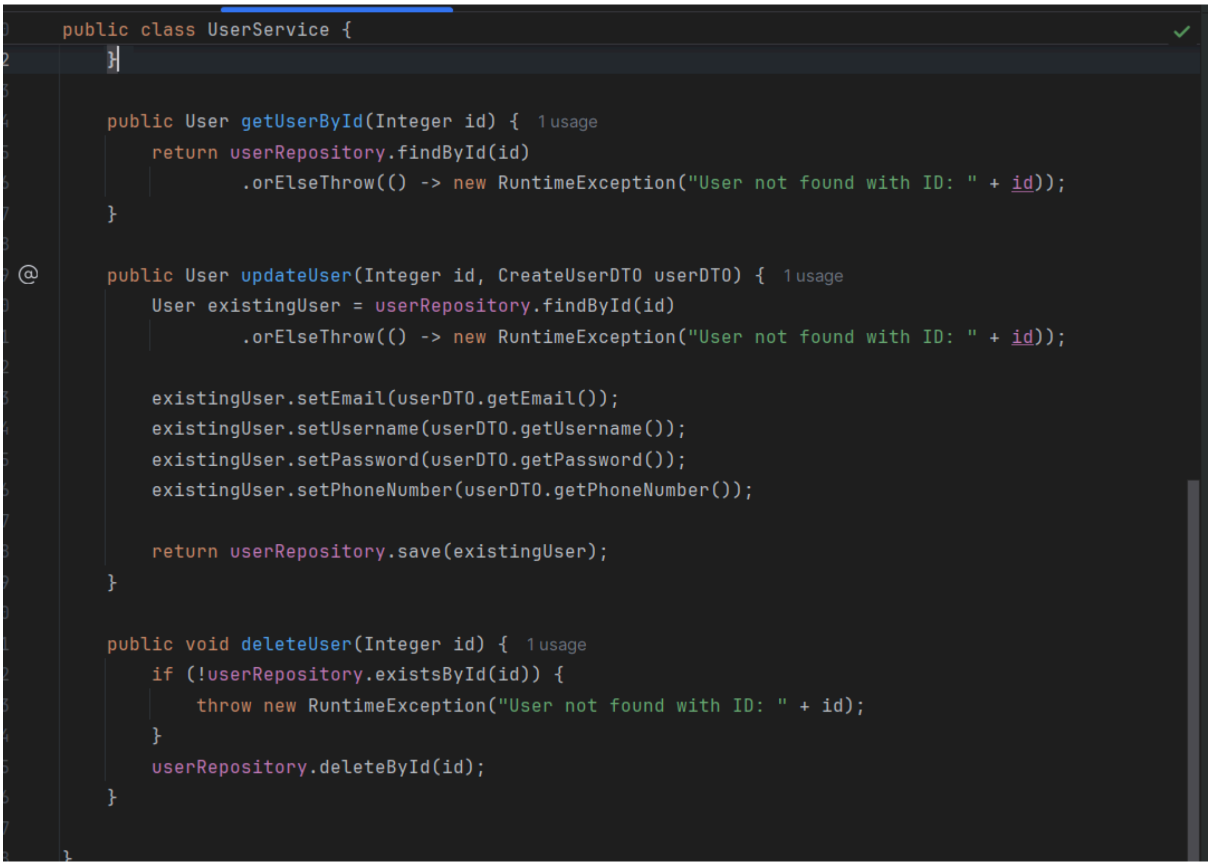


```
    }

    @GetMapping("/getAllUsers")  no usages
    public List<User> getUsers() {
        return this.userService.getUsers();
    }

    @PostMapping()  no usages
    public User createUser(@RequestBody CreateUserDTO userDTO) {
        return this.userService.createUser(userDTO);
    }

    @GetMapping("/{id}")  no usages
    public User getUserById(@PathVariable Integer id) {
        return userService.getUserById(id);
    }

    @PutMapping("/{id}")  no usages
    public User updateUser(@PathVariable Integer id, @RequestBody CreateUserDTO userDTO) {
        return userService.updateUser(id, userDTO);
    }

    @DeleteMapping("/{id}")  no usages
    public void deleteUser(@PathVariable Integer id) {
        userService.deleteUser(id);
    }
```

UserService:

