

COM6519-002

180233692

180221840

180139659

180237379



## Group Assessed Work Coversheet

**Assessment Code: COM6519**

**Description: Stage 2: Product Development**

**Staff Member Responsible:  
Simons, Dr Tony**

**Due Date: 07-05-2019 15:00:00**

**Student Registration Numbers:**



180237379



180139659



180221840



180233692

**Assessment Code:**



COM6519-002

☒ I certify that the attached is all my own work, except where specifically stated and confirm that I have read and understood the University's rules relating to plagiarism.

I understand that the Department reserves the right to run spot checks on all coursework using plagiarism software.

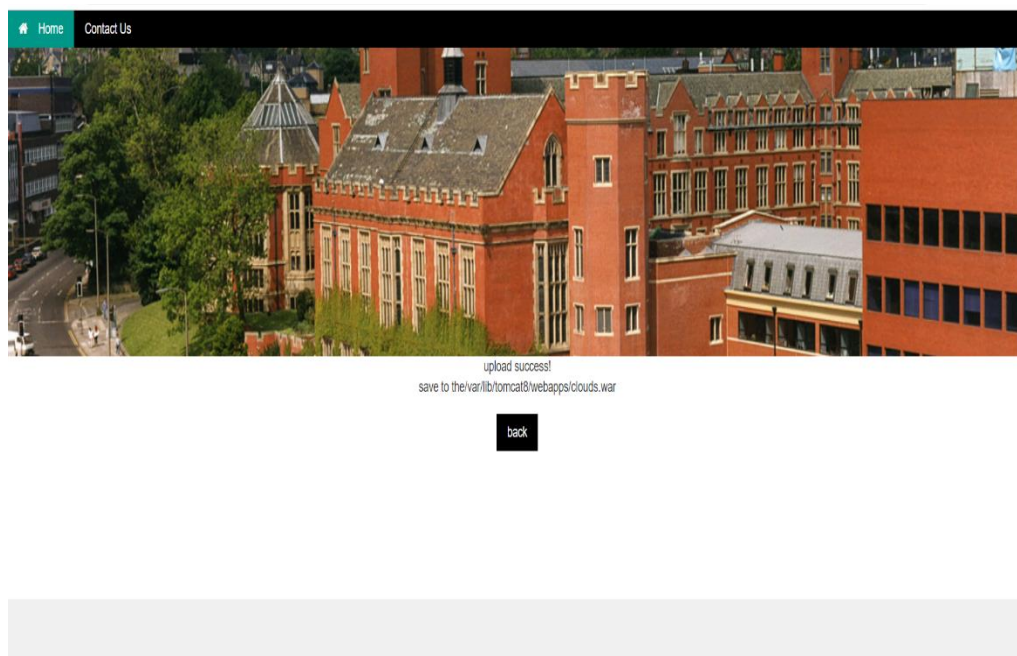
# Product Development

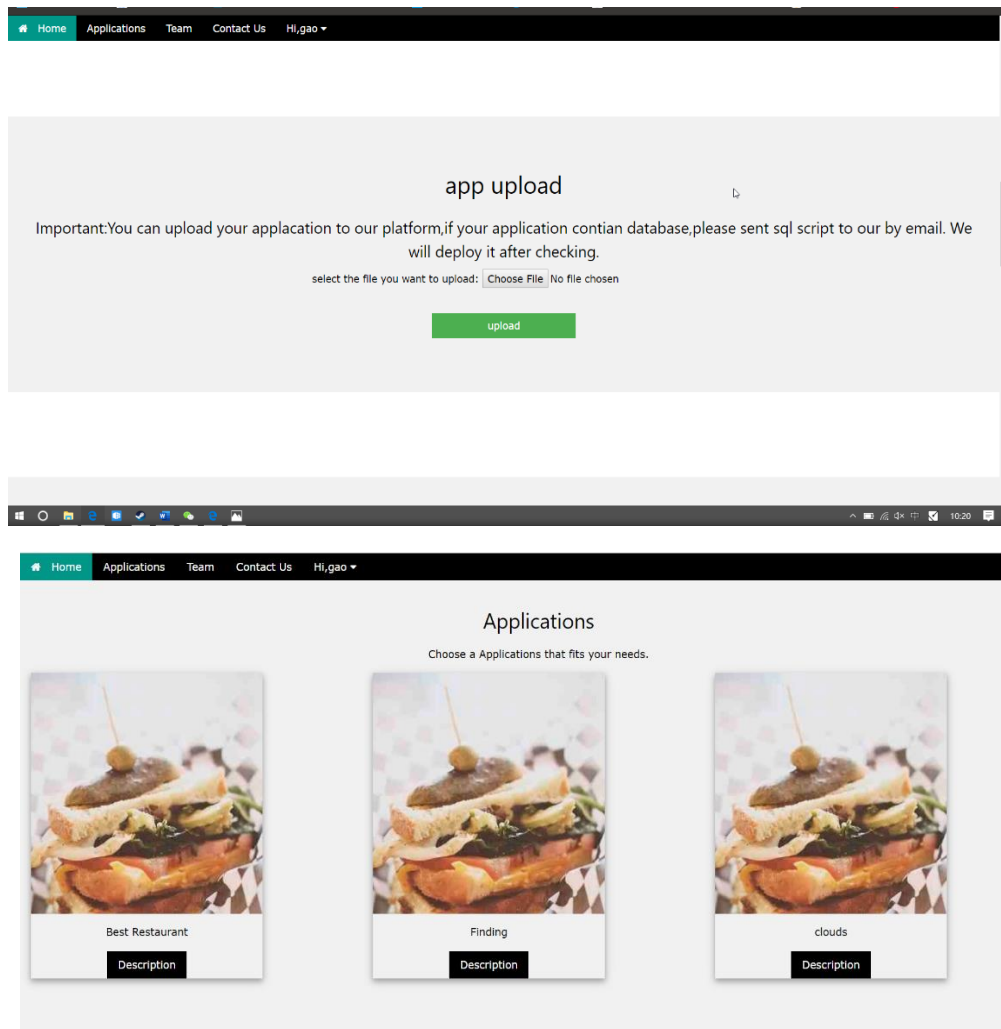
## 1. Business

Based on the application server, our team has established a platform which hosts two micro-services and two applications focusing on offering academic support and restaurant rankings for students in Sheffield, making their daily routine more convenient. The whole work runs on the OpenNebula where two applications are deployed. These two applications respectively aim at finding classrooms for lectures and discovering decent restaurants.

The first application is a restaurant guide app. The customer can browse all the restaurants in Sheffield. Restaurants could be ranked by the score of cost and quality, or with student recommendations. Everyone can score and comment on the restaurant, and all comments can be viewed. This information can provide a reference for users when choosing restaurants. The next application is lecture theatre app. Students can use this app to find the lecture theatre they want to go. Students need enter theatre name and building name, after, it will show mark and a normal access route of target building, as well as the theatre's location in the teaching building. If the normal access route is blocked by building work, this app would mark it and it would support another access route to there.

We are now hosting a cloud ecosystem where app providers can upload their own application on the website after log-in. Following is the related screenshots.





**Figure 1.1: Cloud Ecosystem - Application Uploading Process**

At the very early stage we have already reached the agreement on the development environment, following is the details of the servers and software we use.

