

Guidelines for Final Project

Deadlines: Initial Design (3%): Due 4/13, Final Project (7%): Due 4/27

This is a group project (maximum 4 member teams)

Objective:

Design, develop and implement a working software system in Java. All code must be maintained on **GitHub** (See instructions on Canvas)

Problem Statement – **Online Library**

Congrats! You're hired to create a project that will mimic Online Library system. The Students should be able to browse through the catalogue, borrow books online and manage their transactions. The projects will have two perspectives as follows:

- **Student's perspective:** The student should initially be provided with login/register options. Once the user logs in, the user will be provided with menus to select from. The user should be able to look up the book using the author name, book name and ISBN number. The availability of the book should be displayed to the user. If a specific book selected by the user is not available, then the date of availability (which is the return date of the previous lender) should be displayed to the user.

The user should be able to lend a book and should be provided with options on standard lending durations to select from. Also, a unique lending ID should be generated which will be used by the user while returning the book. When the user decides to return the book, then the current system date should be obtained to check if the user returned the book in time. If the user is late, then a penalty should be added to the user account. The user should not be allowed to borrow another book without paying the penalties.

The user should be able to look at the list of books lent from library and the list should be sorted based on the date of return.

- **Librarian's perspective:** The librarian should be able to view the status of all the books that have been lent so far. Also, the librarian should be able to add, delete and modify the information of any book. The updates should be consistent and should be reflected throughout the software system. For instance: If a specific book has been lent to a student, then the librarian cannot delete the book until the student has returned it.

The software development cycle includes requirement specification, design and analysis, coding and testing phases. You will also prepare a report documenting each step used in the software design cycle and will use UML diagrams to properly represent all classes used in your design. (You may use free software called StarUML which can be downloaded from <http://staruml.sourceforge.net/en/download.php>)

System Features:

The Online Library System that you are designing offers a variety of design choices. You are encouraged to design the software system that provides the most realistic user experience. Here are some of the required functionalities:

- Students should be able to login to their accounts using a unique password with the Online Library System.
- Make your system user friendly by providing sufficient guidelines and help to use the system (For example: if your system is expecting any input in a specific format, be sure to specify that in the instructions as well as the Testing part of your report)
- **Note:** Your Library system must have **state persistence** by saving the necessary data to files. State persistence allows your system to outlive the process that created it. For example, if you add a few books and/or users (students) to the system when you initially run it and then close the execution before running the system again, the system must be able to remember the previous books and/or users and its state. This will allow you to use the previously entered information without having to create student accounts and table information, etc. again and again. You can complete the entire project by storing data in files or any other persistent storage medium.

You have tremendous amount of freedom in designing this system. You are welcome to add any additional functionality to your library system. Just imagine that you are an actual student/librarian at a school/city.

Note: The design of a Graphical User Interface is optional. You can also develop a command-line based system

System Design & Development:

The final project includes two major components: Project Design and Implementation.

Design (3%): The design component will consist of the software design for the project and will include the list of all classes (member variables and functions) to be used, their relationships & collaboration, as well as databases/files. The initial design must precede any implementation. You will start by analyzing the requirements of the project and by identifying the classes required along with attributes and functionalities. A doc/docx document listing the design along with UML diagrams (Class diagrams, activity diagrams, etc.) must be submitted by **April 13th (1159pm) on Canvas**. Please include the following:

- Skeleton Code with all classes (with member attributes and member functions)
- What are class hierarchies and relationships? (in your report)
- All other data structures or files to be used (in your report)
- Peer evaluations (Groups **and** individual) are REQUIRED
- Also setup a private repository for your source code on **github** and add all instructors as collaborators (See instructions on Canvas).

It would be beneficial for you to analyze your system thoroughly and provide detailed UML diagrams. Here are some of the questions, answering those will help you put on the good path:

- How many classes will you have and how will they interact?
- How will you store student/librarian ID, passwords, menu options?
- How will you represent the book selection options and the menu?
- How will you deal with adding multiple students to a library system?
- How will the borrowing system be used to handle a transaction?

Implementation (7%): Once you complete the initial design, setup a *private* github repository and start implementing the student, librarian, and library functionalities separately. Test these separately with individual drivers. Be sure to add the appropriate user-friendly menu options that would allow the user to select various actions easily. Ensure that the menu options only work when correct input is provided by the user. Test each option individually to ensure that all available functionality is properly implemented. **All sourcecode must be shared amongst team members and instructors via github.**

Now imagine yourself as the user of the system (as Student, Librarian) and observe the behavior of the system (and see how it changes). Reflect on the efficiency of your choices. Review your initial design and make the necessary changes to remove any unnecessary attributes and functions. You may have to add some more functionality that you might have missed in your initial design. Implement the updated design and report the updates to the initial design along with a discussion regarding the efficiency of your choices. A doc/docx document listing the final design including UML diagrams must be submitted along with working Source Code and video demo by **April 27th (1159pm)**. See the **checklist** given below.

Testing: Develop and build unit tests for each operator. Create different drivers for testing each component of the system as a Student, Librarian separately and then integrate the entire system and test with a separate main driver. Please ensure that the complete system is properly tested before submission.

Deliverables:

The following items must be submitted:

- **All source and header files** related to the system implementation (on Canvas as well as github).
- **All drivers** that are part of the testing as Librarian, and Student separated in different folders
- A **README** file with any information regarding compilation and testing I need to know in order to successfully compile and run your system. Include **any other files** needed to compile or test
- **Source code must be properly organized, readable, and must use proper best coding practices.**
- **The report file** with the final design with discussion, and details of testing activities. Feel free to show additional testing you performed in the report
- Video Demo (10 minutes long max) showing all features of your working system
- Peer evaluations (Groups **and** individual) are **REQUIRED**

Project Checklist:

You can use the following checklist to ensure that you have submitted everything required for the project:

- Did you submit evidence of successful compilation or testing (screen captures)?
- Did you submit the instructions required to compile your code?
- Did you add all instructors as collaborators to your private github repository?
- Did you submit all files including any new header files used for compilation on Canvas (compressed as one file)?
- Did you implement user's features: password, unique ID, Order Entry, etc.?
- Did you submit a report with your designs (and UML diagrams), evidence of successful testing of all features?