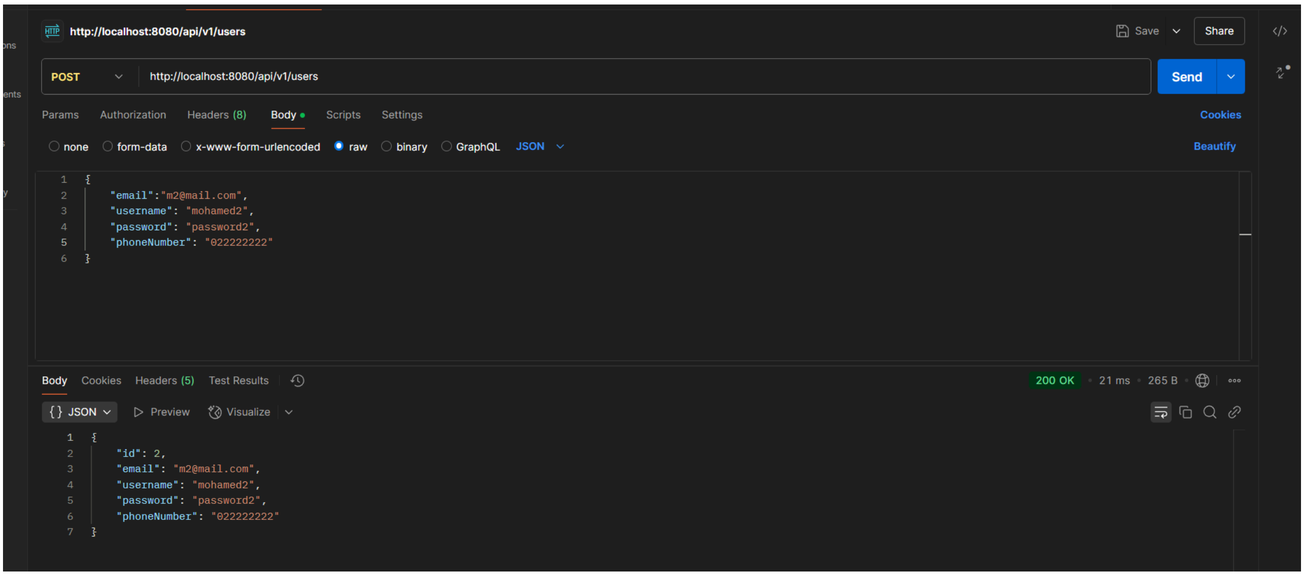


```
public class UserService {
    }

    public User createUser(CreateUserDTO userDTO) {  1 usage

        User user = new User(
                userDTO.getEmail(),
                userDTO.getUsername(),
                userDTO.getPassword(),
                userDTO.getPhoneNumber()
        );

        return this.userRepository.save(user);
    }

    public User getUserById(Integer id) {  1 usage
        return userRepository.findById(id)
                .orElseThrow(() -> new RuntimeException("User not found with ID: " + id));
    }

    public User updateUser(Integer id, CreateUserDTO userDTO) {  1 usage
        User existingUser = userRepository.findById(id)
                .orElseThrow(() -> new RuntimeException("User not found with ID: " + id));

        existingUser.setEmail(userDTO.getEmail());
        existingUser.setUsername(userDTO.getUsername());
        existingUser.setPassword(userDTO.getPassword());
        existingUser.setPhoneNumber(userDTO.getPhoneNumber());

        return userRepository.save(existingUser);
    }
```



```java
public class UserService {
    }

    public User getUserById(Integer id) {
        return userRepository.findById(id)
                .orElseThrow(() -> new RuntimeException("User not found with ID: " + id));
    }

    public User updateUser(Integer id, CreateUserDTO userDTO) {
        User existingUser = userRepository.findById(id)
                .orElseThrow(() -> new RuntimeException("User not found with ID: " + id));

        existingUser.setEmail(userDTO.getEmail());
        existingUser.setUsername(userDTO.getUsername());
        existingUser.setPassword(userDTO.getPassword());
        existingUser.setPhoneNumber(userDTO.getPhoneNumber());

        return userRepository.save(existingUser);
    }

    public void deleteUser(Integer id) {
        if (!userRepository.existsById(id)) {
            throw new RuntimeException("User not found with ID: " + id);
        }
        userRepository.deleteById(id);
    }

}
```

Create Method:



