MSc in High Performance Computing with Data Science

Coursework for Software Development Part 3

Review&Planning Report

Exam No. B114262

March 29, 2018

Table of Contents

Table of Contents ..... **ii**
Revision History ..... **ii**
1. Introduction **1**
1.1 Purpose 1
1.2 Intended Audience and Reading Suggestions 1
2. Prototype Introduction **2**
2.1 Database design 2
2.2 Work flow of the Prototype 2
2.3 Implantation of Prototype 3
2.3.1 Login 3
2.3.2 Register 4
2.3.3 Game mall 4
2.3.4 View Collection 5
2.4 Front-end Prototype 6
3. Review the Project **7**
3.1 Development Environment Review 7
3.2 Functional Review 7
3.3 Nonfunctional Review 8
3.4 Planning Review 9
4. Planning for the future **11**
4.1 Remaining Tasks 11
4.2 Timeline for development 11
4.3 Risks in prototype 14
4.4 Risks in released version in the future 14
Appendix A: How to import database **16**
Appendix B: How to compile and run the program **17**

Revision History

Name	Date	Reason For Changes	Version
xx	2018/3/28	First version	1.0
xx	2018/3/29	Final version	2.0

1. Introduction

1.1 Purpose

This document aims to introduce what has been implemented in the prototype version and give a brief review compared with the requirement critical in submission 1. In addition, there will be a plan for the future development and the timeline of that will be given as well.

1.2 Intended Audience and Reading Suggestions

The intended audience includes:

1. The development of coding personnel
2. UI designers
3. The testers

2. Prototype Introduction

2.1 Database design

In Figure 2-1, this ER graph shows the structure of the database which the program connects to. There are 5 tables, user (basic user information), userCollection (what user has), itemTable (what user could buy), equipmentTable (equipment detail information) and soldierTable (soldier detail information).

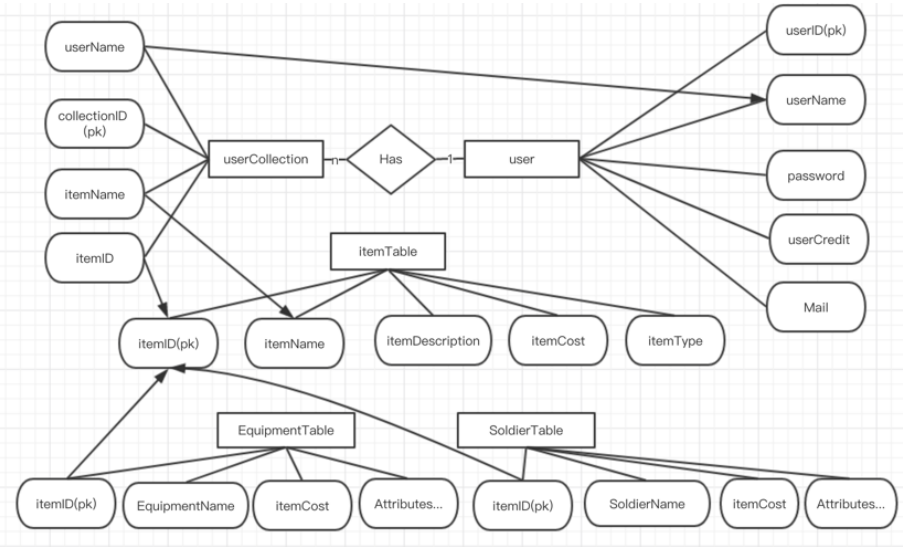

Figure 2-1 ER graph

2.2 Work flow of the Prototype

In the program, Figure 2-2 shows the work flow about the game system prototype. In the login page, users could choose to login or register. If choosing login, users need to input the username and password to check whether this user exists and the correctness of the password. If the check is right, user could enter the main page. If choosing register, user need to input the username (which must be unique compared with those in the database) and the password. In the main page, this prototype realized 2 main functions, Viewing User Collections and Game Mall Function (including buying equipment and buying soldiers). Viewing collections page contains all user's collections. While, when entering the game mall, the command line will show all items users could buy with their ID,

name, description and cost. Users could input the ID of the item to buy the item. However, if there are not enough credits in user's balance (shown in Game Mall Page). User can't buy this item.

