# Lab Report for Software Engineering course
# Lab 5: Demand Change and Prototype Development


| Wang, Chen | Liu, Jiaxing | Huang, Jiani | Tang, Xinyue |
|---|---|---|---|
| 16307110064 | 17302010049 | 17302010063 | 16307110476 |

School of Software
Fudan University


May 21, 2019

# Contents

# Chapter 1

# Demands of this lab

## 1.1 Requirements of this lab

According to the documentation of the lab assigner, this lab should satisfy the requirments in the following perspectives:

### 1.1.1 Documentation

In the documentation, we need to accomplish two parts, that is, we need to: Understand the needs of this experiment, complete the requirements document; Organize overall design documentation and detailed design documentation based on requirements.

### 1.1.2 Code Style

Before the experiment, our group members should agree on the code specification and the unified code style.

### 1.1.3 Project Management

The project is generally hosted on the Huawei Devcloud platform, on which platform we are going to finish the following tasks: Project management should be arranged based on the DevCloud platform, including adding work items, assigning work items, associating work items, managing work item status, and other project management functions;

Issues in the development process need to be recorded and managed on DevCloud, including defect reports in integrated development;

Collaborative development based on Git, submit project documentation and code on a team basis.

## 1.2 Background on the change of demand

Starbubucks's coffee shop has received a large amount of financing and decided to open branches in different countries and regions.

In order to adapt to the habits of users in different regions, it is necessary to develop a project prototype with basic functions, and then customize the system for different regions.Customized development needs to consider the following factors:

### 1.2.1 Drink Customization

At least 2 beverages need to be customized for local features (beverages defined in Lab4 are system default beverages).

### 1.2.2 Currency change

We need to switch to the corresponding currency according to different countries and regions, for example, a cappuccino in China 20CNY, a cappuccino in the US $3 (do not consider currency exchange).

### 1.2.3 Language change(localization)

The system needs to switch the system language to the language of the corresponding country and region according to the language of different countries and regions. The range of language switching is visible to all system users (sales personnel) and customers (beverage buyers). For example, when the system user inputs, the input content should be the language of the corresponding country and region; after the order is successful, the customer sees The transaction content should be switched to the language of the corresponding country and region.

### 1.2.4 Sales strategies

A good sales strategy can bring huge benefits, and the system's default sales strategy is the sales strategy defined in Lab4. Because sales strategies can change as external markets change, systems are required to add and remove sales strategies in a pluggable manner.

## 1.3 Specifications of the Lab

The following requirements are to be met for the custom development factors described in the background in the previous section:

### 1.3.1 Drink Customization

Customize two beverages with reference to the previous lab implementation.

### 1.3.2 Currency change

There may be multiple currencies in different countries and regions. For example, Hong Kong can circulate Hong Kong dollars and Chinese Yuan, mainland China circulates Chinese Yuan, and the United States circulates US dollars. The prices of different currencies for the same item are different depending on the value of the product itself. The system needs to specify the corresponding

price of the corresponding commodity for different currencies (the pricing of different currencies for each commodity, and the exchange rate pricing can be reasonably justified). Currently, the system only needs to support RMB and Hong Kong Dollar (HK$).

### 1.3.3 Language change(localization)

Different countries and regions may have multiple official languages, and the system needs to switch between different languages according to different countries and regions. The system must be able to switch between English and Simplified Chinese.

### 1.3.4 Sales strategies

The sales strategy is added and deleted in a pluggable manner. The sales strategy in lab4 is the default sales strategy and cannot be deleted. The source of information for all sales strategies is order and system information (date and location). On the basis of lab4, the requirements of (1)(2)(3) and the pluggable components of the sales strategy can be regarded as the prototype of the project.Based on this prototype, add the following sales strategy:

1. When the time zone of the system is November 11th, the total price of all commodities (including all prices) will be 50% off.

2. When the order contains at least one tea item and at least one item of the Coffee category, the total price (including all prices) is 15% off.

## 1.4 Additional precautions of this lab

### 1.4.1 Switch interface of currency and language

The switching of currency and language can depend on a “switch”, which is an abstraction rather than a concrete implementation, since the opening and closing of the switch may depend on different external conditions.

### 1.4.2 Prototype-based development

The addition and deletion of the sales strategy is based on the fact that the prototype of the project has been developed, and the sales strategy is added and deleted through the interface of the sales strategy provided by the prototype. When using the prototype, you can’t modify the code of any prototype, and you can only perform secondary development on the basis of the prototype.

### 1.4.3 Methods for switch and pluggable components

The development of ”switches” and pluggable components can refer to the relevant materials.

### 1.4.4 Freedom of change of the existing frameworks

In order to flexibly realize the four requirements in the previous sections above, it is encouraged that every team should design independently. The fields and method signatures in the framework code provided by the assistants in Lab4 can be modified. It should be emphasized that after the prototype is completed, it cannot be modified, and it can only be developed on the basis of the prototype.

# Chapter 2

# Division of work for this lab

# Chapter 3

# Analysis of the demands

# Chapter 4

# General design for the implementation

# Chapter 5

# Detailed design for the implementation

## 5.1 Switch Language Implementation

The switch of language will be mainly displayed in the user interface, so all the information that need to be multi-translated will be separately placed into different constant files. In this iteration of implementation, we only instantiate the Chinese and English versions. And the correspondent service classes will use a typical mechanism called reflection to implement the switch of different constant language files.

In the following two sections, the detailed design of constant files and language service classes will be respectively clarified.

### 5.1.1 Language Constant Files

The two constant files are positioned in the constant package. In these two files, all the variables (there are no methods) are qualified with public static final String since they are all constant strings.

To our attention, all the necessary variables should have the same names in all the language files to maintain the availability of reflection.

### 5.1.2 Language Service Classes

To follow the idea of prototype development, all the concrete service classes should implement their corresponding interfaces. The interface make it clear what the service will implement and its parameters. The most notable design pattern in this class is single-instance pattern.

## 5.2 Switch Currency Implementation

### 5.2.1 Currency Property Files

### 5.2.2 Currency Service Classes

# Chapter 6

# Problems encountered in this project

# Chapter 7

# Measures against demand change

# Chapter 8

# Tools and literature involved in this project

# Chapter 9

# Conclusion for the process of accomplishing this project

# 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`