<b>Manifest-Version</b>	<b>1.0</b>
<b>Ant-Version</b>	<b>Apache Ant 1.9.6</b>
<b>Created-By:</b>	<b>1.7.0_80-b15 (Oracle Corporation)</b>
<b>Specification-Title</b>	<b>Apache Tomcat</b>
<b>Specification-Version</b>	<b>8.0</b>
<b>Specification-Vendor</b>	<b>Apache Software Foundation</b>

<b>Implementation-Title</b>	<b>Apache Tomcat</b>
<b>Implementation-Version</b>	<b>8.0.36</b>
<b>Implementation-Vendor</b>	<b>Apache Software Foundation</b>
<b>X-Compile-Source-JDK</b>	<b>1.7</b>
<b>X-Compile-Target-JDK</b>	<b>1.7</b>

**Table 1.1: tomcat standard**

<b>Manifest-Version</b>	<b>1.0</b>
<b>Ant-Version</b>	<b>Apache Ant 1.9.6</b>
<b>Created-By:</b>	<b>1.7.0_80-b15 (Oracle Corporation)</b>
<b>X-Compile-Source-JDK</b>	<b>1.7</b>
<b>X-Compile-Target-JDK</b>	<b>1.7</b>
<b>Name</b>	<b>javax/servlet/jsp/</b>
<b>Specification-Title</b>	<b>Java API for JavaServer Pages</b>
<b>Specification-Version</b>	<b>2.3</b>
<b>Specification-Vendor</b>	<b>Sun Microsystems, Inc.</b>
<b>Implementation-Title</b>	<b>javax.servlet.jsp</b>
<b>Implementation-Version</b>	<b>2.3.FR</b>
<b>Implementation-Vendor</b>	<b>Apache Software Foundation</b>

**Table 1.2: JSP standard**

<b>Manifest-Version</b>	<b>1.0</b>
<b>Ant-Version</b>	<b>Apache Ant 1.9.6</b>
<b>Created-By:</b>	<b>1.7.0_80-b15 (Oracle Corporation)</b>
<b>X-Compile-Source-JDK</b>	<b>1.7</b>
<b>X-Compile-Target-JDK</b>	<b>1.7</b>
<b>Name</b>	<b>javax/servlet/</b>
<b>Specification-Title</b>	<b>Java API for Servlets</b>
<b>Specification-Version</b>	<b>3.1</b>
<b>Specification-Vendor</b>	<b>Sun Microsystems, Inc.</b>
<b>Implementation-Title</b>	<b>javax.servlet</b>
<b>Implementation-Version</b>	<b>3.1.FR</b>
<b>Implementation-Vendor</b>	<b>Implementation-Vendor: Apache Software Foundation</b>

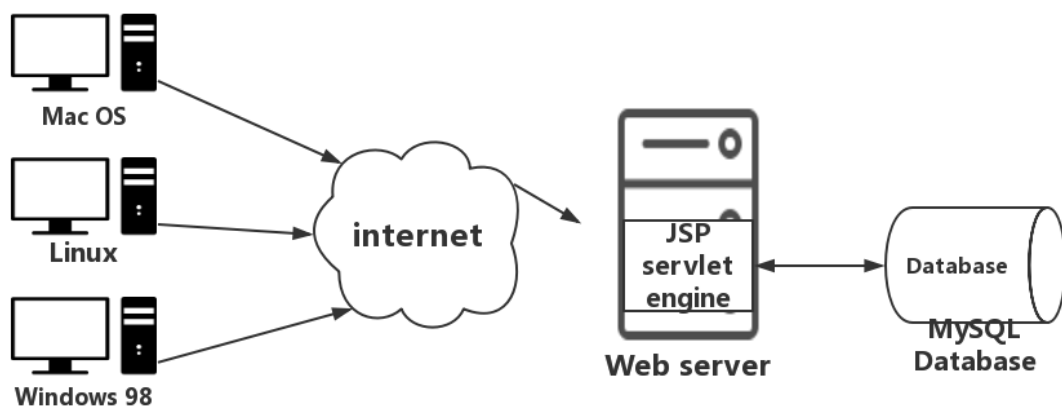
**Table 1.3: servlet standard**

Within the team, we set up several procedures we agree to follow when it comes to cooperation, such as discuss the codes weekly and check if the functions presented in the right way. We keep our codes as neat as possible and add explanations to ensure bug can be easily solved. We used to have disagreements on different aspects, such as what script language we need to use, but we resolved the disagreements by comparing the advantages and disadvantages of several candidate languages which will be discussed in the design section.

## 2. Design

Among all the possible method to realise the product, we chose Java servlet and JSP as our main method.

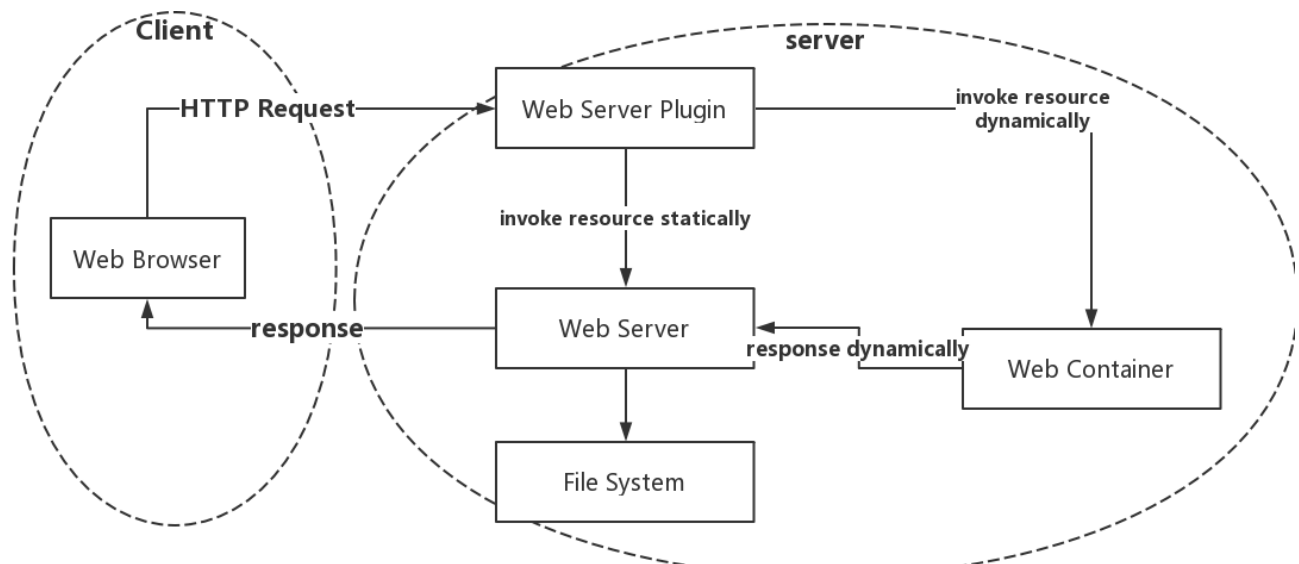
As a script language, JSP (Java Server Pages) technology basis on java language. In order to support HTTP applications, JSP web page provides an interface to the Java library of the server. JSP acquires users' input data from web page list, invokes database and other sources of data, then creates web page dynamically. Except delivering control information and share- information between a different web page, JSP tags are able to access the database, record the selection information of users.



**Figure 2.1: Typical Web Server supporting JSP**

JSP program's functions are similar to CGI's, but it has better performance because JSP can embed elements into dynamic HTML web page directly. The server invokes compiled JSP files, thus, compiling in JSP saves time.

XHTML is part of the family of XML markup languages, it is a transition from HTML to XML whose syntax is more rigorous than HTML. An XHTML editor can only edit a static web page, which has no background database and is non-interactive.



