# Case Studies in Software Design

**Subtask One:** The Library System

**Due Date:** 15.00 on Monday 22nd October 2018

**Marks Available:** 5% of the overall coursework mark

The aim of this subtask is to refresh your object-oriented programming skills in Java and their relationship with UML using a simple example. The skills you refresh here will be very useful when it comes to the second and third subtasks where you will be designing and implementing a larger-scale object-oriented design. This subtask is an individual piece of work.

## The Library System

Our library system is primarily focused around books and members. A library holds a set of books where each book has a record of information associated with it and includes author, title and accession number (which is a unique identifying number across all books in the system). The members of a library can borrow books and again have a record of information stored about them. The specific attributes for these two types of objects are given in Figure 1 which provides the initial class diagram design for the system.

In our simple model, members can borrow up to three books at any one time and can keep them for an unlimited period of time; loans do not have a due date. Furthermore, this version is to run locally and in a standalone manner thus data is to be retrieved locally from disk, kept in memory as objects during execution and written back to disk when the application exits. You are required to use Java serialisation to persist the data.

In addition to the book and member objects in the system, there are three other main classes, including the Graphical User Interface (GUI), as illustrated in the class diagram of Figure 1. The class diagram provides an initial design, however you may need to build upon this design during implementation. You do not need to submit a revised class diagram for this assignment, but you may be asked to justify any changes you made during your walkthrough.

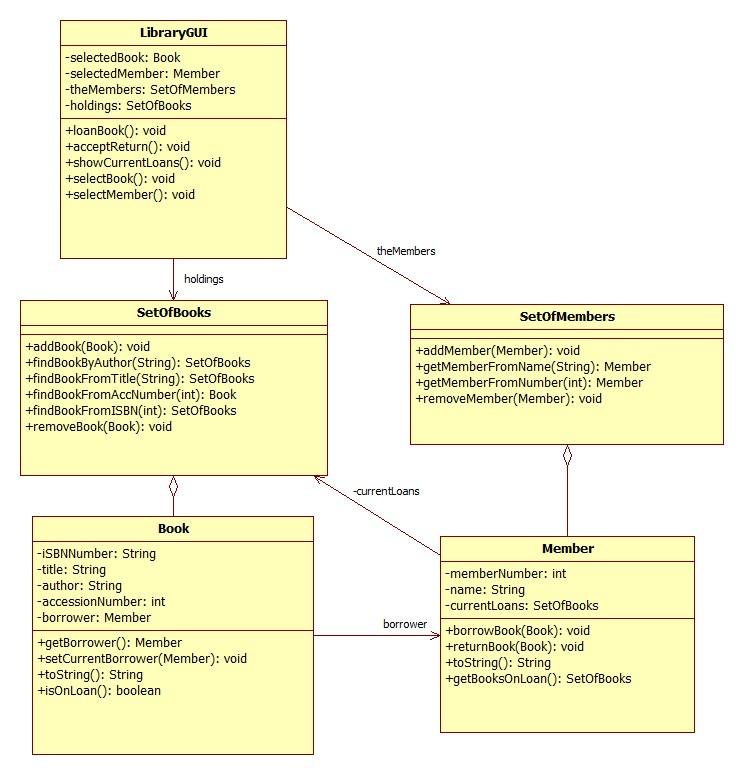
From the GUI, your application should allow a user to perform the following tasks:

* Issue a book loan to a member (up to a maximum of three)
* Accept the return of a book by a member
* Find all the books that a member currently has on loan
* Show all the books that are currently not on loan (could be part of the issue process)
* Add a book into the library catalogue
* Add a member into the library system

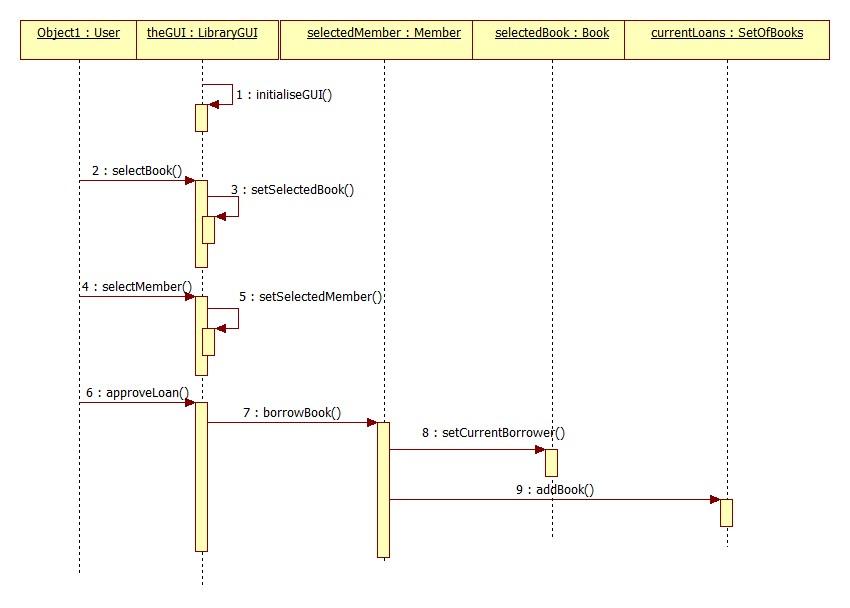
Your system should also persist the data to disk using Java serialisation; you might therefore wish to include an explicit close button on your GUI to capture the closing event.

A typical sequence diagram for issuing a loan to a member is given in Figure 2. Like the class diagram, the sequence diagram is an initial design and its ultimate implementation may require you to adapt it somewhat.

You are provided with a skeleton Library System in Java as a NetBeans project to get you started, however only a limited number of attributes and methods are currently implemented; you will need to complete the implementation using the briefings and designs given in this document.



**Figure 1:** Library System Class Diagram



**Figure 2:** Library System Sequence Diagram for issuing a book loan to a member

## Marking Process and Submission

In this subtask, you are not going to be directly assessed on the appearance of your GUIs. However, you can pick up additional marks by implementing extra features such as searching for books and/or members or delivering an easy to use GUI. See the marking scheme in Table 1 for further details about what we are looking for.

You should upload your complete Java project as a single ZIP or 7z file to the module Blackboard site via the subtask one assignment upload link. Remember to include all your source code in the compressed file.

The deadline for electronic submission is **15:00 Monday 22nd October 2018**.

We shall be grading your work and providing feedback via demonstration. Each demonstration will be approximately 10 minutes and will be held during the tutorial sessions for the week commencing Monday 22nd October 2018.

|  |  |  |
| --- | --- | --- |
| **Feature** | **Mark** | **Works?** |
| Code runs without problems and provides a working GUI showing books and members | 10 | Y / N |
| Show all books not currently on loan | 10 | Y / N |
| Show all members of the library | 10 | Y / N |
| Issue a book loan to a member | 10 | Y / N |
| Accept the return of a book by a member | 10 | Y / N |
| Show all the books that a member currently has on loan | 10 | Y / N |
| Member can borrow up to a maximum of three books | 10 | Y / N |
| Use Java serialisation to persist the data | 10 | Y / N |
| Add a book into the library catalogue / Add a member into the library system | 10 | Y / N |
| Extras - there are various things you could do to gain this final half mark such as:   * Good use of Java GUI * Can search for books / members   *You don’t need to get extravagant in your extras unless you really wish, although anything you do produce needs to be on par with the basic features listed above.* | 10 | Y / N  Remove member  Remove book |
| Total marks out of 100% |  |  |

**Table 1:** Marking scheme. Each yes counts 10% of the marks available for this work