



Report

Assignment 2 *- Analysis, Design and Implementation*



12 Feb, 2017

Semester: Spring 2017
Course: Software Technology
Author: Sarpreet Singh Buttar

Contents

Task 1 Analysis	1
Subtask A Identifying Use Cases	1
Reflection	1
Subtask B Robustness Diagrams	1
Reflection	1
Subtask C Use Case Realization	1
Reflection	1
 Task 2 Design	 1
Reflection	1
 Task 3 Implementation	 2
Reflection	2

Task 1 Analysis

Subtask A Identifying Use Cases

Reflection

Identifying use cases was quite easy for me because one of the use case I already implemented in first assignment and second one I got it from this assignment. However, creating a nice picture of it took some time but I managed to finish it before the planned time. On the other hand, documenting the use case was bit harder than I thought because I was not aware of the use case structure. Some of the content was already mentioned in the description like pre/post condition, primary scenario etc but I was bit confused with initial point such as should I mentioned it in the main scenario or make a separate box for it. At last, I Google it and found some appropriate solution of this. In addition, I would also like to reflect on activity diagram because it was my first interaction with it which brought some misconception in my mind. Somehow, I managed to make it but I have one question regarding it; should I make an action box when user click button or simply write this along the arrow/line?

Subtask B Robustness Diagrams

Reflection

Creating robustness diagrams was bit challenging for me which I was not even aware before. First, I spent some time to understand its concept by watching the lectures. The most difficult part for me to draw in this robustness diagrams was alternative paths. In 'view a list of books' use case, I have alternative path known as when no book is available in the system. I have no idea how to draw alternative paths in robustness diagrams. So I made assumption that there is at least one book in the system. In addition, lecture also does not discuss about making alternative paths or maybe I missed something. I would like to know if creating alternate paths is a part of robustness diagram?

Subtask C Use Case Realization

Reflection

I think creating sequence diagram was quite easy because now I have understand the use cases very well as I have done the activity diagram and robustness diagrams of each use cases in previous tasks. In addition, I would say that the 'List books' took bit more time than 'Delete book' because I have not implemented 'Delete book' yet so I have only rough idea which means I cannot explain it in detail as I explained the 'List book' sequence diagram. So far, I think creating sequence diagram from implementation is more easy rather than other way around. I will find this out in task 3.

Task 2 Design

Reflection

In this task, I easily made two UML diagrams. One sequence diagram which describes the logic for fetching the books from XML file and then converting into objects and send to server as a JSON objects. This diagram describe all the steps which I will take in the task 3. The other diagram is a class diagram and this diagram only explain the relationship of the classes with each other. I really do not need this diagram for implementation because sequence diagram is enough but I still made it because I want to practice UML diagrams. However this diagram gives an overview of the system without looking the implementation. I will update this diagram after finishing the next task by adding the 'RemoveBookResource' class.

Task 3 Implementation

Reflection

Implementing fetching books took more time than I expected because I encounter with one problem. In the XML file the attribute 'Publish Date' is written as 'publish_Date' and in my book class it is written as 'publishDate'. It took some time to figure out this problem. In addition, all the implementation went same as I planned in previous task. While implementing I realised that I can do all these steps in just one line in the 'GetBooksResources' class which was quite reward for me. On the other hand, implementing delete book was also easy. I encountered with two small problems, first I am not familiar with annotation inside the method parameter which took some time to figure it out. Second, I was bit confused whether to made a void method or return Response to server according to API. At last I decided to follow API and return appropriate response. Regarding the implementation first and design later, I do not think that there is a much difference to make sequence diagram first and implement later or vice versa because in sequence diagram we have to describe all the details so its much like coding.