Xavier Cadot

3/11/18

CS 6363.003

Project 1: Library Database Design Document

Designing the database for the library was a daunting task for me as I have never created a database nor a GUI before. This project had both and they needed to be connected. I started this project first on focusing what would be the most efficient way to get this done. My most proficient programing language is JAVA and NetBeans has a built in GUI builder, so I knew that I would be using both of those to solve my GUI dilemma. I then discovered jbdc which allows java to open a connection an enable it to call MySQL commands. I found enough guides online to set that up and made a separate constructor to call when I needed to communicate with MySQL. This constructor has several different types of queries depending on what kind of return type I needed or what kind checking statements I required. Now that I got all of that figured out I would be able to start the project and building the database.

Building the database was no easy task and I made the wrong assumption that it was just like reading in a file. To create the book and author tables, I read each line of the book.csv file and parse them into their several respected tables. I had to debug this for a while as lot of the given data had some tricky situations to deal with. I got smarter with the borrowers file and decided to mess with the MySQL workbench where I directly imported the borrowers into dummy table. I use this dummy table to correctly align the data to match the given Borrower schema.

I design the GUI to be a multiple tab interface to handle all of the different library criteria. It has a Search, Borrow, Fine, and Return tab to fundamentally work as a library. The book search, availability, and checking out of books are all in the search tab. The search is done in a Jtable and is filled once the user pushes the search button. If nothing is in the search box, then it will show the whole database. Otherwise the user can search by author, book title or ISBN. The user will still retain result by putting partial information like a piece of the title or not the whole ISBN. This is because of using the LIKE and % in MySQL which allows for partial information. This search query is called using a subroutine in the MySQL workbench which makes it easier for submitting long queries. Pop up error windows come up when the borrower has the maximum amount of books out or if the book is already check out. Also one will come up if the user inputs a non-existent card ID. An interesting feature that I added is that the checkout button will not be enable into the user searches and chooses a book.

The Borrow tab is used to create a new borrower when they are not in the system. It has four fields for the user to input. These inputs are concatenated into an insert query to create a new tuple in the Borrower table. The program calls first and check the system to see if the user is creating an ID for someone that already has that SSN. This eliminates the possibility of a borrower having multiple card IDs.

Checking in books is done through the return tab. This tab has its own table to search the book name, card ID, or borrower name that has a book. This call a subroutine that call a query to fill the table with data. The user may then select a tuple and check in a book. The return tab is very similar to the search one.

The fines tab was tricky to implement and I had to be careful with the queries I implemented for it. What was really nice is I left MySQL to do all math for me concerning the difference in dates and summing up the fines. I used the date diff and the sum built in functions. I first found all the loan IDs that were past the due date and calculated the fines before storing them into the fines table. I then group the fines by card ID and sum up the card ID that had multiple book loans with fines. Before being able to pay, I check that the card ID had turned in all of his books. I assumed in fines that when the sum of fines is considered paid then makes all the fines within that paid. I also assume that the user must have all of his books turned in to pay the total fine.

Finishing this project took about fives time more than I originally thought. I would have never finished in time if it weren’t for all extensions to the deadline that professor gave us and I am thankful that I got that time. I learned a lot more about sequel when I had to figure out how to implement it for this project. The biggest bug issues was trying to figure out what exactly was wrong and how to approach it correctly. I discovered a lot of bugs towards the end when I was testing, and had to go back and change different parts so the search criteria’s would be correct. Overall this project was a really good experience gave me a good background in MySQL.