Lab Report for Software Engineering course
Lab 6: Demand Documentation

Wang, Chen 16307110064
Liu, Jiaxing 17302010049
Huang, Jiani 17302010063
Tang, Xinyue 16307110476
School of Software
Fudan University

June 20, 2019

Contents

Chapter 1

Revision History of the demand documentation

Modifier	Modify Time	Approver	Modified Chapter
Huang Jiani	2019-6-19	All	Refined Function Requirements

Chapter 2

Project Outline

2.1 Background information of the project

Now some customers come here to make a request to Star Dad, and hope that the development team can develop a beverage online sales system according to the needs of customers. Now you need to carry out the basic arrangement and rational understanding of the customer's needs (which are specified more clearly in the next section), communicate with the customer, conduct line analysis and refinement, and guide the development of the development team.

2.1.1 General demands of the project

There are two specific restrictions or conditions about the project:

1. Number of users: 1000 people / day
2. Budget: 1 million yuan

2.2 Overview of the features of the project

2.2.1 Beverage Store

2.3 Module division of the project

2.4 User characteristics of the project

2.5 Runtime environment

2.6 Conditions and restrictions

Chapter 3

Feature Demands

3.1 Refined function requirements

According to the interview record of the lab assigner, the whole system should satisfy the requirements in the following perspectives:

3.1.1 Administration access authorization

1. Any shop assistant must first get authorized administration access of the system before he conducts all the normal routines including register, login, matching drinks, getting drink descriptions and ordering.
2. Anyone except shop assistants is unauthorized to the administration access.
3. To get the authorized administration access, the shop assistant must do XXXXXX.

3.1.2 Register

1. Any shop assistant can use the unique username and password to register.
2. The username will be persistently recorded in the user.csv (?) after the shop assistant registers.
3. The username must start with **starbb_**;
4. The username can consist of **letters**, **numbers** and **underline**, excluding any other symbols;
5. The username should have a length greater than or equal to 8 and less than 50.
6. The password can consist of **letters**, **numbers** and _, excluding any other symbols;
7. The password must consist of all the three types, i.e. **letters**, **numbers** and _, excluding any other symbols;

8. The password should have a length greater than or equal to 8 and less than 100.

3.1.3 Login

1. Only if the shop assistant logs in successfully can he do the other normal routines including matching drinks, getting drink descriptions and ordering.
2. The shop assistant will log in successfully if and only if the username and password are matched.
3. The login status will be recorded after the shop assistant logs in successfully.
4. If the shop assistant fails to log in, the system will throw a runtime exception to prompt the failed login.
5. If the shop assistant fails to log in because of wrong password, the system will prompt **Username or password error**;
6. If the shop assistant fails to log in, he will not allowed to conduct any other operations.

3.1.4 Matching beverages

1. The shop assistant can get different drinks considering different cup sizes and different kinds and numbers of ingredients.

3.1.5 Obtaining beverage descriptions

1. The shop assistant can obtain and check different descriptions of beverages.
2. The customer can obtain and check different descriptions of beverages.

3.1.6 Order charge calculation

1. The shop assistant can order according to the verbal instructions of the customer.
2. The shop assistant can calculate the order charge including the original price, discount and the total discount charge.
3. The customer can check the order charge including the original price, discount and the total discount charge.

3.1.7 Beverage supported

1. The default beverages include coffee and tea.
2. The default coffee includes Espresso and Cappuccino.
3. The default tea includes GreenTea and RedTea.

4. Different stores can customize their own beverages of local characteristics.
5. Every beverage should have attributes of its name, price and description(?).

3.1.8 Ingredients supported

1. The default ingredients include milk, chocolate, cream and sugar.
2. The prices of ingredients can be fixed by the maintenance personnel.
3. Different kinds and numbers of Ingredients can be added .

3.1.9 Cup size supported

1. There are totally three kinds of cup size : large, middle and small.

3.1.10 Discount supported

1. There are three categories of discount strategies in total : Double eleven, Full count and Combination.
2. Combination strategy has four concrete strategies.
3. The concrete strategies can have superposition.
4. Different categories of discount strategies cannot have superposition.
   (a) Order including both tea and coffee will have 15% discount.
   (b) 2 cups of Large-cup Espresso will have 20% off discount.
   (c) Buying three cups of tea will send one for free.
   (d) Cappuccino second half price.
5. Full count: All drinks full 100 minus 30
6. Double Eleven: All drinks 50% off.