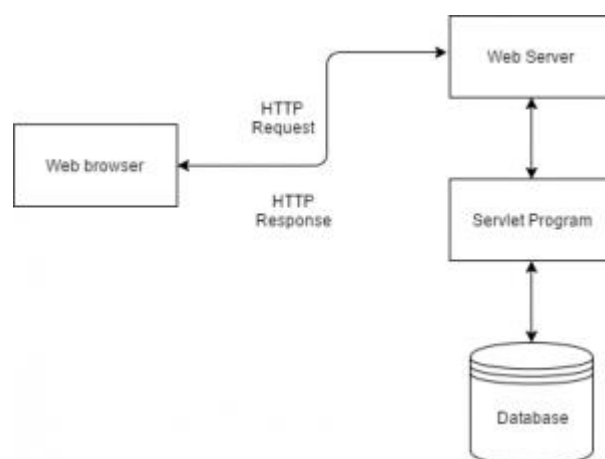
**Figure 2.2: Web page access process**

Figure 2.2 displays the access process of static web page and dynamic web page.

After the comparison, we concluded that XHTML can't fill our requirement of developing a cloud platform and decided to edit the JSP web page.

Running on the web server or application server, Servlet is a Java program used to receive and process the request from the web server and produce the response which will be sent back to the server.

The following diagram shows how the Servlet coordinates with other components:

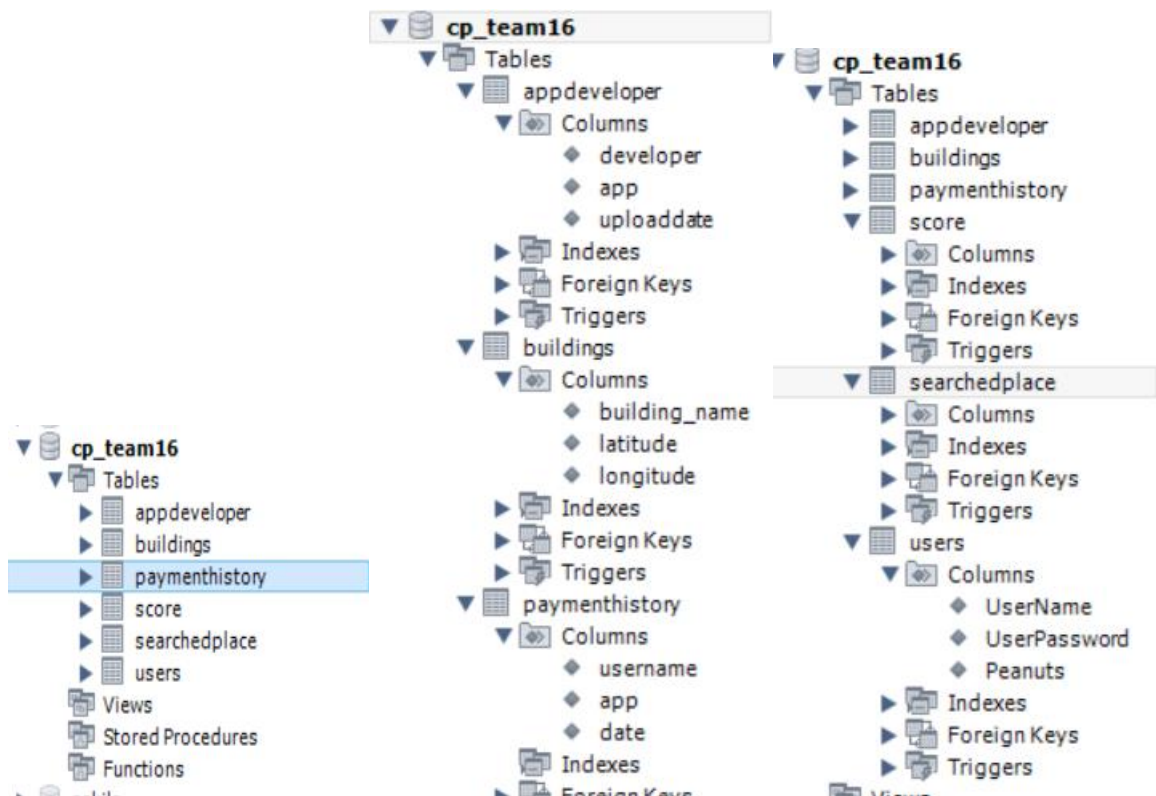


Although there are similar functions between the Servlet and the CGI, we still can not ignore the differences between them.

In conclusion, the advantages of Servlet are as follows:

1. Compare to CGI, the servlet is able to generate dynamic projects in a more convenient way and at the same time reduce the running time,
2. servlet technology keeps all the features of JAVA while making the functions scalable,

3. servlet make communications within different servlet and servers possible and creates easy portability across different web servers,
4. servlet is quite robust and reliable since it is deployed on the server and uses its own protocol and API.



**Figure 2.3: Web page access process**

Pictures above are the basic structure of our databases, all the tables about the micro-services and the apps are listed.

### **Risk analysis:**

We preliminary identify risks of our website: Computers doesn't work, Software bugs: insufficient testing/poor code, Data loss or file loss, member illness, and we analyze the impact of these risks on the project.

<b>risk</b>	<b>Probability</b>	<b>impact</b>	<b>solution</b>
Computers don't work	Low	high	Using the library's computer to continue working



Software bugs: insufficient testing/poor code	high	high	Improve efficiency of coding, do more test
Data loss or file loss	low	high	Prepare the copies of data and file, and upload to Google drive or github
member illness	high	low	Make a timetable in advance, reasonable assignment of tasks to each day