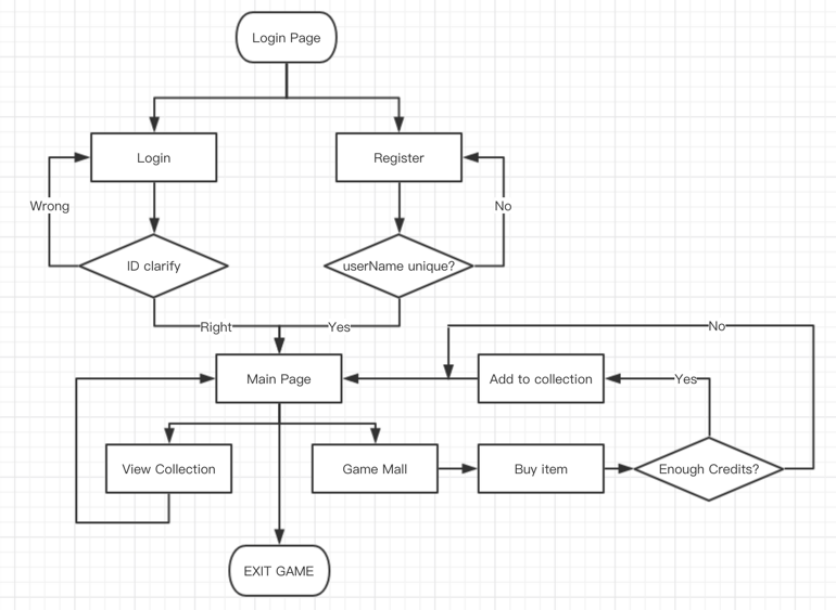

Figure 2-2 Work Flow of the product

2.3 Implantation of Prototype

The implantation of different functions in the prototype program will be illustrated in this part.

2.3.1 Login

In login page in Figure 2-3, user could choose 1. Login 2. Register and 3.Exit. User could input the selection in the command line.

```
Login Page
1.Login
2.Register
3.Exit
```

Figure 2-3 Login Page

If user chooses "1.login", user should input the username and password (username: user1 , password: abc , this is for testing). If user inputs the wrong one, he/she could not enter the main page in Figure 2-4.

```
Login Page
1.Login
2.Register
3.Exit
1
Please input the Username:
hi
Please input the Password:
aaa
Username or password wrong, please login again or register a new account!
Login Page
1.Login
2.Register
3.Exit
```

Figure 2-4 Wrong input

If user inputs the right one, he/she could go into the main page of the prototype in Figure 2-5.

```
Please input the Username:
user1
Please input the Password:
abc
Login Successfully!
----------------------
Welcome to the Game:
1.Enter the Game Mall
2.Show Personal Collection
3.Exit
Please enter your selection:
```

Figure 2-5 Right input

2.3.2 Register

If user chooses “2.Register”, user should input a unique username and password (username can be the same with those in the database). If user inputs the existing one, system will need user to inout again in Figure 2-6.

```
Login Page
1.Login
2.Register
3.Exit
2
Please input the new user name:
user1
user1
User name has been registered, please input a new username!
Please input the new user name:
```

Figure 2-6 Register Error

If the user inputs a proper username, after he/she inputs the password, a new account will be created successfully and the program will go into the main page in Figure 2-7.

```
2
Please input the new user name:
helloworld
Please input the password:
abc
success insert a new user
----------------------
Welcome to the Game:
1.Enter the Game Mall
2.Show Personal Collection
3.Exit
Please enter your selection:
```

Figure 2-7 Register successfully

2.3.3 Game mall

After users go into the main page, they could choose to “buy items” (input 1) or “view collections”(input 2) or “exit”(input 3)

If user chooses “1. Enter the game mall”, the mall will show the list of all items users could buy with the price, the description and the ID in Figure 2-8. Besides, at the top, the user balance will also be seen.

