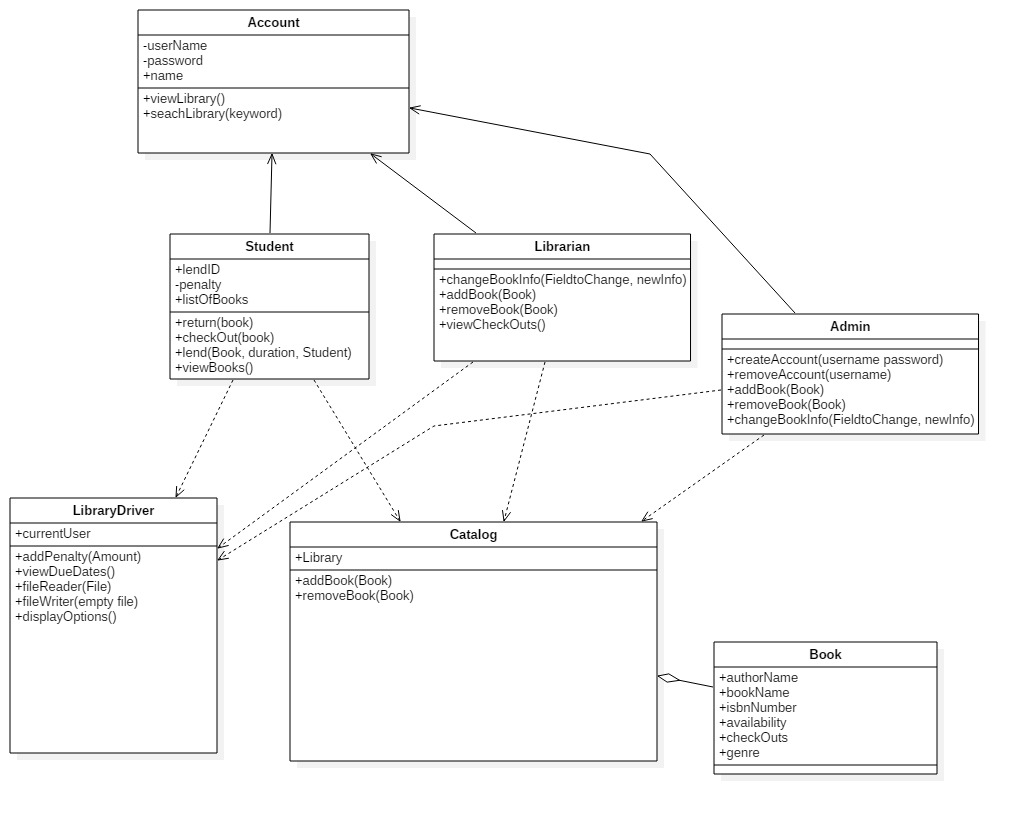
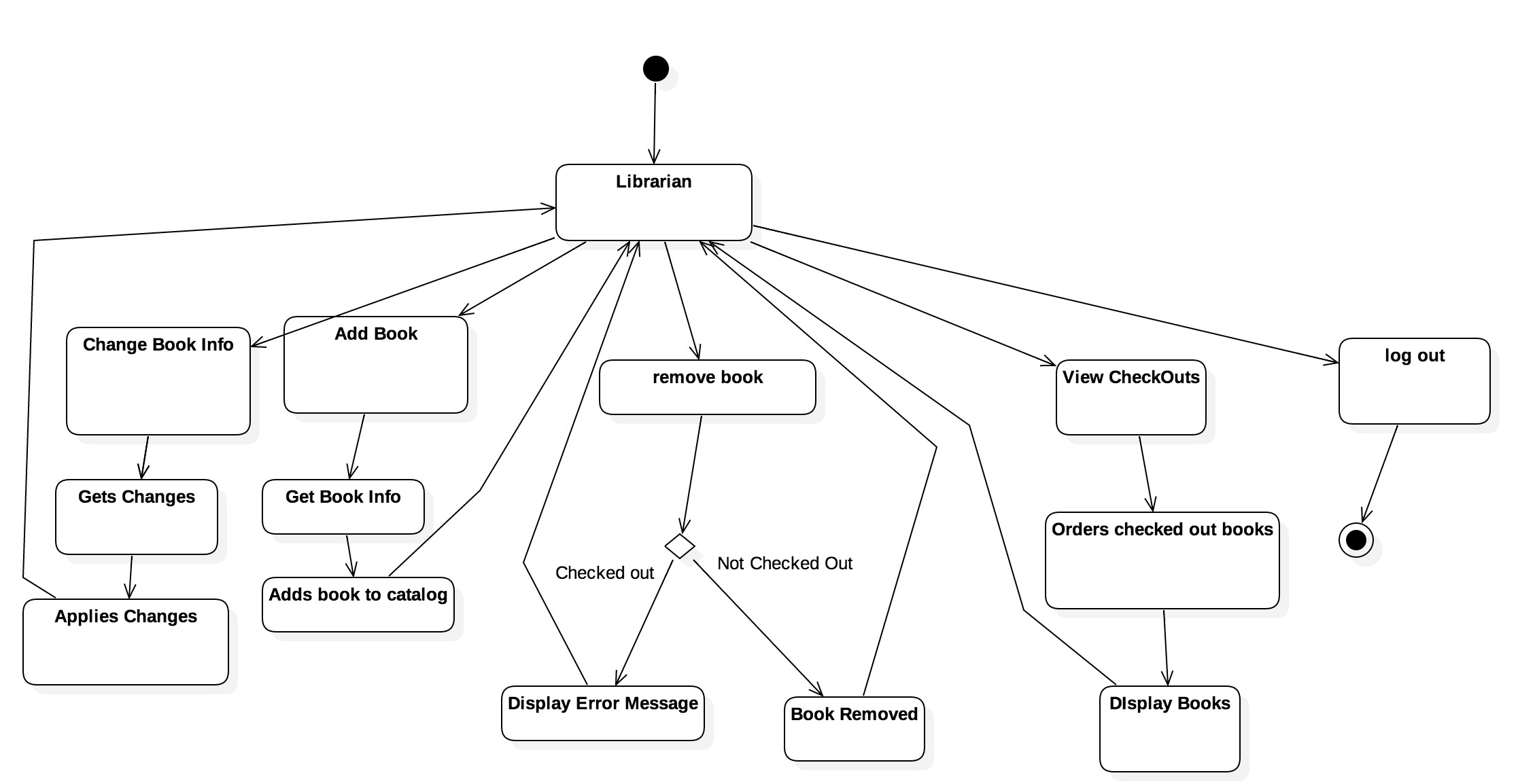
C212 Spring 2018 Final Project Part 1 Group 23