3.1.11 Language switch

1. The system language can be switched to the official language of different countries and regions.
2. The language switch should cover everywhere customers can see and check.
3. The default supported languages are Chinese and English.

3.1.12 Currency switch

1. The currency switch will not consider exchange rate fluctuations.
2. The currency should be switched according to different countries and regions.
3. The default supported currencies are Chinese Yuan, Hong Kong dollar and US dollar.

3.1.13 Price fix

1. The maintenance personnel can fix the prices of all the items.

3.1.14 Configuration and Maintenance

1. The maintenance personnel can configure and maintain all the settings including drinks, ingredients, cup-size, discount, language, currency and price-fixing.

3.1.15 log information supported

1. The system must provide log information for the maintenance personnel.
2. The log information must include records of order errors and successful order cases.

3.2 Detailed description of refined function requirements

3.2.1 Scenario analysis and modeling

We apply the use case diagram to the scenario analysis and modeling procedure. As the diagram shows, there are two actors including customers and shop assistants and the online system in the whole interaction. Ordering and register are the two main use cases. Several sub use cases of ordering like beverage information checking, beverage matching and beverage price calculating. And All these use cases include the smaller login user case.

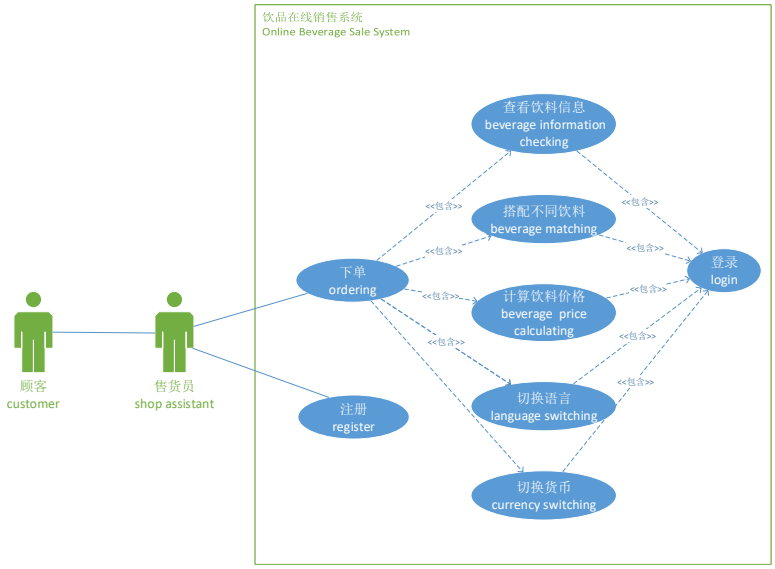

Figure 3.1: Overall Use-case Diagram

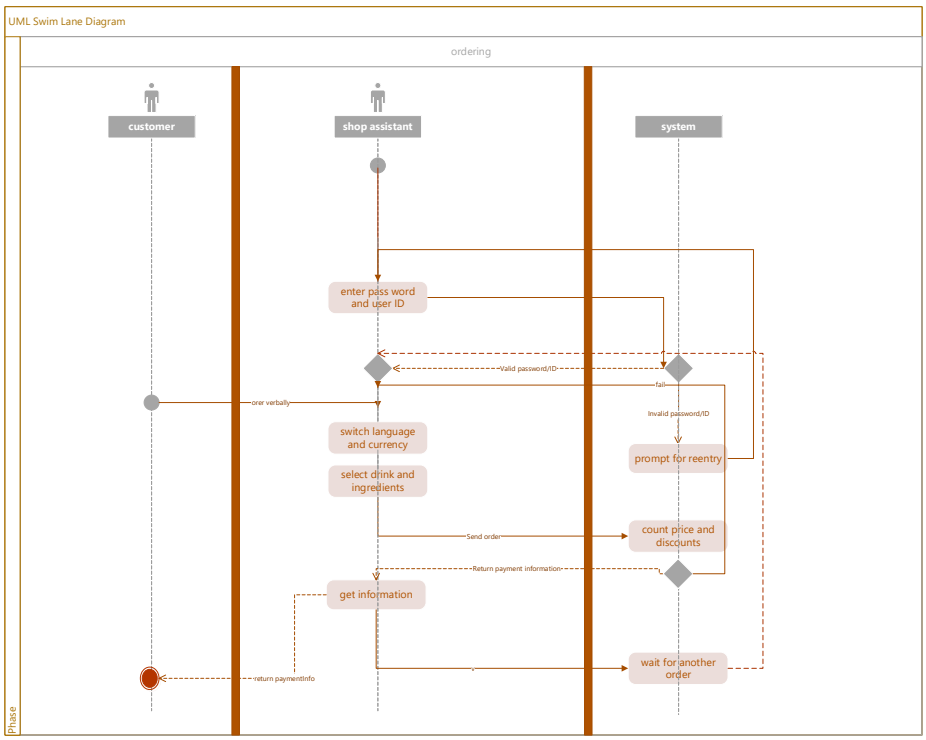

Figure 3.2: UML Swim Lane Diagram

Use case 1: Ordering

In this use-case, we apply the swim lane diagram to concisely describe the relationship between users and the system.

- **Use-case name:** ordering
- **Actors:** customers and shop assistants
- **Target:** The shop assistant can complete the order-making process according to the verbal instruction of customers.
- **Precondition:** The shop assistant has registered before.
- **Triggering condition:** There is a customer waiting to order in the store.
