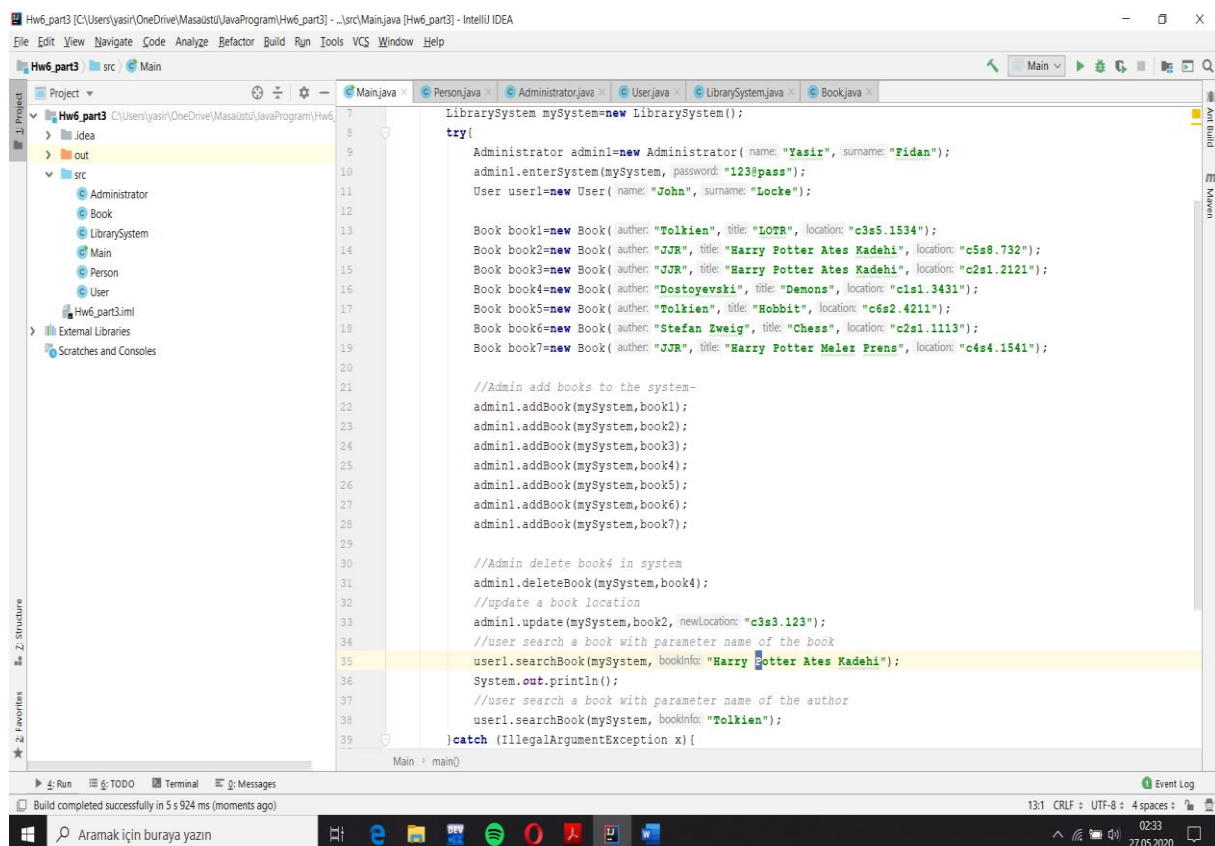


## Homework #6 – Part3 Report

In my design there is 5 class. Person class is a base class for Administrator and User class because, both admin and user can search book in system so I implement search book method in Person base class, so both of them can search a book. LibrarySystem is my system class it has a map like mentioned in pdf and a password for admin login. Admin has login, addbook, removebook and update methods. With login method if method argument and data field password of LibrarySystem matched then he/she enter the system and can use addbook, removebook and update methods freely. If admin try add/remove or update before entering system, program will throw an IllegalArgumentException. Admin can add book to system with addBook, remove book from system with removeBook and update some books locations in system with updateInformation methods. Books have Author name, book title and book location informations.



```
1  LibrarySystem mySystem=new LibrarySystem();
2
3  try{
4
5      Administrator admin1=new Administrator( name: "Yasir", surname: "Fidan");
6      admin1.enterSystem(mySystem, password: "123@pass");
7
8      User user1=new User( name: "John", surname: "Locke");
9
10
11
12
13      Book book1=new Book( author: "Tolkien", title: "LOTR", location: "c3s5.1534");
14      Book book2=new Book( author: "J.R.R.", title: "Harry Potter Ates Kadehi", location: "c5s8.732");
15      Book book3=new Book( author: "J.R.R.", title: "Harry Potter Ates Kadehi", location: "c2s1.2121");
16      Book book4=new Book( author: "Dostoyevski", title: "Demons", location: "c1s1.3431");
17      Book book5=new Book( author: "Tolkien", title: "Hobbit", location: "c6s2.4211");
18      Book book6=new Book( author: "Stefan Zweig", title: "Chess", location: "c2s1.1113");
19      Book book7=new Book( author: "J.R.R.", title: "Harry Potter Melez Prens", location: "c4s4.1541");
20
21
22      //Admin add books to the system-
23      admin1.addBook(mySystem,book1);
24      admin1.addBook(mySystem,book2);
25      admin1.addBook(mySystem,book3);
26      admin1.addBook(mySystem,book4);
27      admin1.addBook(mySystem,book5);
28      admin1.addBook(mySystem,book6);
29      admin1.addBook(mySystem,book7);
30
31      //Admin delete book4 in system
32      admin1.deleteBook(mySystem,book4);
33
34      //update a book location
35      admin1.update(mySystem,book2, newLocation: "c3s3.123");
36
37      //user search a book with parameter name of the book
38      user1.searchBook(mySystem, bookinfo: "Harry Potter Ates Kadehi");
39      System.out.println();
40
41      //user search a book with parameter name of the author
42      user1.searchBook(mySystem, bookinfo: "Tolkien");
43
44      }catch (IllegalArgumentException x){
45
46      }
```

