

**Project 2 Report**  
**EECS3421 FALL 2017**

**First Name: Baidi**  
**Last Name: Liu**  
**EECS User Name: liubd**  
**Student #: 211559093**

## How to run

1,

Make sure there are 7 Java files in a same folder, shown below,

```
red 403 % ls
AppGui.java Book.java Category.java Customer.java JdbcUse.java Offer.java Purchase.java
red 404 %
```

2

Make sure you have your environmental variable set on your machine.

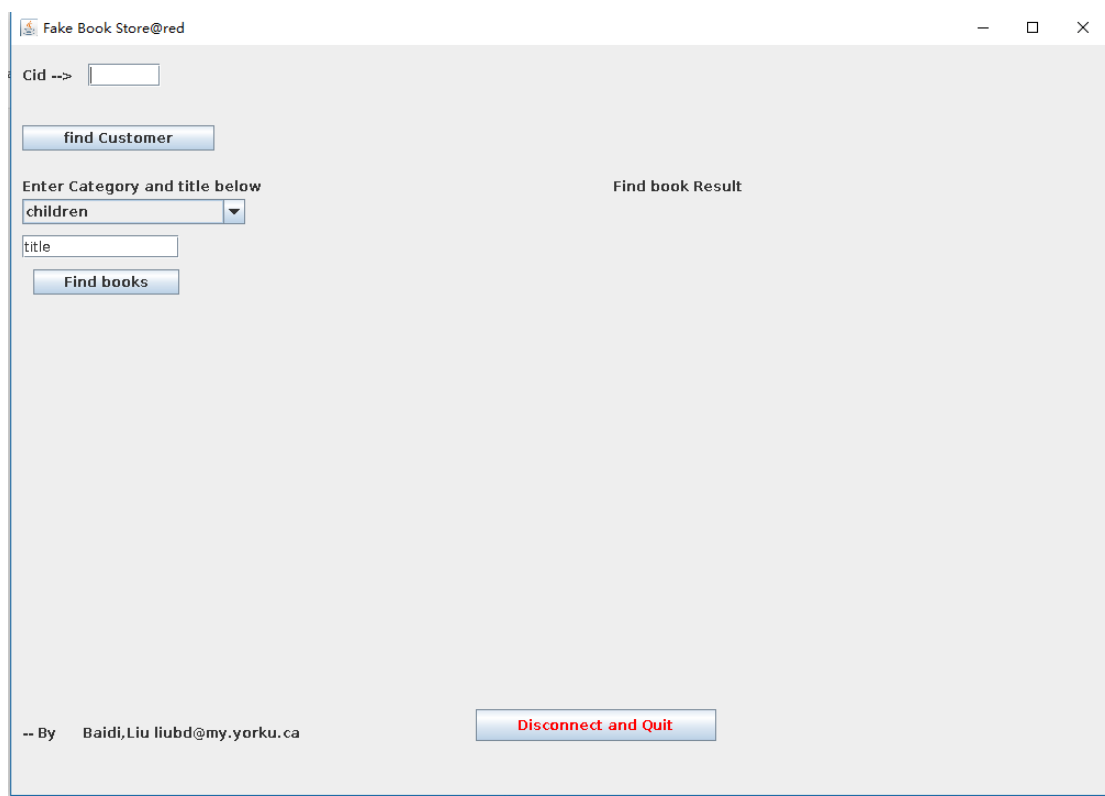
```
% source ~db2leduc/cshrc.runtime
```

3,

Compile and run AppGui.java, see the picture below,

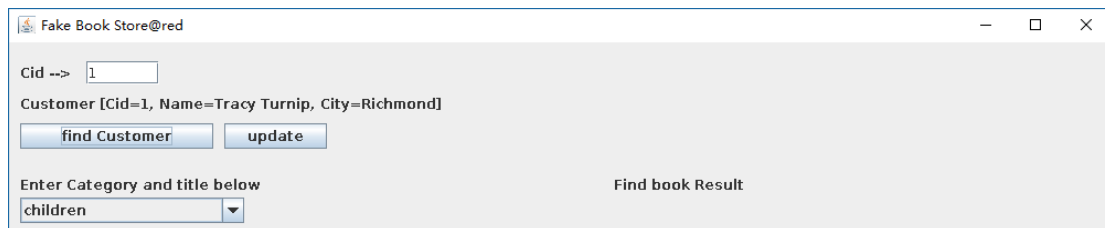
```
red 404 % javac AppGui.java
red 405 % java AppGui
```

Then the GUI will pop up.