- **Main scene:** After the shop assistant logs in the system successfully, he/she waits for the verbal instructions of the customer. After the customer tells him/her the wanted cup-size, ingredients and kind of beverages, the shop assistant can make an ordering, and tells the customer the total price, discounted price and discount information of the order. After the customer has paid the charge, the order is finished and the system will be waiting for another order.
- **Abnormal scenes:**

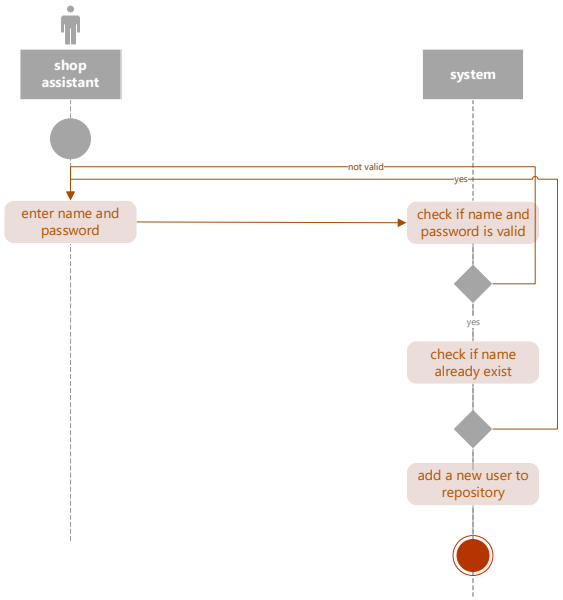

Figure 3.3: Overall Use-case Diagram

1. **Login failure:** If the shop assistant enters wrong username or password, he can get the chance to enter again until he finally enters the correct username and password and logs in the system.
2. **Order failure:** If the order process is interrupted due to the internet or other hardware break, the order process will return back to the calculating procedure after recovery.

- **Frequency:** The ordering use-case will be continuing 24 hours a day only if there is a customer waiting to order.

Use case 2: Register

In the ordering user-case, we apply the activity diagram to describe the whole process.

- **Use-case name:** register
- **Actors:** shop assistants
- **Target:** The shop assistant can complete the register process.
- **Precondition:**
  1. The server of the system is running normally.
  2. The shop assistant has gained the access authorization before.
- **Triggering condition:** None

- **Main scene:** After the shop assistant gains the access authorization successfully, he can enter the system to choose login or register. After he enters the valid username and password, he can use them to login and make orders.
- **Abnormal scenes:**
  1. **Password invalid:** If the shop assistant enters invalid password, the system will prompt corresponding message and let the user enter again untill he enters the valid password.
  2. **Username invalid:** If the shop assistant enters invalid username, the system will prompt corresponding message and let the user enter again until he enters the valid username.
- **Frequency:** Relatively low, since only when there is new and unregistered shop assistant will it be triggered. However, the register use-case should be available 24 hours a day.

