GUI

I used Java Swing to create the GUI, mainly due to the fact that it's the only GUI library that I know how to use. It is easy to use and fairly intuitive to learn and develop. The callback structure for all of the different containers and buttons makes linking the actions to functions simple to add new features. Netbeans gives the user a graphical editor to easily move around and resize the objects. I also prefer it because it is native to most Java instillations and can be easily developed and run on most computers with a Java instillation.

I decided to make the main landing screen the search page because I figured this would be the most commonly used features of the application. It is also the starting point for most of the additional features because the librarians must know the ISBN of the books they are acting upon. To navigate to the additional features, I decided to use a drop-down menu because it's simple and most applications use this type of interface to open sub windows and perform actions not currently visible.

Database Creation

To initially populate the database, I wrote a few python scripts that parse the borrowers and books used to create the initial state of the database. I chose python because I'm pretty familiar with it and the string splitting and subdividing is quite robust. I knew that there would be a lot of substitution and string operations because there is a large amount of noise in the books dataset, especially the authors. The main script takes two arguments so the user can specify which CSV files they want to use for the initial population. In the testing, I created a few different book files to use a smaller set for the testing. I also wrote a separate script that creates the same tables and manually inserts a few books and borrowers in the database. This allowed me to independently test the application and the parsing of the books and borrowers.

MySQL JDBC

I used the default Java Database Connectivity library because the documentation was pretty good and this was my first time using Java and MySQL. It seemed to work quite well and gave good feedback for the queries and statements that I used. To log the MySQL commands I print the query, insert, and update strings to STDOUT so we can see what is actually going on under the hood.

Exceptions

I purposely designed some of the queries in the application to cause exceptions in certain cases. With the proper handling, we can get useful feedback to pass to the user. For the most part though the exception handlers are not intended to be used in the typical case and will cause the desired action to be terminated. A logging function prints the output to STDOOUT and can be used to debug the issue with the command.

Startup

The most important thing that happens at the start of the application is the query of the largest loan_id and card_id. This creates the initial value of these keys and allows the application to simply increment the load_id for new book_loans and increment the card_id when the librarian adds new users. The application will automatically close if it cannot find the "library" database so it is required that the system admin run the included setup script to initialize the database.

Assumptions

The main assumptions that I made in the application is that the user is not malicious and somewhat familiar with the application and the basic use of library systems. I did not do any type of MySQL command scrubbing so malicious users could potentially insert harmful commands into application and corrupt the database. I felt this assumption was appropriate because the system is designed to be used by librarians and not available to all users. I also felt the assumption that the user is familiar with the application because we will be including a getting started guide to help the users become familiar with the application and how to use it. Finally, I felt it was safe to assume that the users will be familiar with library systems because it is designed to be used by librarians who are more than likely familiar with how books are cataloged.

Improvements

Due to my limited knowledge of GUIs and how to create them I feel like the look and feel and also feature set of the GUI could be improved. It currently does not support selectable objects in the various search functions and requires the user to either copy and paste ISBNs and other numbers or manually type them in for each action. The scrollable text fields could also be improved to better format the data so the columns are aligned.

The security and reliability of the application could also use some improvement. The current application does not allow for multiple librarians to use the system simultaneously and opens the database as the root user. There is also no backing up or permeant logging of the database operations.