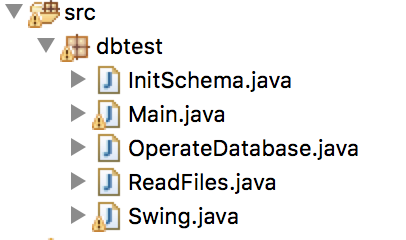
**I. Design Considerations**

1.General Constraints

* Compile source code with Eclipse
* Use a terminal to execute runnable .jar file
* Mac OS/Linux/Windows support
* End-user must support JVM for compile and run the project
* MySQL installed for end-user
* Assume use “root” user to built library data base
* The password for root is set “128517”
* The GUI for library system is built with Java Swing

2.Goals and Guidelines

* GUI will implement KISS principle ("Keep it simple stupid!")
* Allow search books with partial key words
* Search result contain book details including availability
* Be able to check out books based on user login function
* Administer can search book, or reset book availability(check in)
* Allow new user create account
* Set fines mechanism to allow administer to check or update fines
* Constraints are set for register, login, check out, check in process

****

**II. System Architecture**

The system could be decomposed into

three levels, which are database setup,

GUI Design and SQL operations.

1. Database Setup

The library system is designed based on MySQL. As the application read the data in CVS files, we created 7 tables including “book”, “book\_authors”, “book\_authors”, “book\_loans”, “borrower”, “borrower” and “library\_branch”.

Through these tables, system could support fuzzy search function, book check in/out record, fines auto-update, and etc.

To achieve these functions, we first use a ReadFiles class to import data. Next, we call InitSchema.initSchema() to initiate the schema of database's tables. After that, we call OperateDatabase.importData() to insert data into mapping table. The insertion will take about an hour. Then the library database is successfully setup.

2. GUI Design

We use Java Swing to create a fix 400x400 size interface for user. The GUI include:

(1). Login page: Once the application is loaded, the login page will show up in the front. A user or administer could sign in with card\_no.

(2). Register page: New user without a card\_no could click on “register” button to switch to register page to get a new card.

(3). User search page: After sign in, user could search book in this page. Search result will be show as a table.

(4). Book detail page: By double click on the book in table of search result, book detail will be show in another table including branch info and availability.

(5). Administer page: When log in as an administer, this page will show up. It allows administer to check book loan information by searching with book info or borrower info. Administer could also double click on any record in the research result table to check in a book, and sett returne date. By clicking on “check fines” button, it will jump to Fines page.

(6). Fines page:

In this page, administer could check all the unpaid fine records in table grouped by borrowers’ id. Double click a record could set fine to paid. Refresh table and the paid record will disappear.

3. SQL Operations

We use a OperateDatabase class to generate SQL instruction to implement query, insertion, deletion, updating database. In the Swing class, we set listeners to buttons and tables, in order to call the functions in OperateDatabase to operate on the library system.