POST http://localhost:8080/api/v1/users

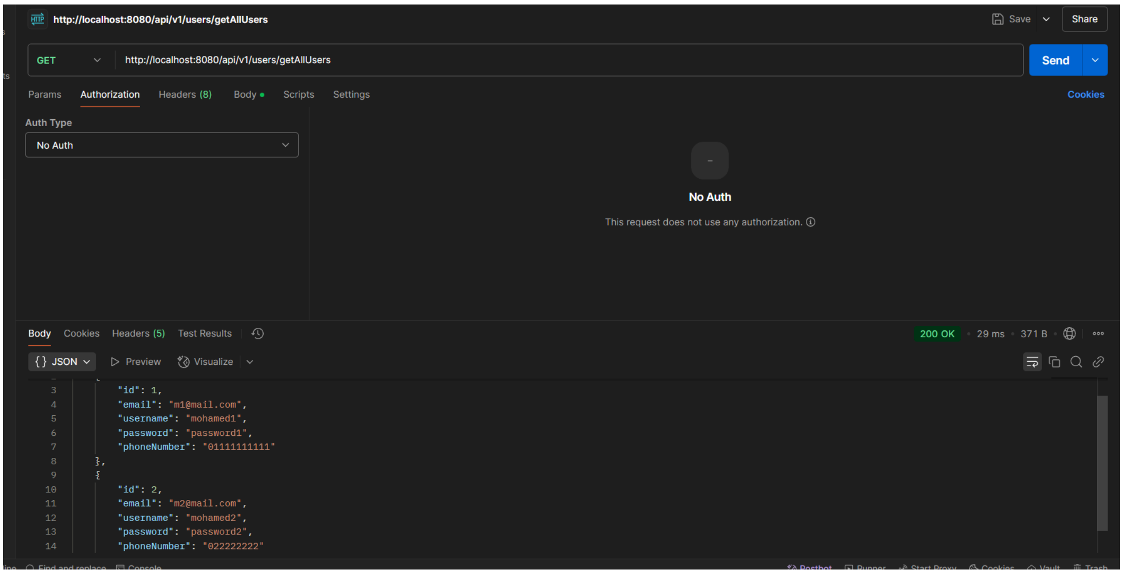
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```json
{
    "email":"m2@mail.com",
    "username": "mohamed2",
    "password": "password2",
    "phoneNumber": "022222222"
}
```

Body — 200 OK · 21 ms · 265 B — { } JSON

```json
{
    "id": 2,
    "email": "m2@mail.com",
    "username": "mohamed2",
    "password": "password2",
    "phoneNumber": "022222222"
}
```

Endpoint to get all users:



http://localhost:8080/api/v1/users/getAllUsers

GET http://localhost:8080/api/v1/users/getAllUsers

Auth Type: No Auth

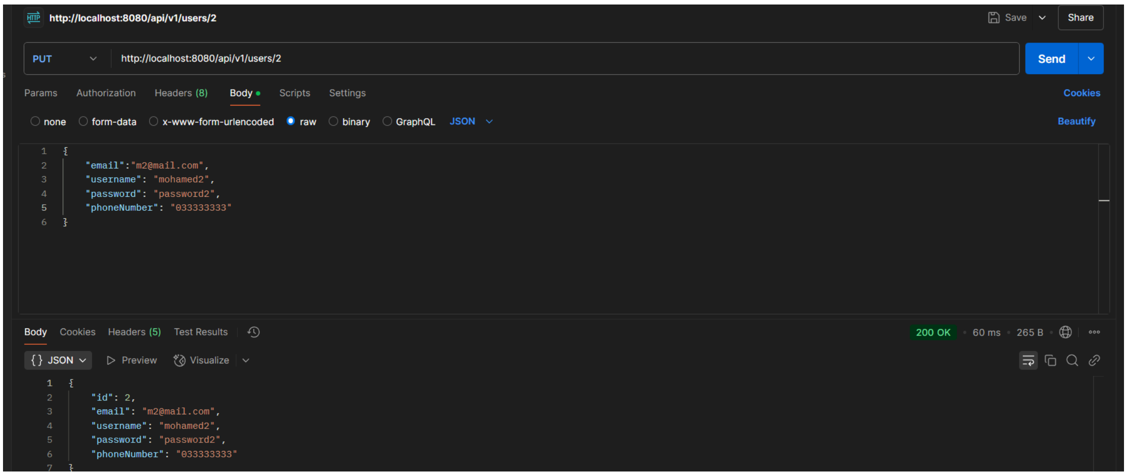
No Auth

This request does not use any authorization.

200 OK · 29 ms · 371 B