**Table 2.1 Risk analysis**

## Platforms UML diagram

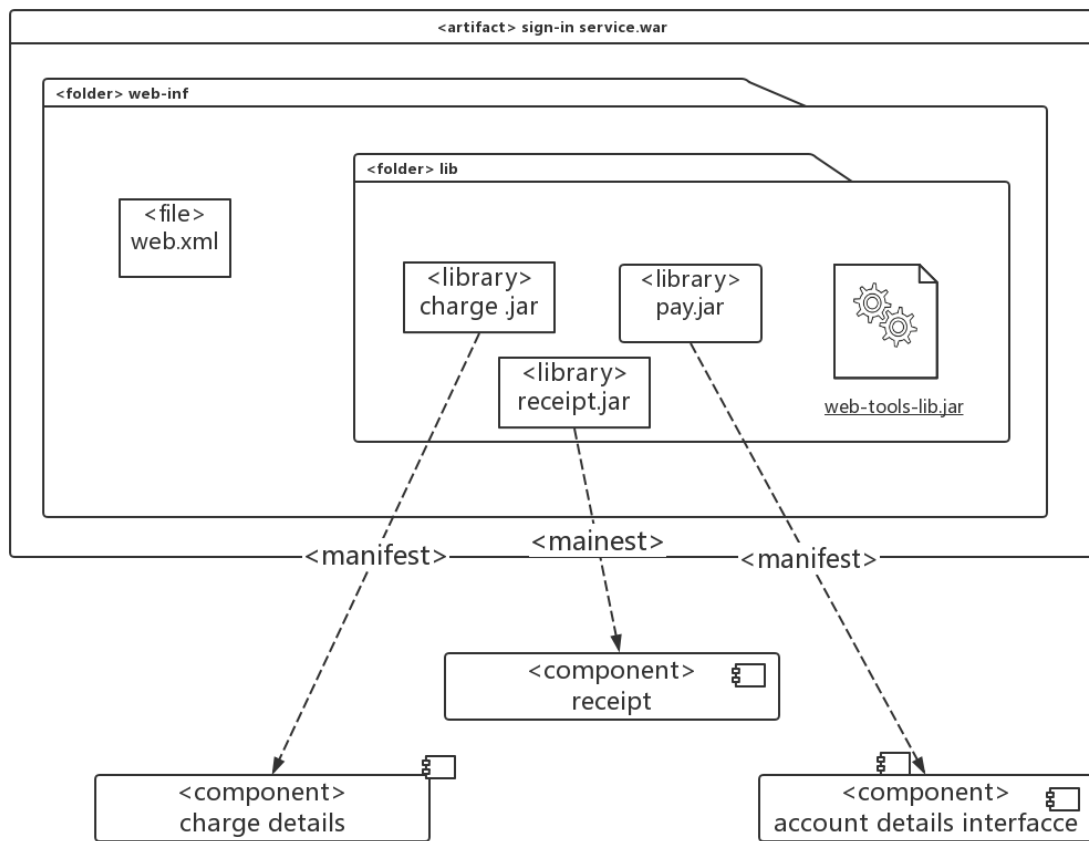
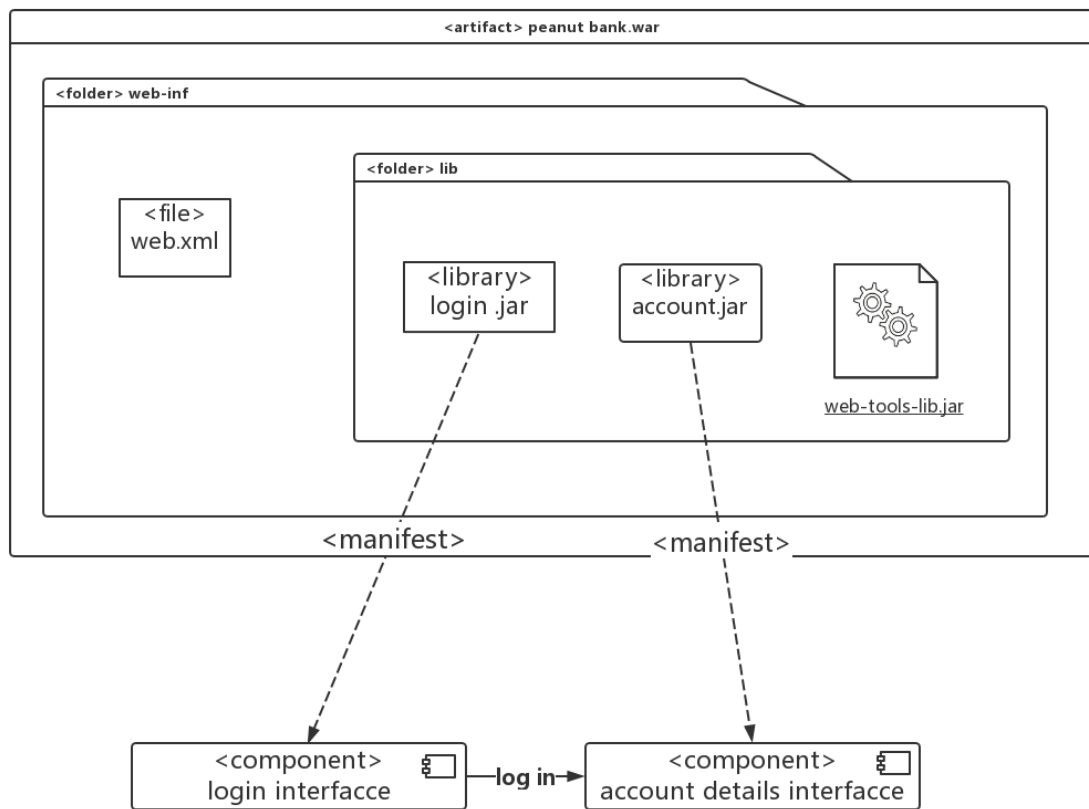
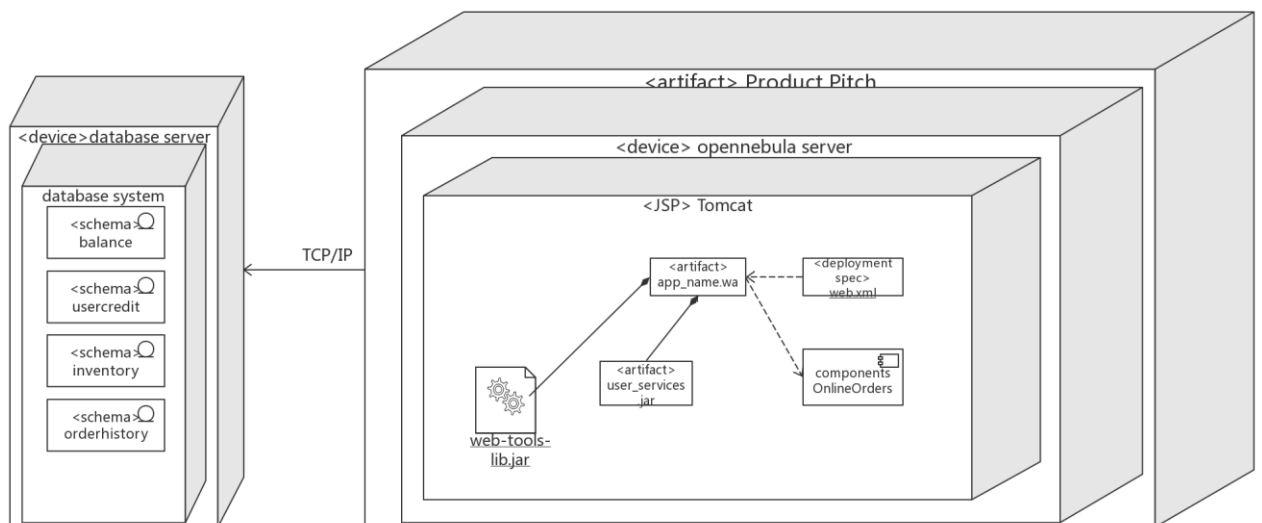


