Final Project

Java EE 2021

Object-Oriented Programming (Java EE)

|  |  |
| --- | --- |
| Name (CN) | Student Number |
| 马成 | 42054022 |
| 赵子熠 | 42054004 |
| 刘致远 | 42054008 |
| 王满 | 42054058 |



**CATALOGUE**

[Part 1: Introducation 3](#_Toc76743426)

[Part 2: Design 4](#_Toc76743427)

[2.1 Data Design 4](#_Toc76743434)

[2.2 UI Design 5](#_Toc76743435)

[2.3 User Event Design 5](#_Toc76743437)

[2.4 MVC Design 7](#_Toc76743439)

[Part 3: Implementation 8](#_Toc76743440)

[3.1 User Sign-up 8](#_Toc76743441)

[3.2 USER register 8](#_Toc76743442)

[3.3 ADD NEW BoOK 9](#_Toc76743443)

[3.4 SEARCH FOR BOOKS 9](#_Toc76743445)

[Part 4: Demo 11](#_Toc76743446)

[Part 5: Conclusion 14](#_Toc76743447)

# Part 1: Introducation

In an age of ever-evolving information technology, we can say that we are all too familiar with the Internet, which has become a globalized Internet network. As the number of users continues to increase, its scale is rapidly expanding and its domain is becoming diversified. The Internet has become an indispensable part of many people's daily lives, especially online shopping, which attracts numerous people with its long distance, convenience and speed. Therefore, online shopping on the Internet is becoming more and more practical today. And the prospects for development are very good.

The 21st century is the era of the Internet, the era of information, time is very valuable, people do not have time to go to the bookstore for various reasons, and do not know which bookstore has the books they need, and at the same time those traditional bookstore operators have no good way to let people know that I have the books that customers need. This hindrance to the exchange of information between buyers and sellers has been the driving force behind the development of 'online bookstore' websites. The establishment of an online bookstore website can be a better solution to this problem, as the online bookstore is more convenient than the offline bookstore, and you can buy your favorite books without leaving your home. Administrators through the background management, you can add the classification of books, and add books to the front page of the home page for users to choose; users to register, after successful registration to login, users can view the recently uploaded books, you can search for books; administrators through the background can add books or delete books, then the user side can also see the administrator added books or deleted books.

# Part 2: Design

## This system mainly faces two types of roles for users and administrators for functional positioning, the basic positioning of functions are as follows.

## 1. For customers of the online bookstore.

## (1) New users can be registered and logged into their accounts.

## (2) The function of viewing books: customers can view the type, quantity, price and all the books in the online bookstore by logging into the webpage.

## 2. Administrator functions of the system.

## (1) Book management: administrators can add and delete books from the system of the online book city, including information such as book name, author, price, picture and its description.

## 2.1 Data Design

In this set of systems, the use of java object-oriented thinking, encapsulated object data, such as books, which itself has the name, author, price, picture, description and other objects, we can encapsulate it into an object inside, the properties set private, external set, get properties to provide external access, and finally these objects through the servlet built-in object will be returned to the data The data will be returned to the page through the servlet.

## 2.2 UI Design

## This system does not use any framework, if I could use the UI, I would probably learn jquery first, then use the more popular bootstrap, vue or the more powerful note.js.

## 2.3 User Event Design

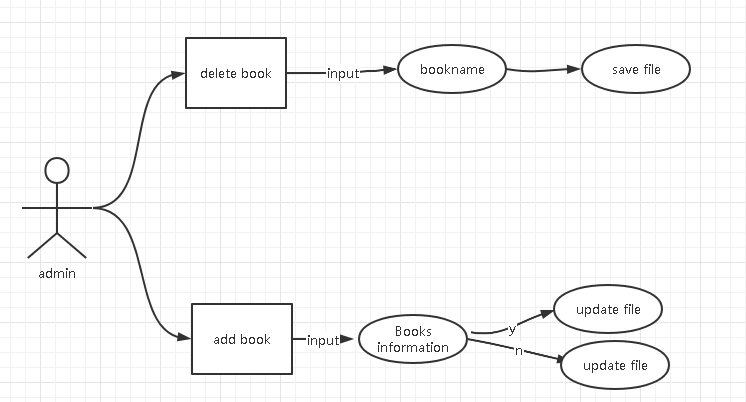
**1、The description of how to use it：**

(1) System users can log in to the system after registering their system account, and enter the system to find books by book name; users of the system who are not registered are called visitors.

