Jerry Hong and Akshay Reddy

Library System & GUI Report

Our interactive Library System code consisted of 3 driver classes, as well as 8 other classes associated with displaying our code output as a GUI. The three driver classes were: Resource, Readers, and Library. The readers class essentially consisted of methods for a user to register himself or herself into the library system. Based on the rules of what a password should have, as well as usernames not being duplicated, a user would be able to register successfully. Then the readers class also incorporated methods to check if the username and password were right when the user logs in. The main purpose of the resource class, which was an abstract class, was to create all of the necessary variables related to the book information, in addition to creating daysLeft variables that would randomly generate a number between 1 and 31 to indicate the number of days left until an item has to be returned. Finally, the Library class generated the database of resources, with all of the relevant information (name, type, ISBN, SearchID, number of copies available, and readersList). The Library class had the main method, while also implementing two central methods: getBook() and findResource(). Given the user inputs the title of the resource, the getBook() method will check how many copies are available. It first matches the input to the title, and it will then either report the number of books available or return -1 if they are all borrowed. If the input does not match to the title, then the method returns -2. The findResource() method, which is displayed in the GUI, uses the getBook() method and prints out a message about how many copies of the specific item are currently available.

The other 8 classes are primarily associated with the GUI. The BorrowBook, Login, Register, FindResource, Results, and ReaderInfo are all individual GUI interfaces with distinct features including text fields input, buttons, and tables, which were all put in their individual JFrames. The buttons all utilize an ActionListener in order to register a response once the user clicks. All the output messages from the driver classes use JLabels for the GUI. The Login and Register classes work together by passing along the username and password info into a hashmap. Anytime the user fails to register because of invalid password or fails to login because user info does not match, the text fields clear out for them to re-input. Once the user is in the Login, the hashmap contains the info successfully registered, and the user must type exactly as is or else the log in fails. Once the login is successful, the Login page disappears and the HomePage opens. Each of the buttons (Reader Info, Borrow Book, Find Resource) all redirect to their respective

GUI page for the user to interact. There is also a Return button if the user wants to return to the homepage. The FindResource GUI features a search field where the user wants to search for a particular title. If the title or searchID matches any items within our database while also having more than 0 copies, it will return the title and the number of copies available. If not, it will specify that the resource is not available. The way the Results page is made is to first create a JTable that has a container and JScrollPane for that JTable which stores the tableModel. A for loop iterates through the item ArrayList and adds each item to the tableModel with the corresponding values. This is done by assigning the corresponding variable with the respective get() methods and storing them in an array. This array will then be added to the tableModel. The Results will output only if the user search input matches (and has > 0 copies) when they are in the BorrowBook page. In addition, the system will say they successfully checked out the item. Finally, the Readers Info page displays the user's information (i.e., username, password, item(s) checked out, days remaining). The table is a similar method used to create the Results page.