3.2.2 Class analysis and modeling

Analysis class extraction

- User
- AccountService
- OrderItem
- Order
- PaymentInfo
- OrderService
- MarketingStrategy
- DoubleElevenStrategy
- FullDiscountStrategy
- CombinationDiscountStrategy
- LanguageService
- MenuService

Function of classes

- **User :** The entity class represents shop assistants. There are two attributes: name and password.
- **AccountService :** The service class is charge of the login, signup, status-checking, name-checking and password-checking of users.
- **OrderItem :** The entity class represents beverages. There are three attributes: name, size and ingredients. Also, it has a public method to calculate the price according to its size and ingredients.

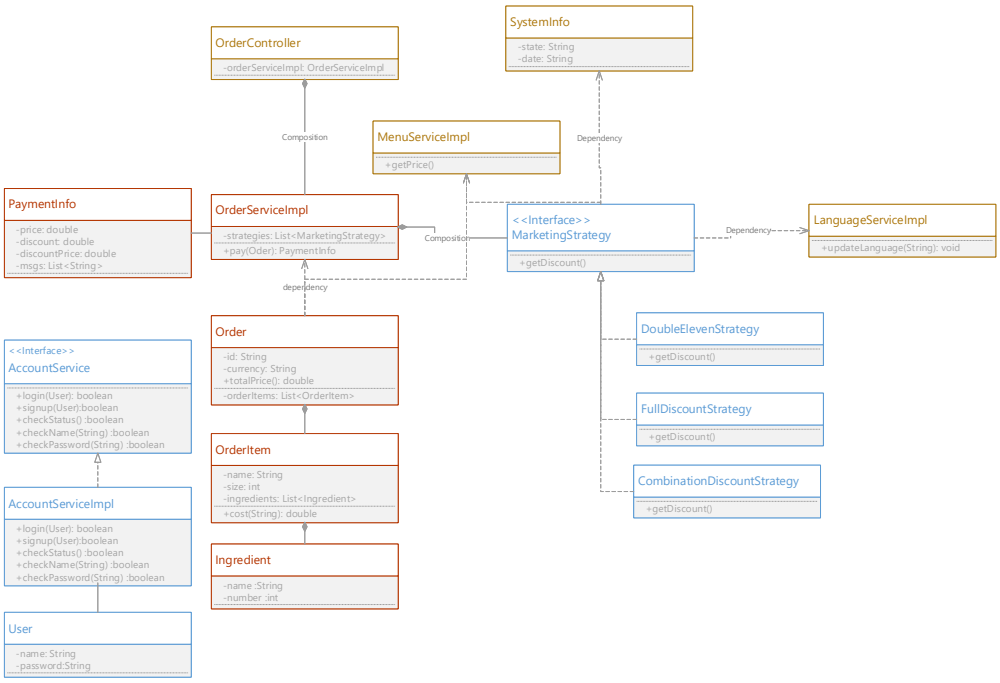

Figure 3.4: Overall UML Class Diagram

- **Order :** The DTO class is used to transfer data from server to client end. There are three attributes: id, currency and orderItems. Also, it has a public method to calculate the total charge of the order.
- **PaymentInfo :** The class is composed of all the return information of the order, including price, discount, discountPrice and messages.
- **OrderService :** The service class includes one attribute strategies and one public method pay.
- **MarketingStrategy :** The strategy class include one public method getDiscount. And all the sub strategy classes including DoubleElevenStrategy, FullDiscountStrategy and CombinationDiscountStrategy inherit it.
- **LanguageService :** The service class has one public method updateLanguage.
- **MenuService :** The service class has one public method getPrice according to different countries and regions.

Relationship between classes

As the UML diagram shows, the relationships in this system can be divided into three kinds: inheritance, composition and dependency.