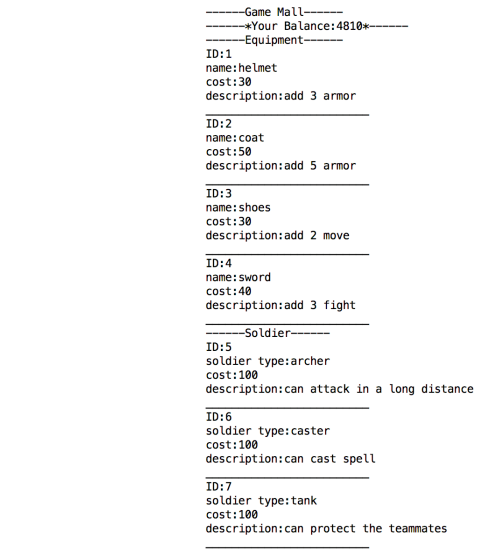

Figure 2-8 Game mall

If the user want to buy one item, he/she need to enter the ID the item (helmet—ID:1 etc.). If the user has enough credits, the trade would be successful in Figure 2-9.

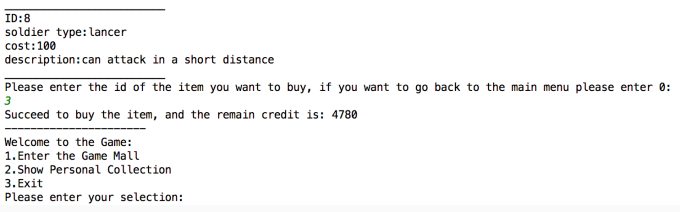

Figure 2-9 Buy item

And the item will add into the user collection. If the credits is less than the cost of the item in Figure 2-10, the trade would be failed in Figure 2-11.

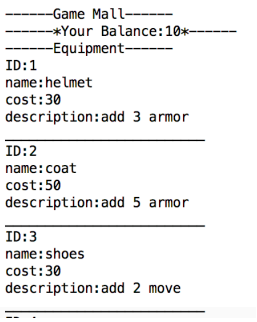

Figure 2-10 Not enough credits

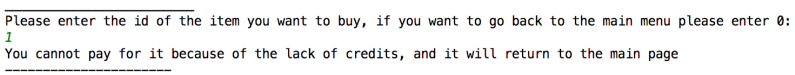

Figure 2-11 Fail to buy

2.3.4 View Collection

If user chooses “2.show personal collection”, all of the user has will be shown in Figure 2-12. If user has bought something in the game mall, the collection will be updated as well.

```
Welcome to the Game:
1.Enter the Game Mall
2.Show Personal Collection
3.Exit
Please enter your selection:
2
------Collection------
------Equipment------
||helmet||
||sword||
||sword||
||coat||
||shoes||
----------------------
------ Soldier ------
||archer||
||lancer||
||tank||
||lancer||
||lancer||
||lancer||
----------------------
----------------------
```

Figure 2-12 Collections

2.4 Front-end Prototype

This prototype also contains the front-end page prototype made by Axure RP 8. In the folder named “front-end prototype”, there are the .html files for the pages which have been explained in submission 2. Figure 2-13 shows the start.html5 file, and it is the start of the front-end pages.

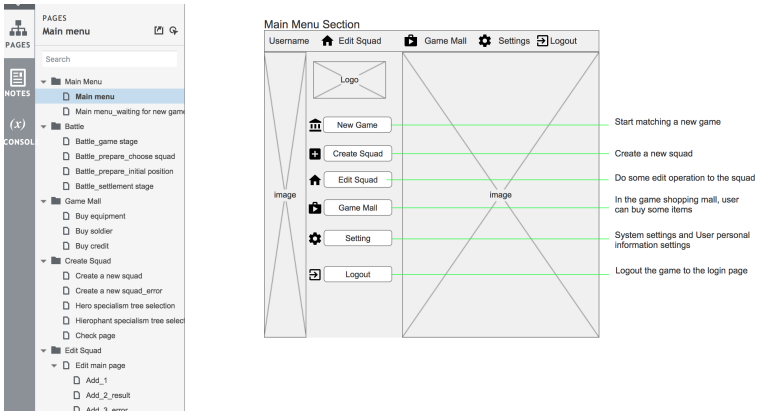

Figure 2-13 Front-end main page

3. Review the Project

This section will give a brief review between the prototype and the expected one in the submission one.