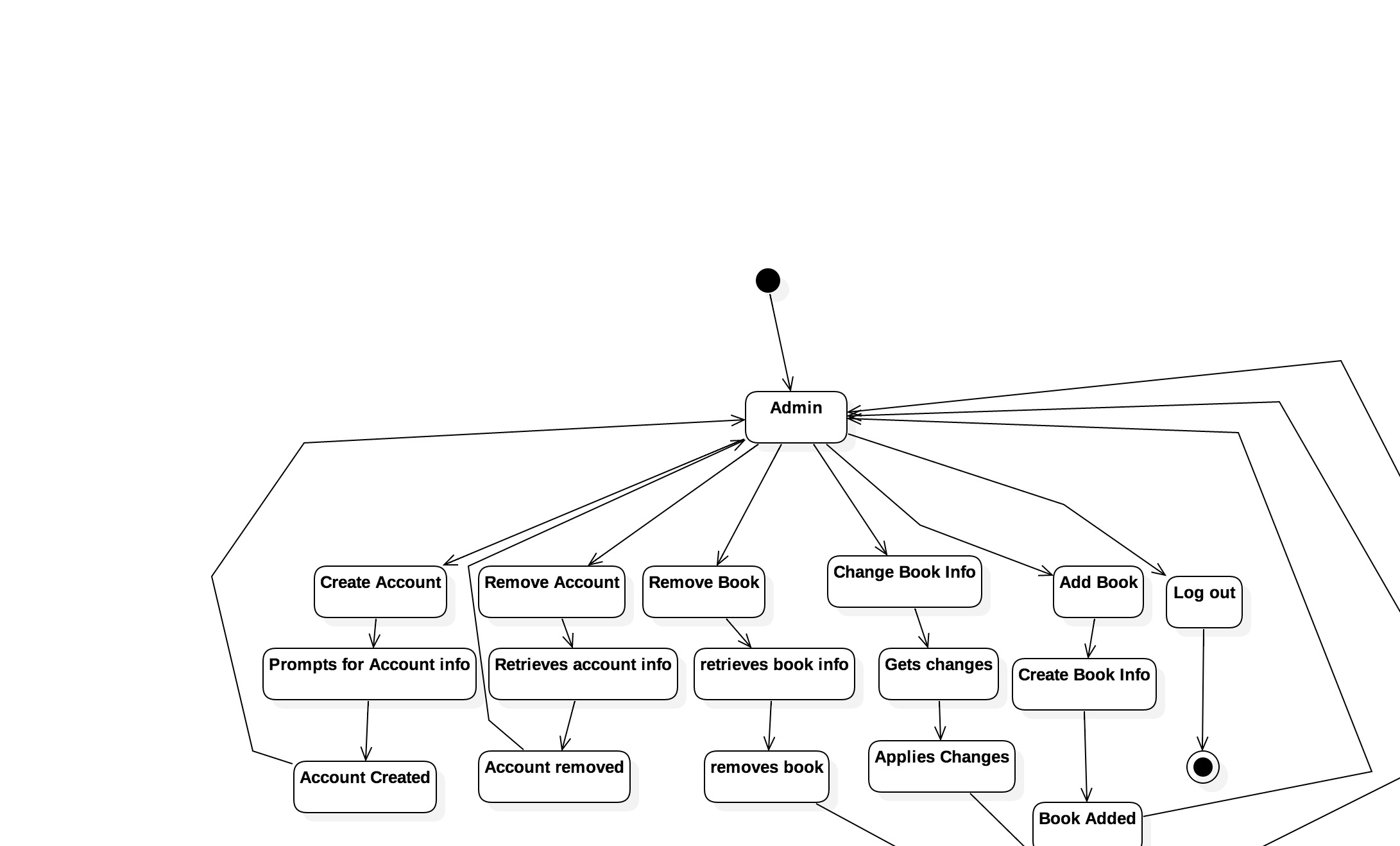
13 April 2018

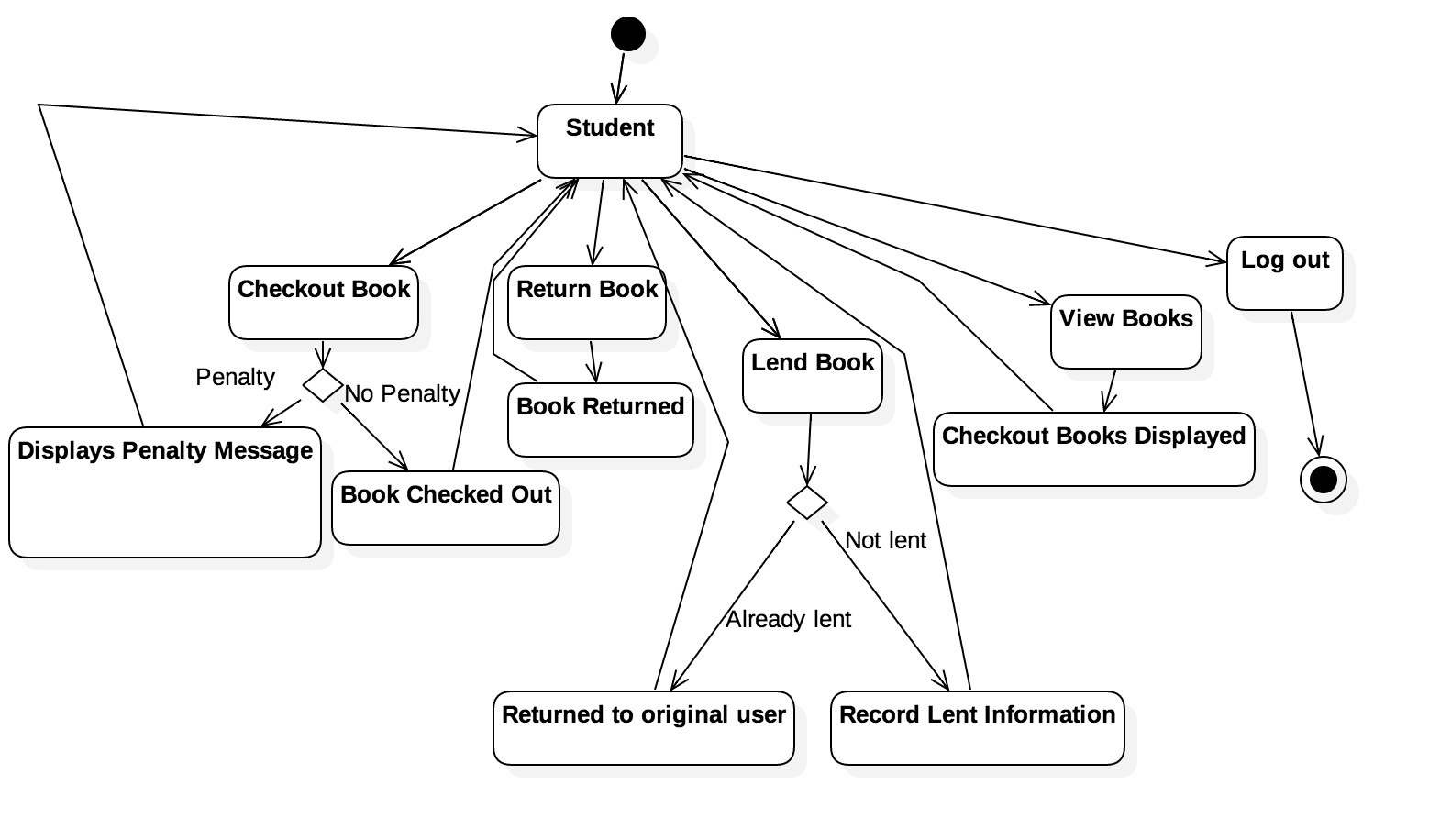
All diagrams are available on our GitHub repository. (<https://github.iu.edu/isherfic/finalProjectC212>)