```
        "id": 1,
        "email": "m1@mail.com",
        "username": "mohamed1",
        "password": "password1",
        "phoneNumber": "01111111111"
    },
    {
        "id": 2,
        "email": "m2@mail.com",
        "username": "mohamed2",
        "password": "password2",
        "phoneNumber": "022222222"
```

UpdateUser:



http://localhost:8080/api/v1/users/2

PUT http://localhost:8080/api/v1/users/2

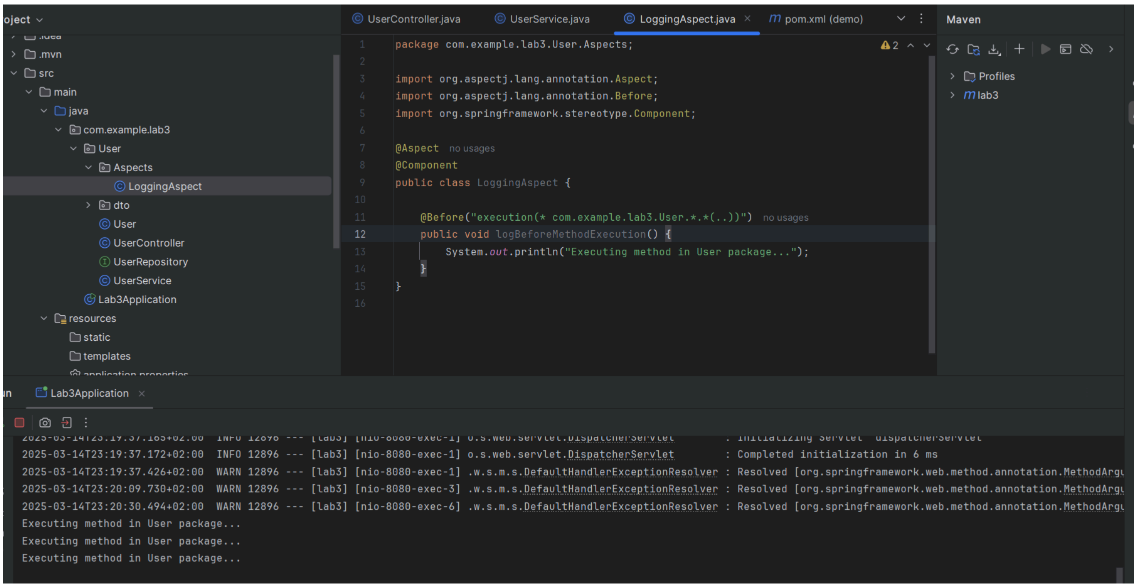
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "email":"m2@mail.com",
    "username": "mohamed2",
    "password": "password2",
    "phoneNumber": "033333333"
}
```

200 OK · 60 ms · 265 B

```
{
    "id": 2,
    "email": "m2@mail.com",
    "username": "mohamed2",
    "password": "password2",
    "phoneNumber": "033333333"
}
```

Aspect Code + example:



```java
package com.example.lab3.User.Aspects;

import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;
import org.springframework.stereotype.Component;

@Aspect
@Component
public class LoggingAspect {

    @Before("execution(* com.example.lab3.User.*.*(..))")
    public void logBeforeMethodExecution() {
        System.out.println("Executing method in User package...");
    }
}
```

```
2025-03-14T23:19:37.165+02:00  INFO 12896 --- [lab3] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-03-14T23:19:37.172+02:00  INFO 12896 --- [lab3] [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 6 ms
2025-03-14T23:19:37.426+02:00  WARN 12896 --- [lab3] [nio-8080-exec-1] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.method.annotation.MethodArgu
2025-03-14T23:20:09.730+02:00  WARN 12896 --- [lab3] [nio-8080-exec-3] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.method.annotation.MethodArgu
2025-03-14T23:20:30.494+02:00  WARN 12896 --- [lab3] [nio-8080-exec-6] .w.s.m.s.DefaultHandlerExceptionResolver : Resolved [org.springframework.web.method.annotation.MethodArgu
Executing method in User package...
Executing method in User package...
Executing method in User package...
```

Github Repo link:
https://github.com/MohamedMaghrabyyy/Aspect-and-Service-Oriented-Programming-Labs