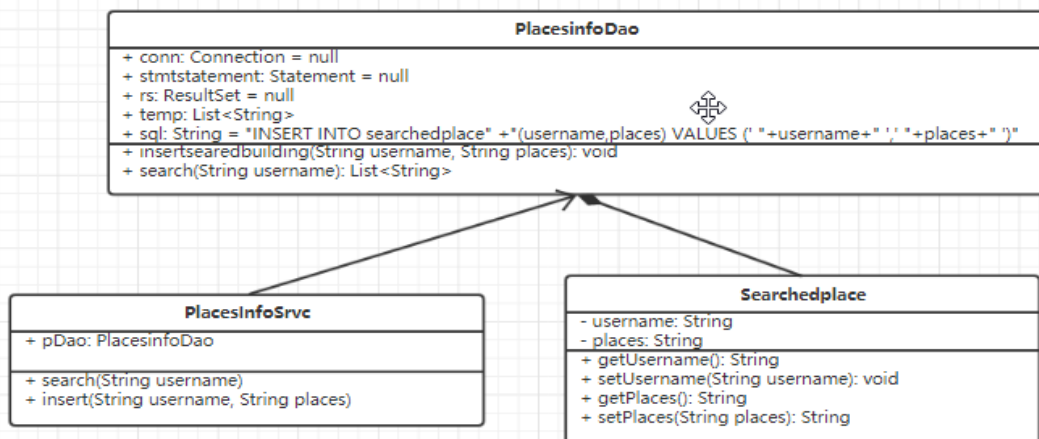
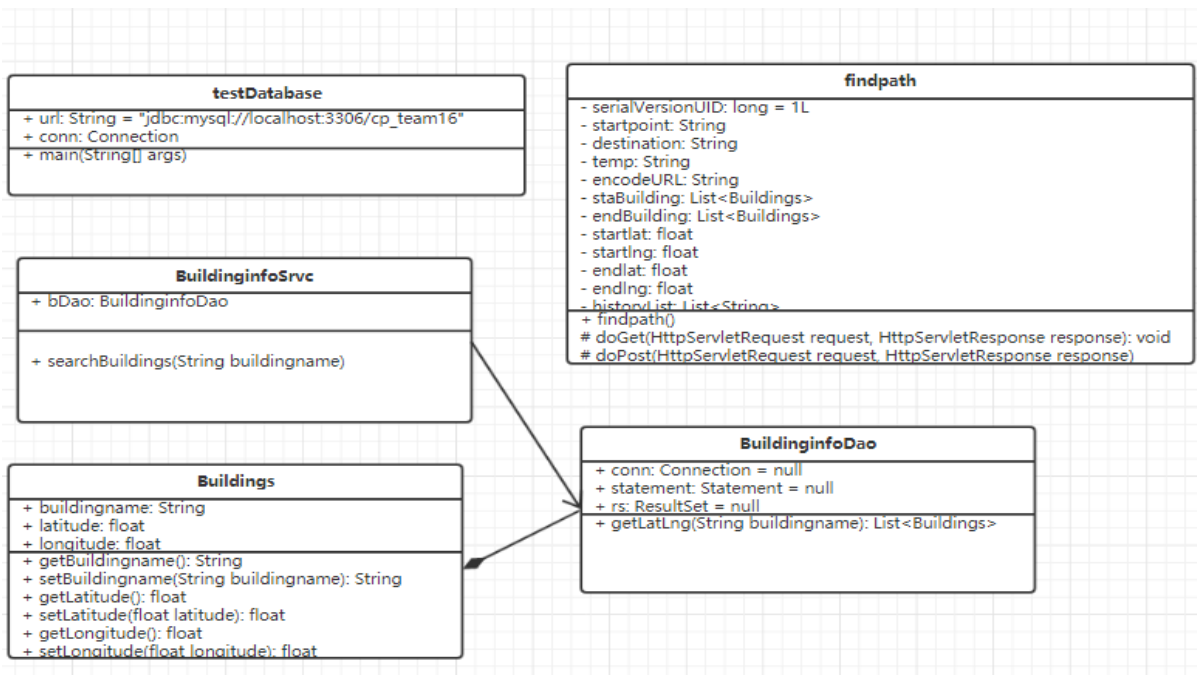
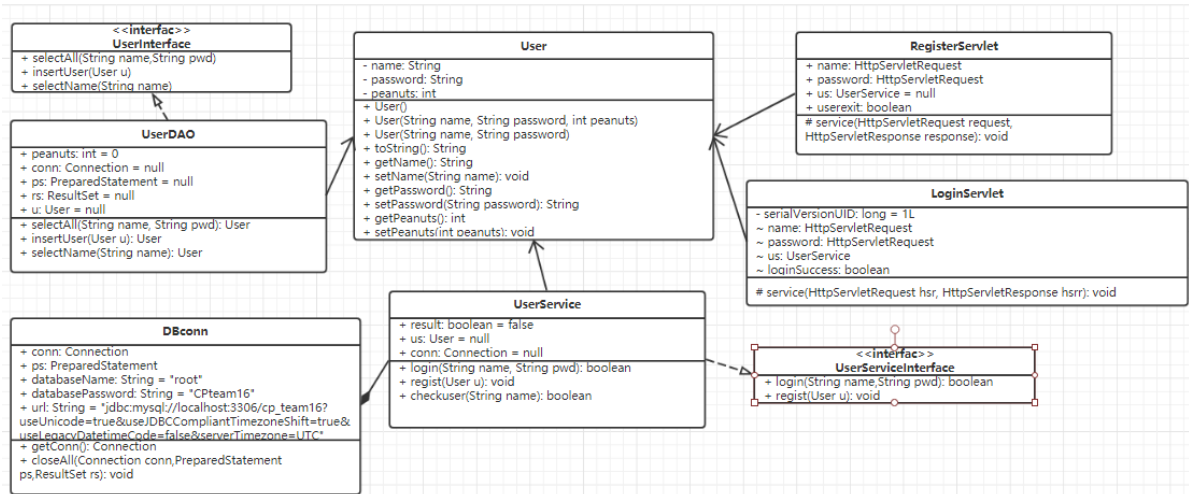
Figure 2.3: UML diagram of single sign-in service

