ECS505U Software Engineering

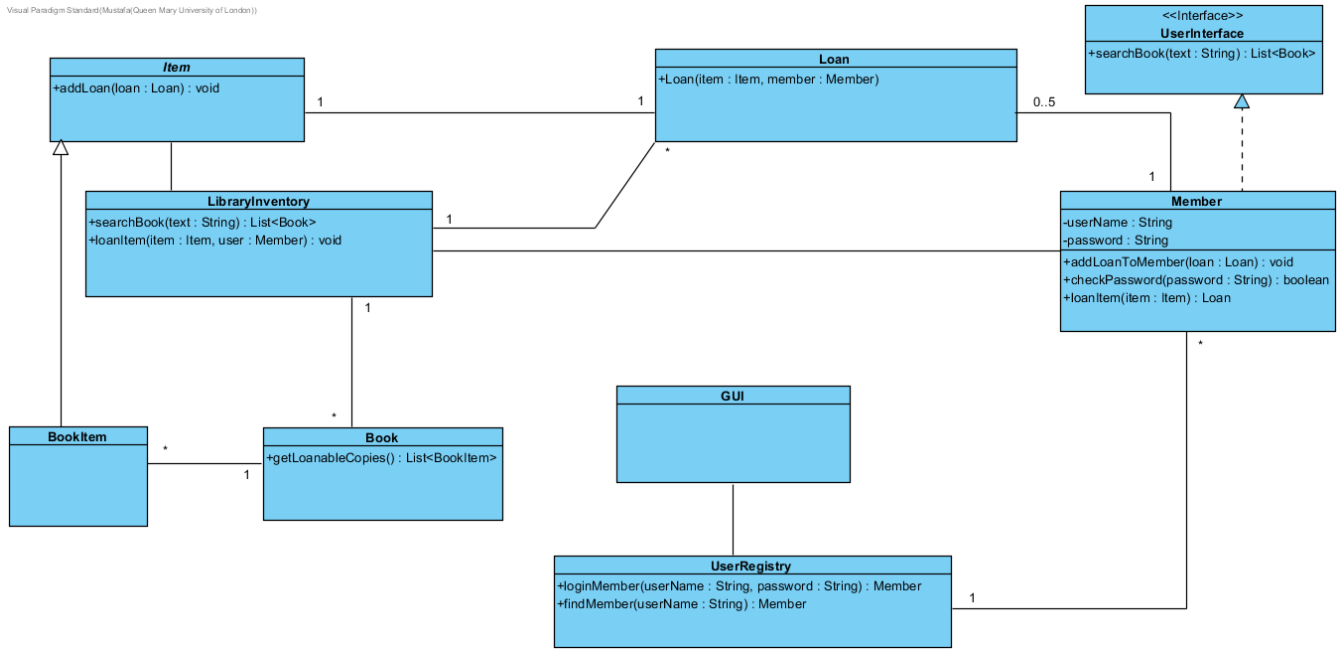
Laboratory Exercise (Assessed, 2.5% of final mark)

Creating UML Sequence Diagrams

Version 1.4, 14/11/2017

1. Prerequisites

- You should have access to Visual Paradigm. For campus students, version 14.1 is pre-installed on all machines. For non-campus students, check the QMplus course page for download and licensing information.
- You should watch the video titled "UML Sequence diagram video tutorial" available on QMplus course page.



In this exercise, you are expected to create two sequence diagrams for two use cases we described in our previous lab exercises. You are expected to use the class diagram above which is included in the visual paradigm project file (the file is available on the QMplus course page below the lab exercise instructions). Draw your sequence diagrams using the provided project file.

In the use case descriptions, some of the possible steps are skipped and others are shortened (as Apple always say in their commercials**) to make the diagrams simpler. There is no need to query as to why some steps are not there. Please do not speculate into how those steps should have been and draw your sequence diagrams based on the descriptions provided.**

2. Login Use Case (this use case is skipped in the first exercise)

In this exercise, we will describe the interactions among objects when a member performs login (use case). We assume that the following objects are already in the system: GUI (is the object representing graphical user interface), userRegistry (a UserRegistry object) and member (a Member object representing the system user).

1. A member makes a **login request** using the GUI providing his/her username and password.
2. GUI calls the registry for login with the give username and password.
3. Registry finds the member object with the given username (Skipping the possibility of wrong username).
4. Registry calls the member object to check if the given password is correct and assign its result to a local variable called login result.
5. If the password is correct registry returns the member object otherwise returns null (to the GUI).
6. GUI **display**s **login result**. (Ignoring redirecting to account page or back to login if password is incorrect)

You can name the interactions between GUI and the actor with the bold text.

3. Loan Use Case

In this exercise, we will describe the interactions among objects when a member performs loan (use case). We assume the following objects are already in the system: GUI (is the object representing graphical user interface), inventory (an LibraryInventory object), book (a Book object), bookItem (a BookItem object) and member (a Member object representing the system user).

1. A member (already logged-in) makes a **search request** for books using the GUI.
2. The GUI send search request to the member object.
3. The member object calls inventory to search for book.
4. Inventory returns list of books (List<Book>) to the member object.
5. Member objects returns the list to the GUI.
6. For all the books (from the previous step) GUI gets all loanable copies. (Skipping the step in which books check all book items to find out loanable ones)
7. The GUI **display**s **search results** to the member.
8. The member makes a **loan request** by selecting one of the books from search result (using the GUI).
9. The GUI calls member object to loan selected book (item).
10. The member object sends loan request to the inventory for the selected item.
11. The inventory creates a loan object for the item and member. (constructor call)
12. The inventory adds the loan to both the book item and the member object (to book item first).
13. The member object returns the loan object to the GUI.
14. The GUI **display**s **loan result**.

You can name the interactions between GUI and the actor with the bold text.