3.1 Development Environment Review

This review is in table 3-1

	Requirement Instruction	Current Prototype
Developing Language	Jave EE	Jave SE
Project Result	Website Game	Command Line
Database chosen	Hibernate	MySQL
Project Structure	Struts2+Spring+Hibernate	Single node

Table 3-1 Environment review

- For developing language, I choose JavaSE because there is only one developer to finish this project and I have to work on the local PC. Considering the efficiency, I use javaSE instead of Jave EE. But, it is convenient to combine the work in Java SE connected to Java EE.
- In the beginning, the game is expected to develop as a website game. While, in the prototype version, I use command line instead because it is more light and suitable for both developing and testing. If using SSH structure in the future, we only need to put the function code in the service class and give some interfaces to the front-end page.
- The database used for the prototype is MySQL because Java has JDBC to connect to MySQL which means we could clear check the statement in the database. If we use Hibernate as the database in the future, what we only need to do is to change the SQL code into HQL and build the domain class, the DAO class and the DAO implantation class.

3.2 Functional Review

This section will first analyze the currently developed functions with the functions in the requirement instructions.

- 1. For the login function, the current prototype has completed what it is required in the submission 1. This part contains checking the input information and giving proper tips to help user login or register. As a result, this function is not altered and has been finished.

- 2. For the registered function, the current prototype almost implements most of the expectation one including recording new user name, password and checking the unique of the username. The only missing part is the user email record because in real registration part, after filling the email address, user would receive a confirmation email but it is hard to realize in the command line version. So, this column is not added into the database.
- 3. For the Game mall section, first of all, the buying the equipment and the soldier components completely meet the requirement mentioned before including showing the detailed information about the items and adding one new item into user collection and checking the user balance in case that the amount of credits is less than 0 and also recording user credits. But this prototype did not realize the Buying Credits component since this section refers to user real-world information and important finance information, even refers to the legal issues. Therefore, this part needs to communicate with the company about the developing details in the future.
- 4. Last, as for the viewing collections component, now the prototype has finished most of the functions in the requirement instruction including showing the equipment and the soldiers, which user has. However, this section did not contain neither hero nor hierophant because this prototype did not include the squad creation part so the hero and hierophant initialization could not be achieved. As a result, the information of hero and hierophant is not enough to build the table or shown in the collections. This would be added after the squad creation part finished.

As for those components not realized in the prototype, those would be introduced in the section 4.1 Remaining Tasks.

3.3 Nonfunctional Review

	Requirement Instruction	Current Prototype
Performance	Parallel	Serial
Multi-platform	Multi-platform	Multi-platform
User Interface	Web	Axure

Table 3-2 Nonfunctional review

- About the performance, this program keeps serial processing the game. Mainly because the developer has not found a proper parallel structure for this project. In the future, this project will change into the parallel one to meet more users.

- As for multi-platform, this program is developed by Java so any platform could run this program unless Java is not installed there. This section meets the basic requirements.
- Considering the user interface, the current prototype has done a front-end one with the help of Axure but it is only some actions jumping to other links. There still need more improvements for the UI in the future.