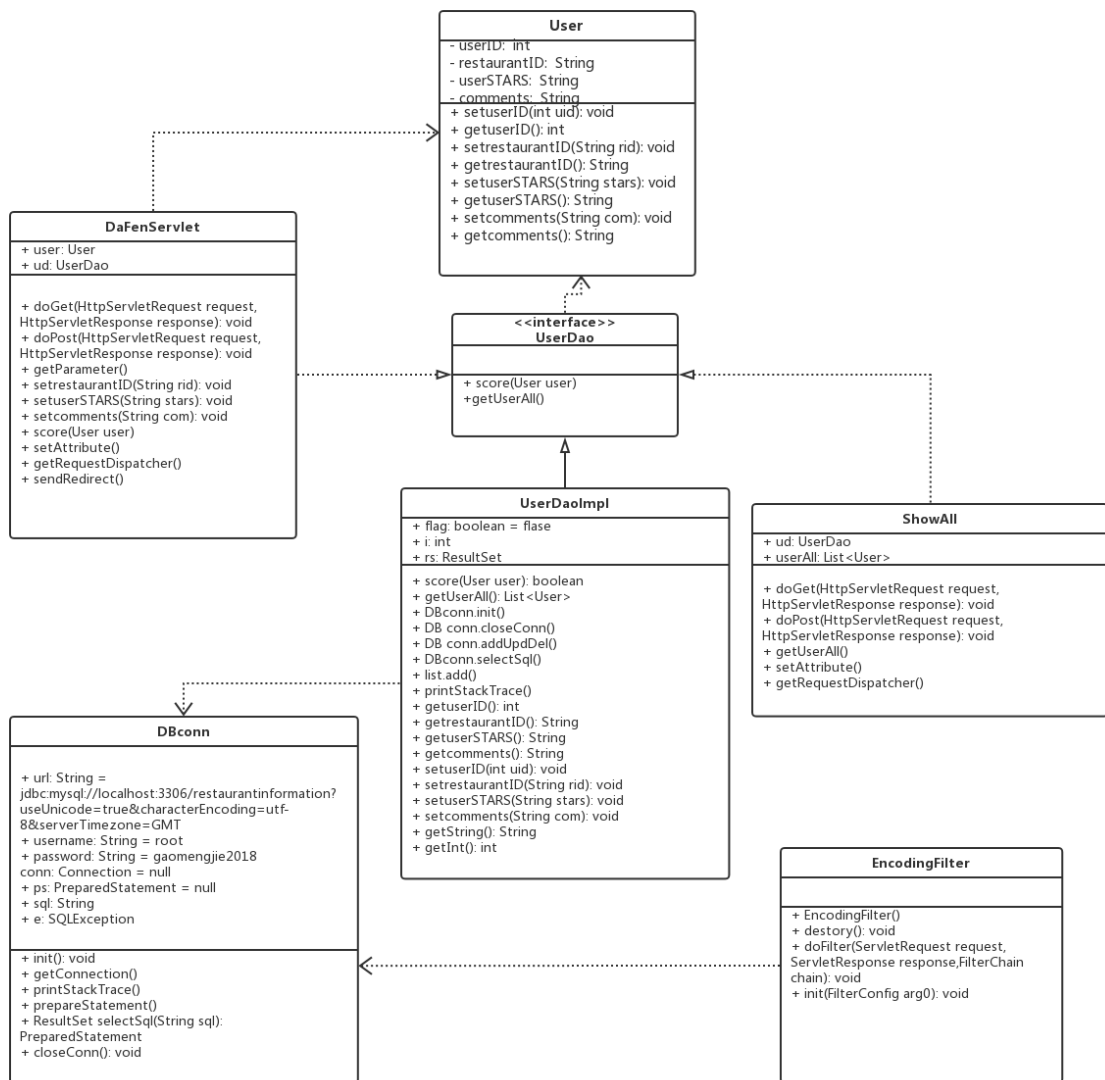


**Figure 2.4: UML diagram of peanut bank**



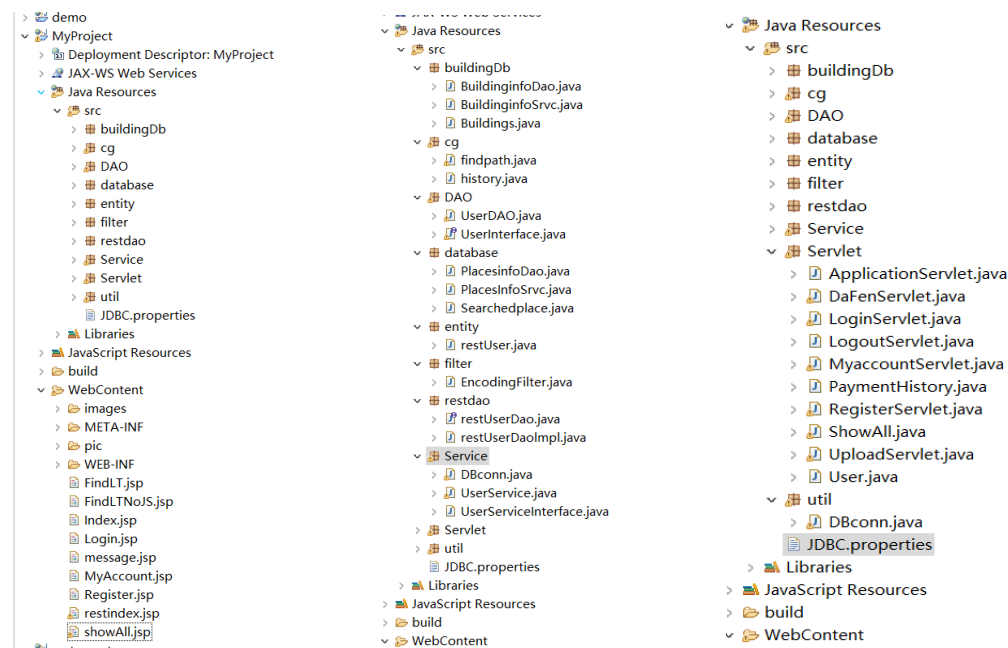
**Figure 2.5 UML deployment diagram of the project**





**Figure 2.6 UML class diagram of the software**

## Apps:



**Figure 2.6 directory structure**

There are two main file which are Java Resources and WebContent. Java Resources include src folder which is the source directory for our Java servlets .java files and lib file is the library directory for our Java library .jar files. WebContent include all the jsp file, and the package of database also in a folder named WEB-INF which store in WebContent. And a folder named META-INF is a standard directory for configuration information.

We explore a war package and then upload this package by FileZilla.

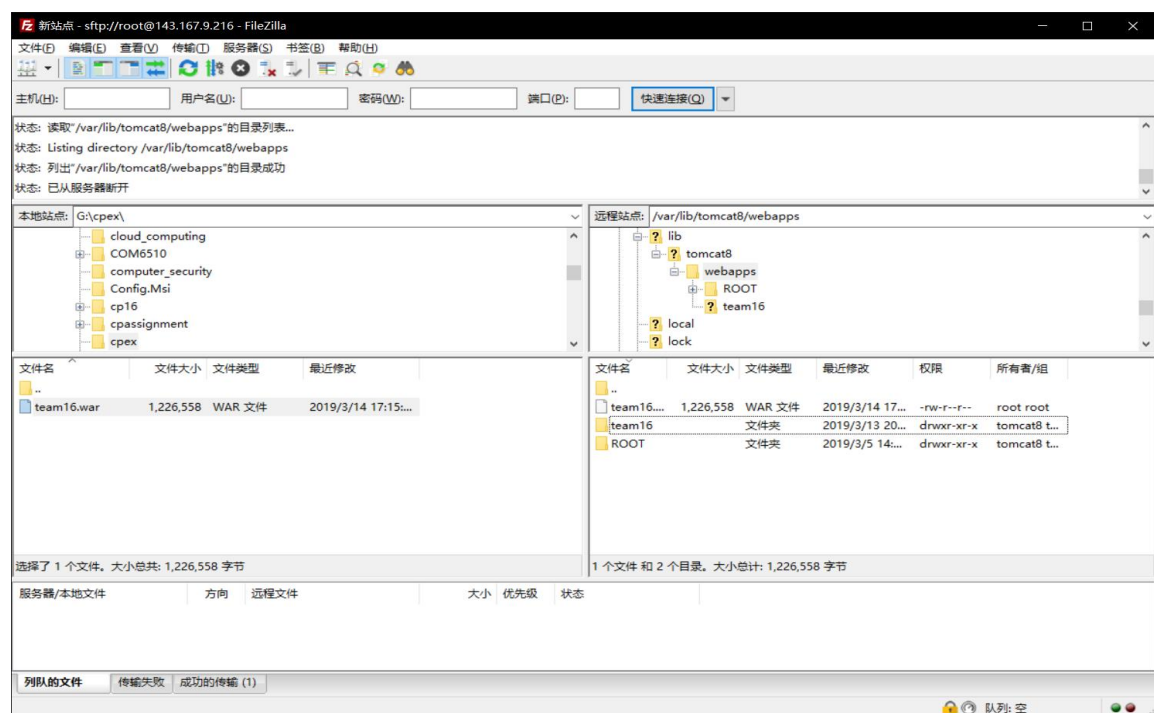
### 3. Technical

Our cloud platform can be accessed through the following link:

<http://143.167.9.216:8080/MyProject/>

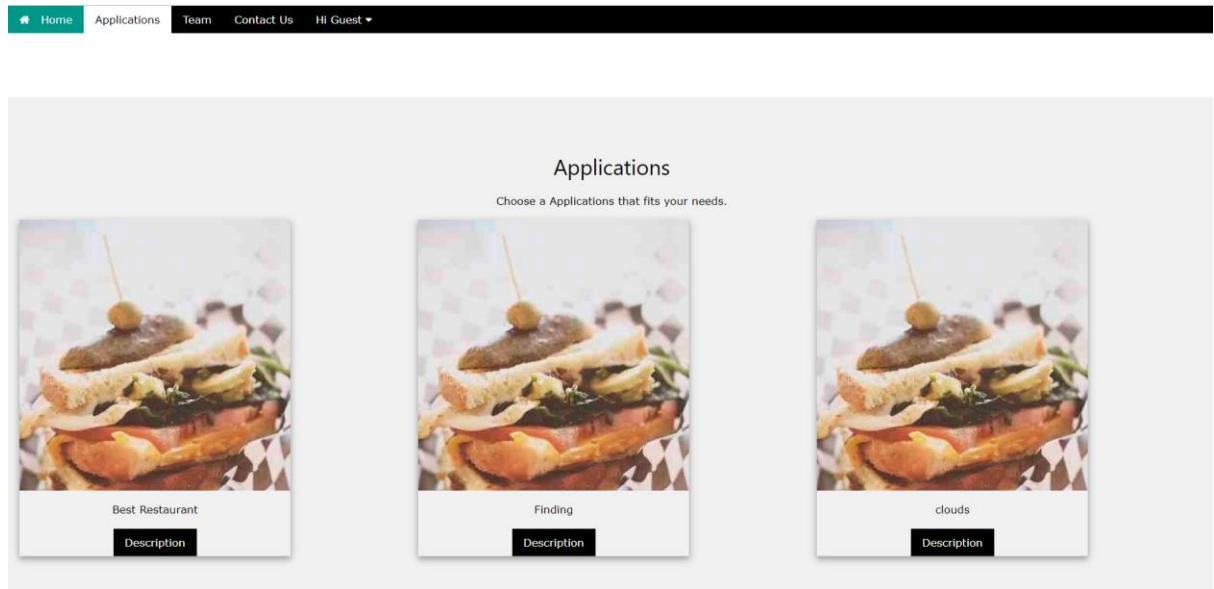
We made several attempts to test how to upload the data to the cloud since we failed to do so following the instructions in the slides.