Figure 3.5: Data Flow Level 0 Diagram

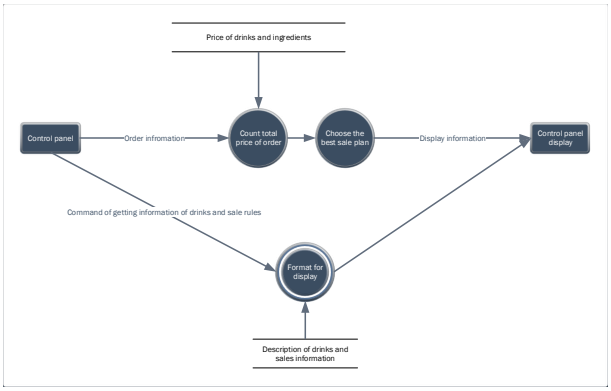

Figure 3.6: Data Flow Level 1 Diagram

3.2.3 Data Flow analysis and modeling

We apply the DFD(data flow diagram) to describe the flow of the input data and output data. We

level 0:

The top/level-0 diagram is an overall description of the whole system. The external entity are only control panel and control panel display. The data flow diagram for level 0 is shown as Figure 3.5.

level 1:

3.2.4 Behavior analysis and modeling

In this part, we apply the UML status diagram to conduct the behavior analysis and modeling.

Whole status transfer

Chapter 4

Performance Demands

4.1 Massive end user support

4.1.1 Detailed Description

There will be about 200 stores using this system at the same time. And the scale will be about 1000 people per day. The people using this system includes XXX.

4.1.2 Proposed measures

Since the system will be built on several servers, the strategies like load balancing can be used to handle the big amount of customers.

4.2 Stability over long period of time

4.2.1 Detailed Description

Since the store will be open 24 hours a day, the system must be running all the time and at least maintained evert half a year.

4.2.2 Proposed measures

This requires that the system should not stop or exit by accident at any time. We should handle all the possible accidents and catch the exceptions.

Chapter 5

Appendix

5.1 Demand interview outline

5.1.1 Constraints (5 minutes)

1. What is the number of servers?
2. Is there a regulation or limitation on the system running server-side operating system?
3. What are the rules or restrictions for the client device operating system (in the store)?
4. Is the salesperson's computer equipped with a screen for the customers?
5. Are there still any other special hardware conditions (supplementary)?
6. Does the company have specific legal restrictions?
7. The development time and delivery time points given?
8. Is there an intermediate time frame when partial project need to be examined?

5.1.2 Performance requirements (5 minutes)

1. Is 1,000 person/day referring to the salesperson or the customer (personal * month) ? What is the upper limit?
2. What is the number of the stores in the description "Support the usage of massive stores concurrently"? What is the average number of orders in the store? And what about the peak order number?
3. Order response time?
4. "Long-running and stable support system ": the requirements for a client front end and user-friendliness, cannot be unable to play due to any reason except network or hardware issues, cannot crash under massive clients, cannot have severe bug after online

5.1.3 Functional requirements(15 minutes)

1. Drinks, ingredients, cups in the beverage store supported by the system
2. Superposition and mutual exclusion of offers
3. Language, currency, pricing
4. Register & log in
5. Order versus Associate / independent with drinks, price, and description?
6. What the customer sees: the information returned in the order content and paymentInfo , including the beverage, original price, discounted price, discount information (whether it is necessary to specify the description of each beverage)
7. How the salesperson gets permission
8. Log information provided by the system
9. Process confirmation again

5.1.4 Possible modification of the code

1. Increase currency dollar
2. Salesperson access
3. Offer modification
4. View all aspects of drinks and ingredients (cup type, price) individual customer / salesperson interface

5.1.5 About the requirements document

1. “Give a corresponding map and explanation for each demand”

5.2 Organized interview records

5.3 Code change logs in response of new demands

Bibliography

[1] Wikipedia contributors. (2019, March 22). JUnit. In *Wikipedia, The Free Encyclopedia*. Retrieved 14:53, April 1, 2019, from `https://en.wikipedia.org/w/index.php?title=JUnit&oldid=888928403`