## 1, Find Customer and Update Customer

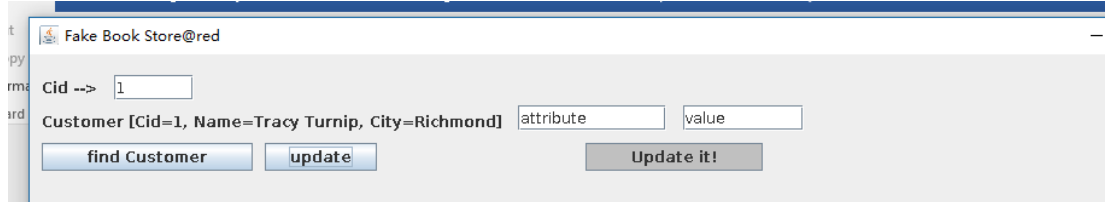
Now let's try to find the customer's information by his/her Cid, enter '1' into the Cid Text Field and press button "find Customer". Then the information of the customer will be shown right below the Text Field, and a "Update" button will pop up. See the picture below,



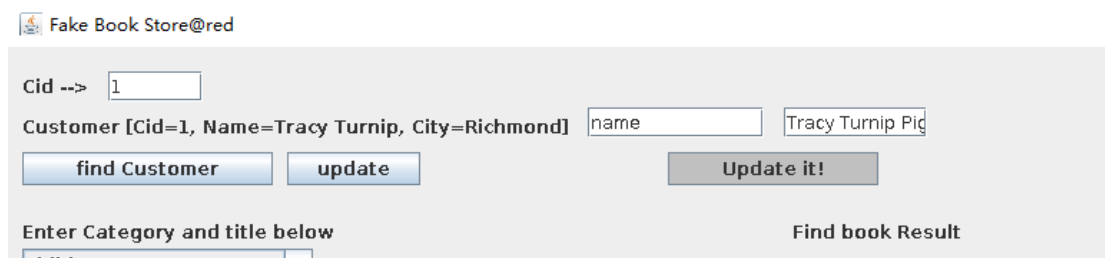
Now let's try to update the Customer's information, when we press the 'update' button, a confirm window will pop up to confirm your request.



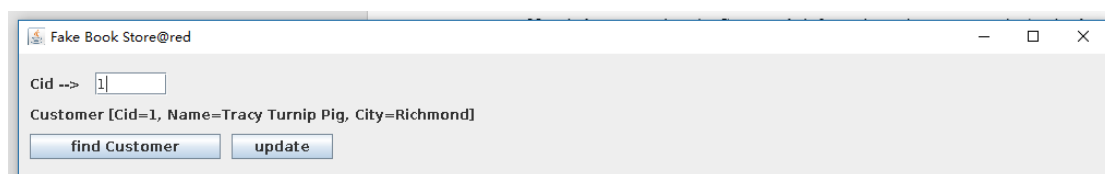
Hit yes! Then a "Update it" button and 2 text fields will pop up, as shown below.



Let's enter the attribute and value into the text field accordingly. For example, "name" and "Tracy Turnip Pig", and hit the "Update it!" button.



Now the Customer's information has been updated, the customer's name has changed to "Tracy Turnip Pig" and the update session of the GUI is gone until we hit the "update" button again.



Of course, we must change it back afterwards.

## find\_customer and update\_customer methods

These two methods are located in the JdbcUse class. Whenever the user hits the find customer button, a SQL query written in queryText will be prepared and executed. The output will be presented to the JLabel element in the AppGui.class. If the input cid is not valid, an error message will be presented instead.

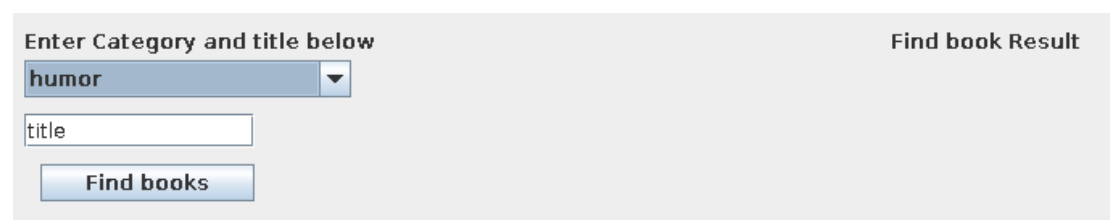
The update option will only appear after the Cid has been entered correctly. The “Update it” button will call the update\_customer method. executeUpdate() method will be used here to execute the update. And it will commit to the database with the latest information of the customer.

## 2. Fetch Categories and Find Books

The Categories section is a drop-down list which shows all the categories of books in the data base, see below.



Let's select “humor” in the categories drop down menu. After we select a category, the “Find books” button will pop up.



Enter “Relational Algebra” in the title text field and click “Find books” button. Then the result will be shown in a JList element. A session of Find minimum price of the book will appear.

Enter Category and title below

humor

Relational Algebra

Find books

Find book Result

Book [Title=Relational Algebra, Year=2000, Language=English, Cat=humor, Weight=144]

Select a book from above and Click to find min pric...

Find Min\_price

min:

### fetch\_categories and find\_books methods

The `fetch_categories` method will send a SQL query to the database to fetch all the categories and store them into an array-list and return it. The output will be presented as a JComboBox element in the GUI.