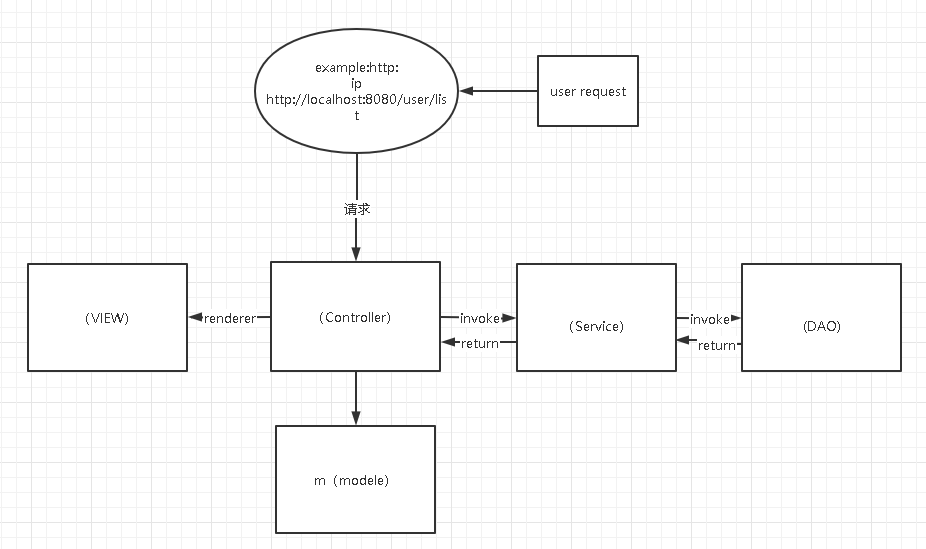
The use case diagram for users is shown in Figure 1.

### 

As shown in Figure 1 User use case diagram

Description of the administrator's business flow: The administrator has access to the backend of the system and can manage the system's book information, including adding books, deleting books, etc. The use case diagram for the administrator is shown in Figure 2. Figure 2 Administrator use case diagram

## 2.4 MVC Design



As shown above, we go through the mall system, starting with a user-initiated request, such as the http request in the example above, through the servlet request, then to the file to read the data, the data returned layer by layer, and finally to the View page rendering, here we will not describe one by one, here we repeat the purpose is to deepen the understanding of the MVC concept.

# Part 3: Implementation

## 3.1 User Sign-up

1. String name = req.getParameter("inputName");
2. String pwd=req.getParameter("inputPassword");
3. Map<String,String> map=new HashMap<>();
4. File file=new File("C:\\2.txt");
5. Path path=file.toPath();
6. try(Stream<String>lines= Files.lines(path)) {
7. lines.forEach(line -> {
8. String[] items = line.split(" ");
9. map.put(items[0], items[1]);
10. });
11. }
12. if(map.containsKey(name) && map.get(name).equals(pwd)){
13. req.getSession().setAttribute("name", name);
14. req.getRequestDispatcher("index").forward(req, resp);
15. }else{
16. resp.sendRedirect("Error.html");
17. }

As the above code: user login module, through the user input parameters inputName and inputPassword, iterate through the 2.txt text under the C disk folder, loop to match the user inputName and inputPassword, if matched, call req.getSession(). setAttribute("name", name); put the object into the sesion, finally jump to index, login success, if no match, then jump to Error.html, login failure.

## 3.2 USER register

1. String user=req.getParameter("inputName");
2. String pwd=req.getParameter("inputPassword");
3. FileWriter file=new FileWriter("C:\\2.txt",true);
4. file.write("\r\n"+user+" "+pwd);
5. file.flush();
6. file.close();
7. resp.sendRedirect("rig-success.html");

As in the code above: register the user, get the user input parameters inputName and inputPassword through req.getParameter, by calling the jkd tool class FileWriter file=new FileWriter("C:\\2.txt",true);. Save the user name and password inside the file and eventually jump to the page rig-success.html page.