3.4 Planning Review

- Requirement analysis and validation
  Plan: 2018.02.07-2018.02.26
  Reality: 2018.02.01-2018.02.20
  Status: Finish in advance
  Review: This part fits the time schedule of plan.
- Project UI design
  Plan: 2018.02.27-2018.03.20
  Reality: 2018.02.10-2018.03.09
  Status: Finish in advance but not completely suitable
  Review: This part finished in a prototype in Axure but not the real UI. In the future this part need to be improved in detail.
- Database Design
  Plan: 2018.03.30-2018.04.19
  Reality: 2018.03.23-Now
  Status: Some finished, some unfinished
  Review: Currently database has been designed for the prototype part but not the real project part.
- Login& Register development
  Plan: 2018.04.20-2018.06.14
  Reality: 2018.03.23-Now
  Status: Finish the command line version but not meet the requirement
  Review: Only finish the command line login and register components, still need to be developed into web version.
- Game mall development
  Plan: 2018.04.20-2018.07.20
  Reality: 2018.03.25-Now
  Status: Partly finish

Review: Finish the buying equipment and soldier functions but not finish buying credits function.

- View collection development
  Plan: 2018.04.30-2018.05.30
  Reality: 2018.03.25-Now
  Status: Partly finish
  Review: Finish the part without hero and hierophant in command line version. Need more improvement with the user interface's and other parts' help.

In short, most of the time estimates are correct (in the view of prototype) and the workload for prototype is suitable and appropriate for one developer. However, if achieving one whole project to a release version but not just the prototype, there still need one developing team.

4. Planning for the future

4.1 Remaining Tasks

- Squad management component development
- Game mall function --- Buying Credit component
- Battle system component development
- System setting development
- Administration subsystem development
- Project UI design
- Change the database connection way --- From JDBC to Hibernate
- Build the structure of Struts2 and Spring
- Connect the front-end to the back-end
- System Integration test
- Environment Integration test
- User Document
- Release the game

4.2 Timeline for development

Considering the heavy workload of this project, it is really hard to finish this project still with only one developer. Therefore, the timeline below is fitted for one well-trained developing team including the project manager, the developers, the UI designers and the testers.

2018.03.30-2018.04.30

- Web UI design (do 20%)
- Database design for the whole project
- Build the Struts2 and Spring structure combined with the prototype version
- Character Model design (do 25%)

2018.04.30-2018.05.30

- Web UI design (50%)
- Squad creation development (50%)
- Character Model design (50%)

- Change the database connection way --- From JDBC to Hibernate
- Write the domain level of SSH structure

2018.05.30-2018.06.30

- Web UI design (70%)
- Squad creation development (100%)
- Battle System development (40%)
- Character Model design (80%)
- Write the DAO level of SSH structure
- Finish Game Mall remaining task
- Testing

2018.06.30-2018.07.30

- Web UI design (85%)
- Battle System development (80%)
- Squad Management development (20%)
- Character Model design (100%)
- Write the Action and Service level of SSH structure (50%)
- Testing

2018.07.30-2018.08.30

- Web UI design (95%)
- Battle System development (95%)
- Squad Management development (60%)
- Write the Action and Service level of SSH structure (75%)
- Administration subsystem development (30%)
- Testing

2018.08.30-2018.09.30

- Web UI design (100%)
- Battle System development (100%)
- Squad Management development (80%)
- Write the Action and Service level of SSH structure (90%)
- Administration subsystem development (60%)
- Connect front and back of the program (40%)
- Testing

2018.09.30-2018.10.30

- Squad Management development (100%)
- Write the Action and Service level of SSH structure (100%)
- Administration subsystem development (80%)
- Connect front and back of the program (80%)
- Improve system performance and other nonfunctional requirements
- Testing

2018.10.30-2018.11.30

- Administration subsystem development (100%)
- Connect front and back of the program (100%)
- Improve system performance and other nonfunctional requirements
- System Integration test
- User Documents

2018.10.30-2018.11.30

- Improve system performance and other nonfunctional requirements
- Environment Integration test
- User Documents
- Company check first version

2018.11.30-2018.12.30

- Fix the bugs found from the company
- Improve system performance and other nonfunctional requirements
- User Documents
- α Testing
- β Testing

2018.12.30-2019.1.30

- User Documents
- Fix the bugs from the testing
- Deliver the Project
- Release the game

4.3 Risks in prototype

- **Insufficient Documentation:**
  Method: Eliminate the risk
  Strategy: When the developer works on the project, he needs to record what they have done and how to do in the document. After finishing the prototype version, the developer have recorded the procedure about how to import the database into other people's computer and how to run the program on other computers.
- **Technical Issue:**
  Method: Eliminate the risk
  Strategy: The developer chose Java which is the familiar language for the developer so that the prototype could be finished successfully.

