

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

Wang, Chen	Liu, Jiaying	Huang, Jiani	Tang, Xinyue
16307110064	17302010049	17302010063	16307110476

School of Software  
Fudan University

May 20, 2019

# Contents

<b>1</b>	<b>Demands of this lab</b>	<b>2</b>
<b>2</b>	<b>Division of work for this lab</b>	<b>3</b>
<b>3</b>	<b>Analysis of the demands</b>	<b>4</b>
<b>4</b>	<b>General design for the implementation</b>	<b>5</b>
<b>5</b>	<b>Detailed design for the implementation</b>	<b>6</b>
5.1	switch language implementation . . . . .	6
5.2	switch currency implementation . . . . .	6
<b>6</b>	<b>Problems encountered in this project</b>	<b>7</b>
<b>7</b>	<b>Measures against demand change</b>	<b>8</b>
<b>8</b>	<b>Tools and literature involved in this project</b>	<b>9</b>
<b>9</b>	<b>Conclusion for the process of accomplishing this project</b>	<b>10</b>

## Chapter 1

# Demands of this lab

## 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>