Personal Reflections

Subtask 1.A:

I decided to identify the use case for “add book” and “delete book” cases and document them. Firstly, by identify the actors, identify the start and the end point for the use case and descried the functionality that helps actors achieve their goals by using activity diagram. It wasn’t that hard to end this subtask and I looked at the lectures and got some information from the internet. It gave me a good experience in how to draw activity diagrams and identify the alternative flows.

Subtask 1.B:

I made robustness diagrams for the use case in subtask 1.A. It is harder than the activity diagram. As I understood, the boundary elements are the system elements that the actor interact with, control elements are the steps that the system takes to achieve the actor requirement and reach him goal and entity elements are the classes that will be changed as the end point of the use case scenario. It was the hardest diagram to draw and to understand and I don't know if I made them in a right way.

Subtask 1.C:

Use case realization represents how a use case will be implemented in the system more in detail. I used the sequence diagrams to illustrate the interaction between classes in the use case. The first use case “List books” is the call of books that I implemented in the previous assignment was manageable but I don't know if the “delete book” one is right while I don't know how to implement its functionality. Anyway, I tried to do it as I understand with some help from the internet.

Task 2:

To describe the solution for designing this function I used the sequence diagram. I show the classes that are participated in the implementation process (the main screen, Getbooksresource, bookList, the XML file and book) when the user clicks the book list button, it will call the book list that is created as a bookList object contains the books

that are found in the XML file and converted to book objects and will finally translate this books object to JSON to be shown on the main screen.

Task 3:

For this task, it was hard because working on reading XML file and converting it to objects was new to me. I spent about 7 hours to complete it. Half the time was reloading vagrant. Coding with the compiler as eclipse is very easy because it detects the mistakes automatically but here I takes about a minute in every vagrant reloading and then you discover that you forget a semicolon, but any way I could do it with helps from the Q/A online sessions and with helps from friends. I tried to work as the design I wrote in the previous task and then I changed some steps that were does not work in the implementation. And for the delete book I delete the book from the book list and then marshalling the existed book in the list to rewrite the XML file so the main screen will get the books from the XML file without the deleted book.