CS631: Fall 2017: Deliverable 3

Group 2: Zhenbo Qiao, Mike Lan, Leon Cheung

**Library Catalog: Overall Description and Assumptions**

1. **Goal**

After reviewing the solution for Relational Database Schema, we focus on two parts: create the library database with real data for testing and write corresponding python code including the sql queries and design a GUI for users.

1. **Implementation**

Besides create the database schema with instances, we have made some revisions on relational schema below:

* Add a table named “ADMIN” to store the login in “ID” and “Password”
* Add a table named “FINE” to store the “READERID”, “FINE”, and “OVERDUE”
* Add a attribute named ”FREQUENCY” to “COPY” to show the total number of times each copy have been borrowed
* Add a attribute named “FREQUENCY” to “READER” to show the total number of the documents he/she has borrowed
* Make the “READERID” in “READER”, “BORNUMNBER” in “BORROWS” and “RESNUMBER” in “RESERVES” be auto-increment so that each time we insert a new tuple, the corresponding number will increase by 1
* For details, refer to User Guide (attached separate) and Documentation (on the next page)

1. **Assumptions**

* We never delete the reader data in “BORROWS”, so that we can keep all borrow history in our database
* We assume that administrator can add document copies for any branch and manages that branch’s copies

1. **Difficulties**

* To get the 10 most popular books of a year, how to set the start and end date of year so that we can get the results.
* There was trouble obtaining the average late fees per user. We were able to use Python code to implement a function, but if the database were to be significantly larger then how would we accomplish this more efficiently. One idea is to include a due date in the BORROWS relation. This would make determining the calculation easier. In our implementation we had to code that in from the given borrow date. The main issue for this was getting the divisor, which is the number of late books per user to calculate the average.