**Class diagram:**



**Librarian activity diagram:**

**Admin activity diagram:**

**Student activity diagram:**

**Classes:**

Account: The parent class of Student, Librarian, and Admin. Will never be instantiated, just serves the purpose of reusing code that will be shared between the three different kinds of Accounts.

Student: A Student Account. Will be able to check out, return, and lend books. Also has a penalty instance variable that tracks the Student’s late fees.

Librarian: A Librarian Account. Will be able to modify book information, add books to the system, remove books from the system, and view the status of all checked out books.

Admin: An Account with the ability to modify all data in the system. Has the ability to add other accounts to the system.

LibraryDriver: The class that will contain the main loop of the program. It will read state information on startup, prompt the user to log in, and save state information when the user exits the program. In addition to this, it will display options to the user and execute the user’s choice by interacting with the other classes in the system.

Catalog: Container class for Books. Will contain methods for searching the Catalog and different methods for displaying the Catalog of books to the user in a variety of options.

Book: This class will contain all pertinent information about a Book. Author, title, who’s checked it out, when it’s due, etc.

**Class heirarchies and relationships:**

Student, Librarian, and Admin all inherit from Account. Book aggregates Catalog. LibraryDriver uses all other classes and drives the entire system.

**Other files to be used:**

We’re going to have plaintext files for information on all Accounts, Books, and the Catalog. This will make the system state-persistent. We also plan on encrypting all user passwords using a simple encryption method.

**Skeleton code:**

Skeleton code can also be found in our repository under the Skeleton\_Code folder. It contains basic Java skeleton code and simple comments for instance variables and method declarations.

**Group evaluation:**

Group 23 Peer Evaluation

***Note****: Two versions of peer evaluations are expected to be submitted. (1) Each group must agree on the effort evaluation based on actual contributions. This is to be submitted along with other group deliverables. (2) Each group member must also submit their own evaluation of group effort individually. Ideally we would hope that both the group and individual evaluations are the same and each group member has contributed equally*.

|  |  |
| --- | --- |
| **General Information** | |
| Group Number: | 23 |
| Who is completing this? | Isaiah Sherfick |

|  |  |  |
| --- | --- | --- |
| **Overall Group Effort Evaluation** | | |
| If you were rewarded $21 for your hard work and had to distribute this among all teammates, how much would you give to each member for the actual work done? | | |
| Group Member Name | Amount | Justification (**Required**) |
| *Lexi Floom* | $5.25 | Showed up to every meeting and did a fair amount of work |
| *Trevor Dunbar* | $5.25 | Showed up to every meeting and did a fair amount of work |
| *Isaiah Sherfick* | $5.25 | Showed up to every meeting and did a fair amount of work |
| *Gary Baker* | $5.25 | Showed up to every meeting and did a fair amount of work |
| Additional Comments (Optional): | | |
| **Supplementary Remarks** | | |
|  | | |