4.4 Risks in released version in the future

- **Unachievable Schedule**
  Method: Control the risk
  Strategy: Project Manager needs to monitor the progress of project regularly. Developers should finish every step in the time plan on time. This risk will always exist. To reduce the damage, the team needs to finish their job following the plan table.
- **Management Issue:**
  Method: Avoid the risk
  Strategy: Some people in the team are responsible for the communication between the client company and the developing team so that the client company could know the progress of the project in time. Besides, there should be a team manager to collect the whole team statement and manage the whole project.
- **Extra Functionality:**
  Method: Control the risk
  Strategy: Communicate with client company regularly to confirm the requirement in time so that the risk could be reduce to a low level. Besides, speed up the whole project with the proper quality. If there are some additional functionality, developers can have enough time to finish it.
- **Lack Quantitative Measure:**
  Method: Avoid the risk

Strategy: Choose and use some quantitative software to monitor the program. Find some useful methods to count the work progress.

- **System Performance Issue:**
  Method: Assume the risk,
  Strategy: The system may have many users to play so the performance is one of the most important factors to the game. First, the team could choose parallel programming to develop this game. Then, about the database, the team could use some distributed database like Hibernate to replace MySQL if needed, so that the speed of calling database would improve.

Appendix A: How to import database

In order to import the .sql file into your computer, you need to follow these steps:

- Install MySQL
- Start MySQL with the command:
  ```
  $ sudo /Library/StartupItems/MySQLCOM/MySQLCOM start
  ```
- Use terminate to login MySQL:
  - First check whether the Path contains Mysql path:
    ```
    $ echo $PATH
    ```
  - If not, add the path of mysql:
    ```
    $ PATH="$PATH":/usr/local/mysql/bin
    ```
  - Then, login with the name and password (this command need to input MySQL password, in the writer's database, the password is 123456, if the password is not the same, please change the password in the source code classes and compile the source code again):
    ```
    $ mysql -u root -p
    ```
- In the mysql command line, import the sql file:
  ```
  mysql> source /Users/<the path of SD.sql>/SD.sql
  ```

That is how to import the sql file.

Appendix B: How to compile and run the program

Please follow the instructions in the “Readme.txt”.

Shell Input:

∗ ./compile.sh

compiles the source and builds jar (Game.jar) into the root directory.

∗ java -jar Game.jar

run the compiled runnable jar

* java -jar SoftwareDevelopment.jar

run the default jar (mysql username:root password:123456)

Tips for Exceptions:

If MySQL’s username is not "root" or the password is not "123456", please come into the src/com directory, and change the code in all 5 java classes:

```
static final String USER = "root";
static final String PASS = "123456";
```

into:

```
static final String USER = "<username>";  //Your username
static final String PASS = "<password>";  //Your password
```

Then, compile again and run the Game.jar so that the JDBC could run successfully in the program

File Structure:

|code

--src : the Java code about this project

--lib : all of the external jars required for compilation except standard java jar files

--compile.sh : the script used to compile and show the project results

--MANIFEST.MF : rule the output main class and rule the external jars import

--SoftwareDevelopment.jar : the default runnable jar

* java -jar SoftwareDevelopment.jar

--Game.jar : the jar compiled by the script "compile.sh"

* java -jar Game.jar

--SD.sql : the sql file which needs to be imported into Mysql database. About how to import, please follow the instruction in "How to import database" section

|file

--B114262_SD.pdf : The introduction about the prototype, the review of the program and the plan for the future development

|front-end prototype

--xxx.html : the prototype of front-end pages of the game

--xxx : other documents like css and js