User instructions for librarians

The GUI has five tabs, namely: Book Search, Check Out, Borrower, Check In and Fines.

- In **Book Search** tab, the librarian can enter in the text fields any combination of Book ID, Title, and/or either Author Name or any combination of author's firstname, middlename, and/or lastname. On clicking the Search button, the following are displayed in a table: BookID, Branch ID, Title, Author name, Number of copies at each branch as Total, and Number of available copies at each branch as Available.
- If Search button is clicked without entering any values in the text fields, error message is displayed.

- In the **Check out** tab, the librarian should enter in the text fields a Book ID, a Branch ID and a Card Number. All values must be valid.
- If Check out button is clicked without entering any values in the text fields, error message is displayed.
- Otherwise, clicking the Check out button creates a new tuple in book_loans table with a new unique value for the primary key loan_id, and a message is displayed. The date_out is the current date and the due_date is 14 days after date_out.
- If a borrower already has 3 book_loans, then the checkout fails and returns a useful error message.
- If the number of book_loans for a given book at a branch already equals the no_of_copies at that library_branch, then the checkout fails and returns a useful error message.

- In the **Borrower** tab, the librarian can enter in the text fields any combination of the required details and search if the borrower exists by clicking the Search button.
- If Search button is clicked without entering any values in the text fields, error message is displayed.
- If the borrower does not exist, a message is displayed.
- If the borrower exists, a table containing the borrower’s Card Number, First Name, Last Name, Address and Phone, is displayed.
- A new borrower with a new unique value for the primary key card_no, can be created in the system by entering all name and address values and then clicking the Create new button. A new tuple is created in the borrower table and a message is displayed
- If any name or address value is null, the borrower can not be created and an error message is displayed.
- If a new borrower is attempted with an existing borrower’s name and address values, then the system rejects the creation of the borrower and returns a useful error message.

- In the **Check in** tab, the librarian must enter the Card Number and/or any combination of the other required details.
- If the Show loaned book(s) button is clicked without entering any values in the text fields, error message is displayed.
- Otherwise, on clicking the Show Loaned Book(s) button, if an invalid card number was entered or, if there are no loaned books with entered details, an error message is displayed.
- Otherwise, on clicking the Show Loaned Book(s) button, the following are displayed in a table: Loan ID, Book ID, Branch ID, Card Number, Date out, Due date.
- The table shows only those book_loans tuples corresponding to the Card Number, for which Check in has not yet been done.
- Once the book_loan tuple for the book that is to be checked in, is located on the displayed table, select the tuple by mouse click. This populates the corresponding loan_id in the non editable text field loan_id.
- Thereafter, clicking the Check in button sets the date_in attribute in the book_loans table for the tuple corresponding to the loan_id, to the current date. A useful message is displayed.
- If Check in button is clicked without the loan_id text field populated, error message is displayed
- If the date_in is past the due_date, the borrower is charged a fine. A message is displayed and a new tuple is created in the Fines table.

- In the **Fines** tab, on clicking the Show Fines button, if there are no unpaid fines, a useful message is displayed.
- Otherwise, a table showing Card Number, sum of 'fine_amt's as Total Fine, and Paid attributes are displayed, grouped by card_no. This shows the sum of the 'fine_amt's for each borrower.
- Selecting a tuple by mouse click populates the corresponding card_no in the non editable text field card_no.
- If a fine is to be paid, then click on Pay Fine? button. If there are no unpaid fines at all, a useful message is displayed.
- Otherwise, clicking on the Pay Fine? button displays a table showing each fine that has not been paid.
- Select the tuple for which fine is to be paid, by mouse click. This populates the corresponding loan_id in the non editable text field loan_id.

- Thereafter, clicking on the Paid button shows a message and sets paid to true in the corresponding tuple in the fines table in the database.

- If Paid button is clicked without the loan_id text field populated, error message is displayed.

Design decisions and justifications

- Swing components can change their appearance based on the current "look and feel" that's being used. One can use the same look and feel as the platform one’s on, or use a different look and feel
- Swing components follow the Model-View-Controller paradigm (MVC), and thus can provide a much more flexible UI.
- Swing provides "extras" for components, such as:
    - Icons on many components
    - Decorative borders for components
    - Tooltips for components
- Swing components are lightweight (less resource intensive)
- Swing provides built-in double buffering
- Swing provides paint debugging support for when you build your own components

Technical dependencies

Software versions:
NetBeans IDE 7.4
Java: 1.8.0
Libraries: SQL, Swing, AWT