SEPP CW3

SE part

By Binbin Zhan (s1945599)

Louis(Chang) Yu s1802871

4.5

For the tool I used in this coursework, I would like to introduce idea which I used to write java code.

I chose IDEA based on following reasons. Firstly, IDEA has a simpler and more comfortable interface (compared to Eclipse).Secondly, IDEA has better support for Java projects built with Gradle. In addition, IDEA is more friendly to package management and use In the end, it's much easier to debug than Eclipse.

IDEA will automatically scan the entire code and build the project. For the missing package, add dependencies in build. Gradle and sync to download it automatically. After you writing the code, run main.java for debugging. Make changes to the code based on the debugging information.

The advantages of IDEA is it has simple and friendly interface, project management is very convenient, debugging information is complete and easy to observe, which can well display the function stack and the value of each variable.

I would like to try another compiler next time.

For the approach I used in this coursework, I would like to talk about Gradle which I used to manage my Java code. I chose Gradle because it has more concise dependency management system (compared to Maven) that supports dynamic version dependencies. The project structure is more elegant and supports mixed builds of languages such as Java Groovy which satisfy the agile process requirement in this coursework.The plugin mechanism is much simpler and more flexible than Maven's XML based configuration for Gradle tasks.

Basically, I modified Gradle files of the project and used the groovy language to write the JAR packages that the project

needs in compiling, running and testing. After editing they will automatically generate the corresponding Gradle tasks.

The advantages of Gradle is it has less configuration dependent code but has all functions like packaging, compiling and testing.we can also customise tasks to achieve required functions.

Next time I will write Gradle myself and try to build the whole project from scratch.

4.6

The whole system design is followed the concept of client-server architecture. First of all, the client, shielding individual, catering company and supermarket parts would directly interact with the user and operate the data. Secondly, localhost would send corresponding request to the server after finishing the operation to data. Then the server would take operation and send back to client.

In my view, client-server architecture would be a better choice than the service-oriented architecture, this is because in this system, we only need to use some simple server functions. If we divide server functions according to different requests, the workload of service gateway will increase, and this is unnecessary.

4.7

For security of this system, we suppose the DDoS attack may happen, we can defend this type of attack by recording the invalid IP and Use firewalls to reject requests from blacklisted IP addresses. Moreover, the passwords of server administrator might be cracked, we can Increase the complexity of the administrator password, change the administrator password

regularly and limit the number of password errors per day for the account.

For reliability of the system, we can perform a stress test, that is, plenty of clients make requests to the server per unit time, to see whether the server can withstand the pressure of requests.

For usability of the system, the first principle is to make sure the system adapts to different browser standards to ensure normal display on different browsers. The second one is that all links must be properly displayed on the page.

4.8.1

We use java constructs a runnable food box system, there might be some problem at Junit test. local variable scope was tried to be restricted. And wide use of local variable to make sure the safe of the functionality of required 8 use cases for the food box system.

4.8.2

Object oriented design was used to implementation the food box system. Encapsulation interfaces, override was used to reduce the redundance of the code.

4.8.3

Code has comments when its meet logic which is not obvious. And the documentations followed the instruction and guide from the lecture. If there is change or updates needs, both the code and documentation can be modified within reasonable time and cost.

4.8.4

In the Food box system code, Junit 5 was used to test methods. It may not cover all the method, but critical method like registering, order food box, edit food box, been covered.

4.8.5

The system can operate frequent correct inputs, and few edge case, but there might be error for some inputs edge cases .

4.8.6

The system can operate frequent correct inputs, and few edge case, but there might be error for some inputs edge cases

4.8.7

For the 3 required classes, Junit 5 test the most critical method for the 3 required classes.

4.8.8

Junit tests covered frequent correct inputs and will throw exception for some incorrect inputs, there may missing some test for edge cases.

4.8.9

Most the unit tests are passed. And most the method are covered.

4.8.10

Agile process was used for this course, since the goal is to building a food box delivery system to service the public during the epidemic. Using widely used tools like python to run Flask server and run test using Junit5 and using gradle to avoid building wheels. Maximum the focus on the developing and testing.

4.8.11

Since learn Java in year one it is easier to use python, and use object oriented programming language java are easier to simulate the thing in reality. It seems like that is main reason of using java rather than Haskell. And gradle can easily Junit test in java intellji to test each method.

4.8.12

impact of using a Client-Server architecture on the design is security, we may assuming DDos attacts will happen. And having solution for security during the object-oriented design.

4.8.13

the main security or privacy attack is DDos, we may recording the invalid Ip and reject suspicious IP from the record by using firewalls.

4.8.14

Stress test was used for testing the reliability of the system. Testing whether the server can normally operate under plenty of clients request within a certain time

4.8.15

The chosen 2 appropriate principle of user interface design is the simplicity principle and the visibility principle to ensure the web and mobile app interface is easy and communicating clearly without distracting user with redundant information.

4.8.16

I tried to be objectively assessing the essay and pay effort to the self-assessment, but it seems like there be some problems that hard to find in my own point of view.