We use Eclipse Jee to create our dynamic web project to edit locally. The current main structure of our project can be concluded as follows, currently, we have 5 basic interfaces, namely index, login, myAccount, best Restaurant and myTimetable, and have completed the simple connection between these pages to achieve mutual jump.



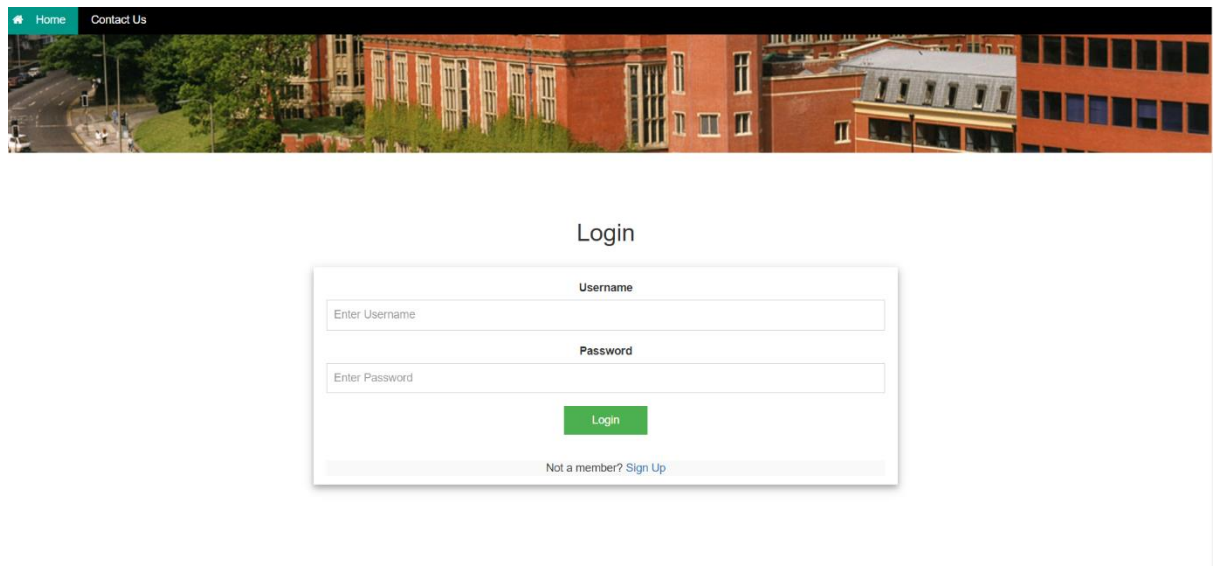
**Figure 3.1: FileZilla**

Figure 3.1 above is the interface of FileZilla, which is the tool we used to upload our web project to the virtual machine in Windows. We decided to use Windows rather than Linux because Windows provides us with more visible interfaces such as FileZilla and JEE and they are easier than using the terminal in Linux.



**Figure 3.2: Homepage**

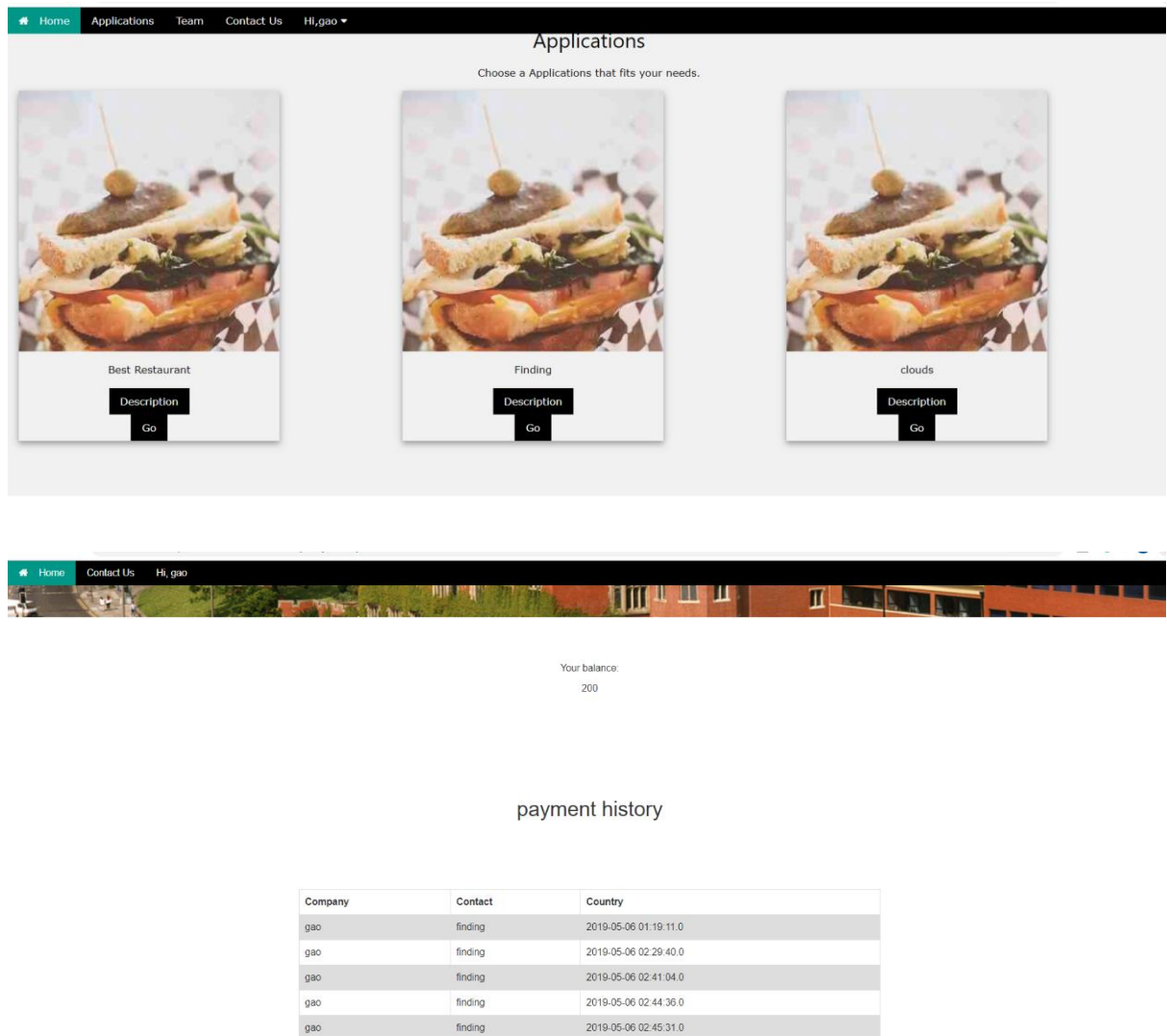
Figure 3.2 is the homepage of our cloud platform. We use the CSS file from <https://www.w3schools.com/w3css/4/w3.css>. Normally, this is the homepage for users who have not signed in for the first time. Every picture we used is saved in PIC folder and uploaded to the tomcat server. This is the shows all two apps and a test app for now, however all of them is non-applicable since user has not signed in.



**Figure 3.3: Login web page**

Options are provided for both first-time users and clients who already have an account. After logged in, users will be redirected to the application page where the peanut credit presented






**Figure 3.4: Homepage after login**

Figure 3.4 shows that after first time login, all applications are listed correctly and now able to be utilised. All information about the peanut bank will be presented at another page where contains balance and purchase history.