## 3.3 ADD NEW BoOK

1. String part2 =request.getParameter("book");
2. Part part = request.getPart("IMG");
3. Part part1 = request.getPart("TXT");
4. String p="C://"+part2+".jpg";
5. part.write(p);
6. part1.write("C:"+part2+".txt");
7. FileWriter b=new FileWriter("C:\\Game.txt",true);
8. b.write("\r\n"+part2);
9. b.flush();
10. b.close();
11. response.sendRedirect("ok.html");

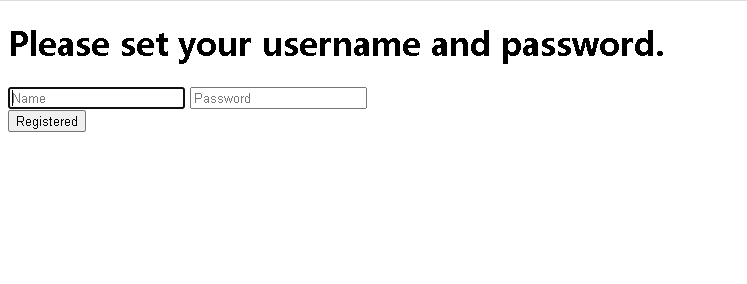
## As in the code above: add a book, call /upload with the user input parameters book etc. and write the book information entered by the user to the Game.txt text in the C drive folder.

## 3.4 SEARCH FOR BOOKS

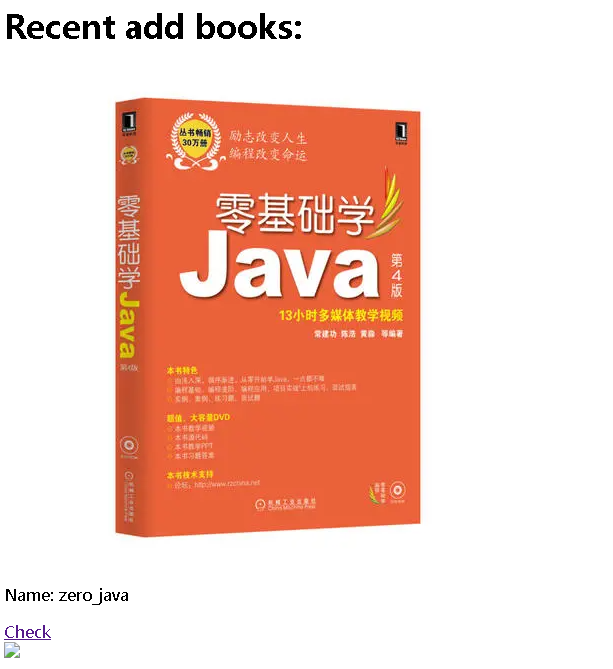
1. String name = req.getParameter("inputName");
2. List<String> list=new ArrayList<>();
3. File file=new File("C:\\Game.txt");
4. Path path=file.toPath();
5. try(Stream<String> lines= Files.lines(path)) {
6. lines.forEach(line -> {
7. String[] items = line.split(" ");
8. list.add(items[0]);
9. });
10. }
11. if(list.contains(name)){
12. req.getSession().setAttribute("name", name);
13. req.getRequestDispatcher("result.jsp").forward(req, resp);
14. }else{
15. resp.sendRedirect("result2.html");
16. }

As above code: search for books, through the user input parameters inputName, iterate through the C disk folder under the Game.txt text, loop to match the user inputName, if matched, call req.getSession().setAttribute("name", name); the object into seesion, and finally jump to result.jsp, if no match, then jump to result2.html.

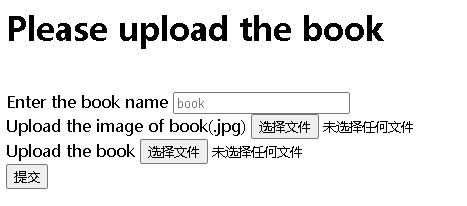
# Part 4: Demo

1. Users enter their username and password and click on the Registered button to register as a user.

2. Users click on the View Recently Uploaded Books button to view information about recently uploaded books.



**1. Book upload on the administrator's side.**



**2. Delete book information from the administrator side.**



# Part 5: Conclusion

The course design is an important part of training students to apply the knowledge they have learnt and to find, propose, analyses and solve practical problems, and is a concrete training and investigation process for students' practical working ability. In this course design, we not only further consolidate the knowledge we have learnt about Java Web programming, but also learn what we have not learnt from books and classes. In this course design, we also encountered a lot of difficulties, not deep enough to grasp the knowledge points, or some running errors have not seen, but through querying Baidu, constantly trying new methods, etc., but eventually overcome. The front-end and back-end of the online bookstore designed for this course have many shortcomings, and there are still many functions that have not been implemented, so we still need to continue learning and then improve.

Through the practice of this course design, we also realized more deeply that only theoretical knowledge is not enough, only the application of theoretical knowledge to practice, the full combination of theory and practice, is more conducive to their own knowledge of the mastery, higher than their own practical ability.