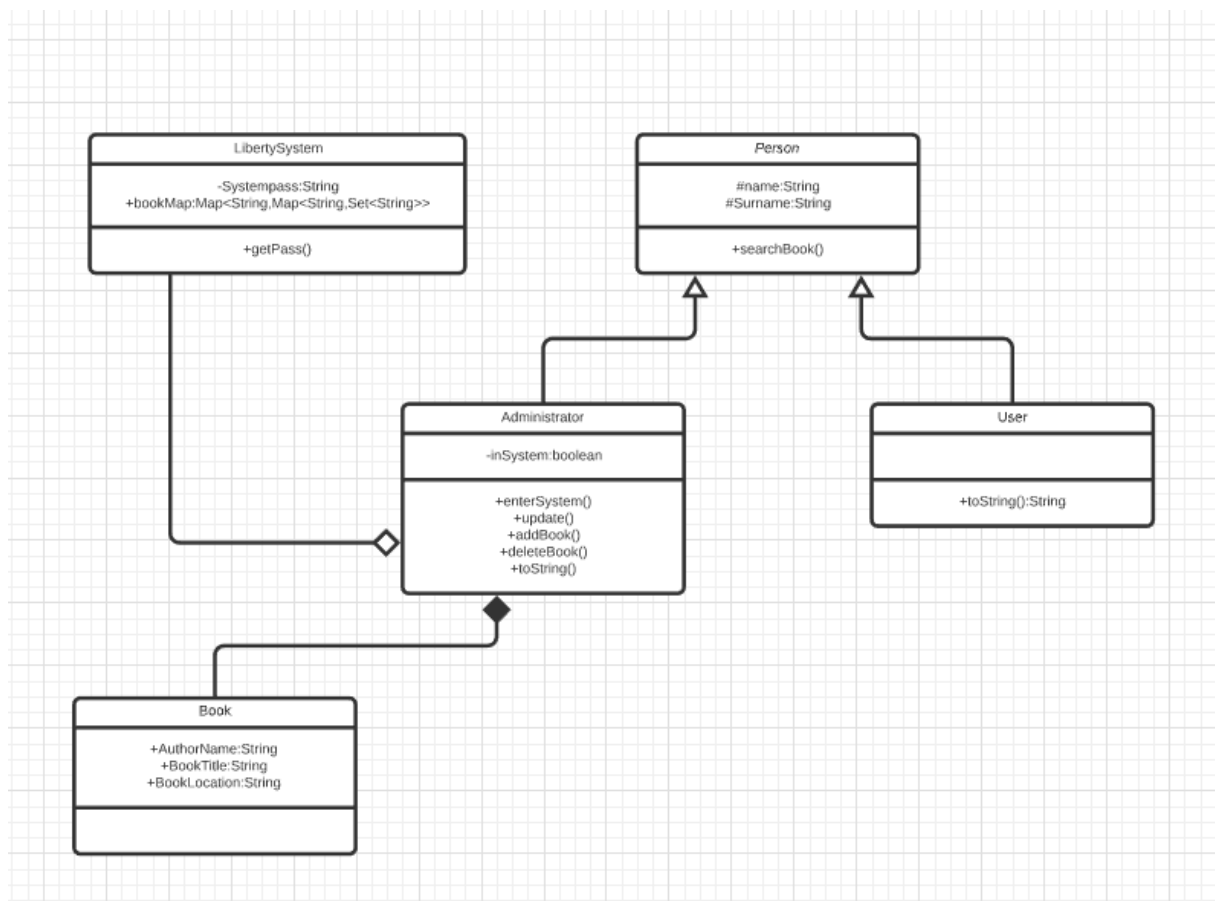
This is my main ss. In line 9 I create an admin after that I use enterSystem method and my admin enter system and create a user in line 11. Then in line 13 and 19 I create some books and in line 22 and 28 I add this books into the system. In line 31 I remove book4 from the system now there is no book4 in my system. In line 33 with admin update method I update book2 location as c3s3.123. After that I call searchbook 2 times, first search use a book name argument and second search method use a author name. Both of them work as mentioned in pdf.

```
Run: Main x
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" -Didea.launcher.port=
Author name: JJR
Locations: [c2s1.2121, c3s3.123]

Author name: Tolkien
Books: [LOTR, Hobbit]
Enter Book name for see book locations:
LOTR
Locations:
c3s5.1534

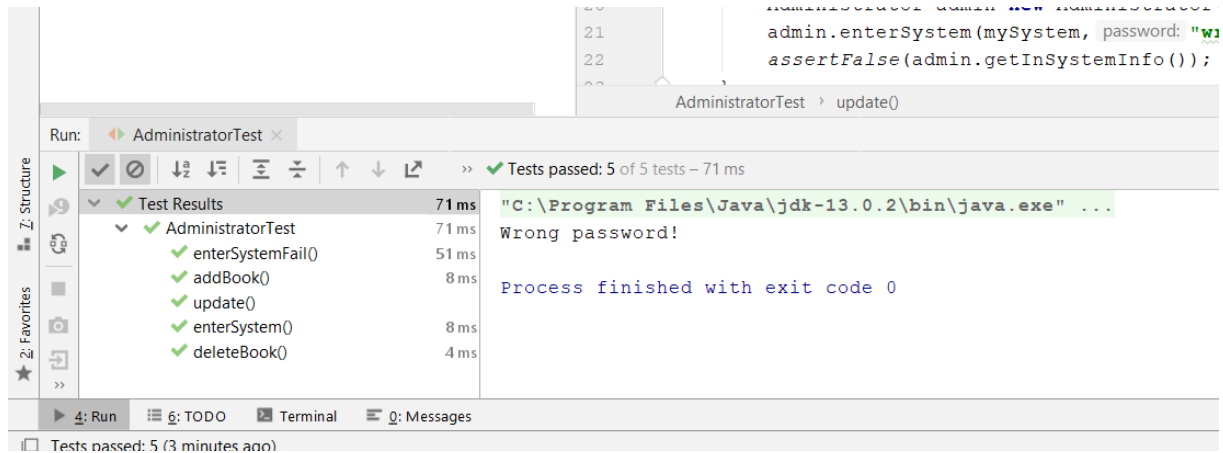
Process finished with exit code 0
```

This is my output. First search use book name as argument so it prints Author name and All locations of that book. Then my second search method call use author name as argument, so it prints author name and all books that author have, after that it ask user a input then its prints input book locations in system(in this example I enter LOTR and its location c3s5.1534 so it prints it).



## Test cases:

You can find my test cases under Test library, I use Junit5 for testing.



Test Case Id	Test Scenario	Expected Results	Actual Results	Pass/Fail
1	Enter system with correct password	Admin.inSystem=true	As expected	Pass
2	Enter system with wrong password	Admin.inSystem=false	As expected	Pass
3	Update a book location	Book new location =c1s1.111	As expected	Pass
4	Add a book to the system	Book("Tolkien","LOTR","c3s5 1534") added to system	As expected	Pass
5	Remove a book from system	System has no book	As expected	Pass

**Muhammed Yasir Fidan**

**161044056**