After purchasing the applications, pictures of the apps can be clicked to guide users to the web app pages.

Home




13.Bobo Express

Location:S1 4QQ




14.New Zing Vaa

Location:S11 8TD




15.Wild Rice

Location:S2 4LJ



16.羯子王

Location:S2 4LN



17.Noodle Inn Centro Sheffield

Location:S1 4GH


Choose a restaurant code: 1

Give a rating: 1

Your comments:

RESET SUBMIT

HomeContact UsHi, gao



score

userID	restaurantID	STARS	comments
18	17	5	The noodles are tasty
19	1	1	not bad
20	1	1	bad
21	1	1	dasdad

back

**Figure 3.5: Restaurant Ranking page**

Figure 3.5 is page for the Restaurant Ranking app. As shown in the picture, each restaurant near the university is presented sorted by the distance. Users are able to rate these restaurants in the aspects of prices, services or hygiene issues. In addition, the app can rank all restaurants according to price, service, hygiene and comprehensive quality, and show them to users. Apart from that, consumers are allowed to leave comments below to show their impression of the restaurants.

Home hi, gao

You also can find your way using google map [Using Google map](#)

Select your start place:

The Diamond

Select your destination:

The Diamond

Go Reset

History:  
Dental School  
38 Mappin Street

**Figure 3.6: Start page of the lecture theatre app**

Figure 3.6 shows the start page of the lecture theatre finding app where you can type your current location and the destination, subsequently users will be redirected to a new page. What's more, all the search history will be stored in an MySQL database while listed at the bottom of the page.

One marker? The place you are finding is here!

**FROM: Diamond**

**TO: Edgar Allen House**



Figure 3.7: Page using Google Statics Maps API

The path will be presented in a picture using the Google Statics Maps API, the current location and the destination will be marked using different colour makers.

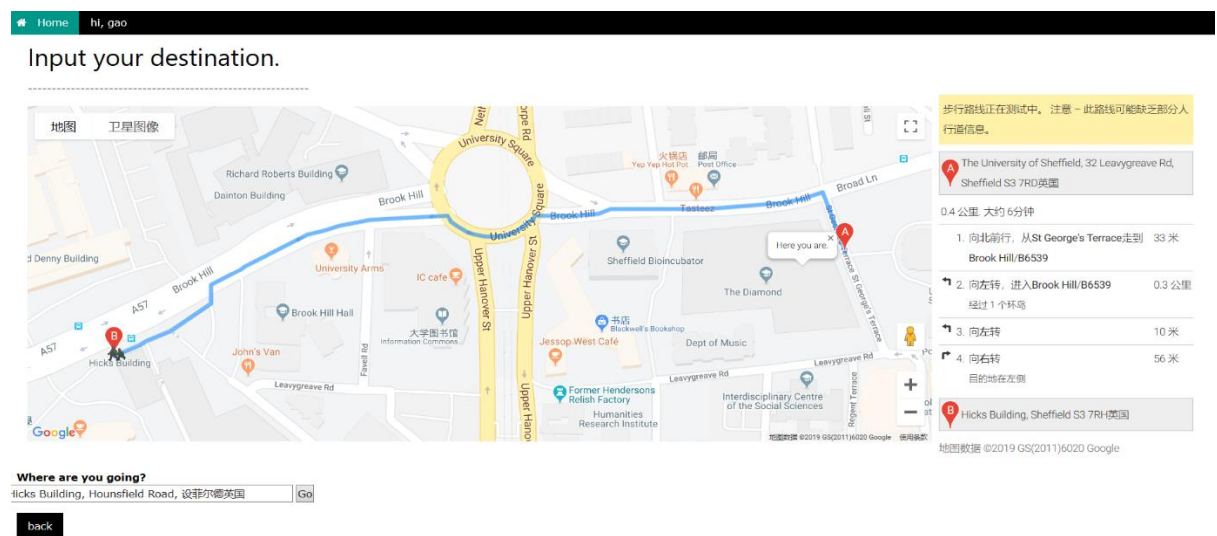


Figure 3.8: Page using JS

Alternatively, users are also able to directly use Google Maps to finish the navigation which gives more detailed information. In this case, since we are currently utilising a browser of Chinese language version, information is still showed as Chinese characters.

## Security

The main security issues we assumed is mainly SQL injection and Cross-Site Scripting (XSS) attacks.

SQL injection usually appears when an input is asked from the users, such as their username, and instead of the real information in need, an SQL statement is provided which will result in an unknown running of the database.

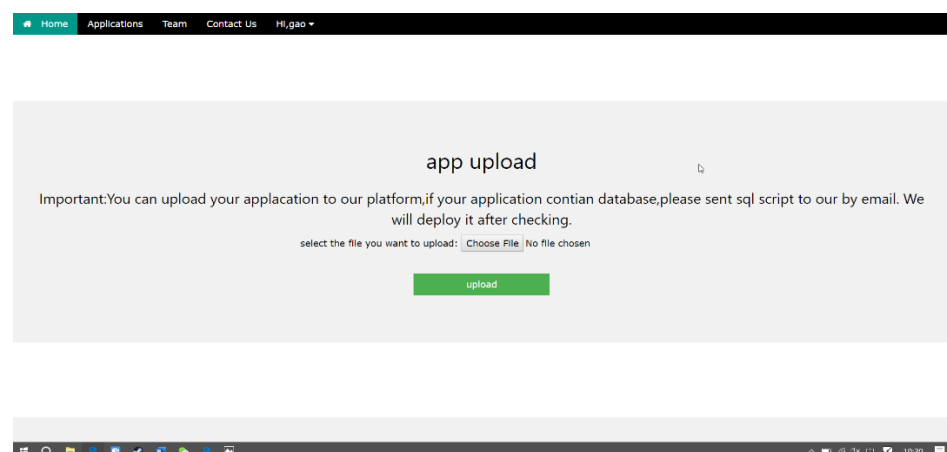
Cross-Site Scripting (XSS) attacks are a “type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.”

```
public String getcomments() {
    comments=comments.replace("#", "");
    comments=comments.replace("@", "");
    comments=comments.replace("$", "");
    comments=comments.replace("|", "");
    comments=comments.replace("&", "");
    comments=comments.replace("'", "");
    comments=comments.replace("-", "");
    comments=comments.replace("<", "");
    return comments;
}
```

**Figure 4.1: Input Verification**

The major method we are using is to filter and replace the dangerous characters which could potentially cause illegal input, protecting the database from the SQL injection and XSS attacks.

Apart from that, location data is rather sensitive in this project, hence we decided that the information about current locations and destinations will be pre-set as places using coordinates(longitude and latitude), hence users are only capable of selecting these locations rather than typing their own words, which efficiently prevent the SQL injection and XSS attacks.



During the uploading process, directly submitted database will not be acceptable, instead, procedures of the verification on the database will be executed manually before the deployment.

#### 4. Teamwork

Each of our team members has made best effort to this project. The tasks we've been assigned to do are as follows:

Progress	Tasks
Design	We finished the project structure design through discussions, we made sure the structure is approved by every member of us.
Coding	Gang Chen and Xiang Li respectively are responsible for the Servlet part, which is the realisation of two web apps, Mengjie Gao and Tan Chen created the JSP files and the login/peanut banks micro-services, apart from that, Mengjie Gao and Gang Chen built the MySQL database and connected the database to the software.
Reviewing	We all took part in the review and debugging progress, everyone in the team had provided some solutions to occurred bugs.
Deployment	We realised the deployment of the whole platform in a library computer, therefore we all made our effort to built the project.

Name	Artefact
Gang Chen	Find the lecture theatre App
Mengjie Gao	Peanut Bank
Xiang Li	Restaurant Ranking App
Tan Chen	Login

**We all agree that all each member of our team has made 100% of their effort, which makes the total effort 400%.**