The `find_books` method will send a SQL query to the database to fetch the books with a selected category and a matched title from the title text field. An array-list with the books will be returned and presented to the JList element on the GUI.

### 3, Find Minimum Price and Calculate the Total Cost

At this state, my program looks like below. We have `Cid = 1` and the book Selected.

Fake Book Store@red

Cid --> 1

Customer [Cid=1, Name=Tracy Turnip, City=Richmond]

find Customer

update

Enter Category and title below

humor

Relational Algebra

Find books

Find book Result

Book [Title=Relational Algebra, Year=2000, Language=English, Cat=humor, Weight=144]

Select a book from above and Click to find min pric...

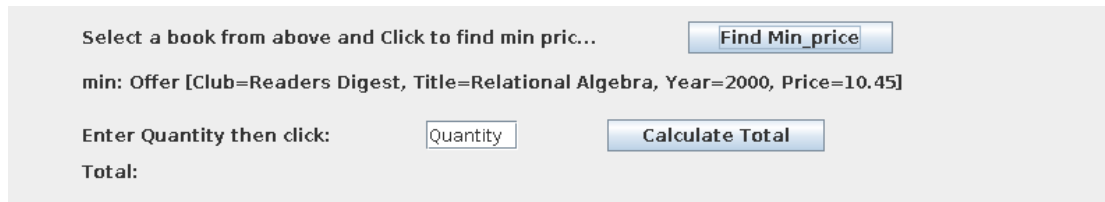
Find Min\_price

min:

-- By Baidi, Liu liubd@my.yorku.ca

Disconnect and Quit

Let's click the "Find Min\_price" button, it will call the min\_price method and find the minimum price that the customer with Cid = 1 can get according to his/her membership. After we click the button, the information of the book will be shown with the minimum price and a option to calculate the total cost with quantity.



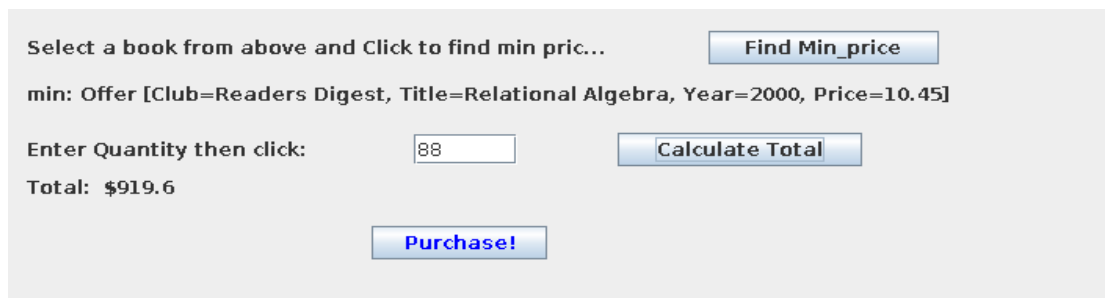
Select a book from above and Click to find min pric...

min: Offer [Club=Readers Digest, Title=Relational Algebra, Year=2000, Price=10.45]

Enter Quantity then click:

Total:

Let's say the customer wants to buy 88 books, we will enter 88 into the Quantity text field and click "Calculate Total".



Select a book from above and Click to find min pric...

min: Offer [Club=Readers Digest, Title=Relational Algebra, Year=2000, Price=10.45]

Enter Quantity then click:

Total: \$919.6

A "Purchase!" button will pop up and the Total of \$919.6 has been shown.

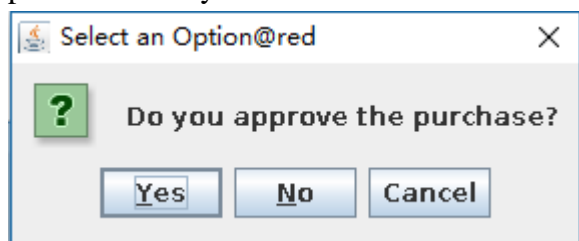
### min\_price method and calculate total

The min\_price method will send a SQL query to the database with all the requirements from before (for example, Cid, Category, title, year) and return an "Offer" object. I used toString method on the object, so it can be presented to the min JLabel elemnt.

For calculating total, it is done inside the AppGui.java class, it basically just takes the quantity from the text field and the min\_price and multiplies them then presented to the Total text field.

### 4, Purchase

Now, we have everything settled and we are ready to purchase the books. Let's hit Purchase. Then a window will pop up and ask the user, "do you approve the purchase?" hit yes!



A message will be shown below indicate that the purchase was successful.

Fake Book Store@red

Cid -->

Customer [Cid=1, Name=Tracy Turnip, City=Richmond]

Enter Category and title below

Find book Result

Book [Title=Relational Algebra, Year=2000, Language=English, Cat=humor, Weight=144]

Select a book from above and Click to find min price...

min: Offer [Club=Readers Digest, Title=Relational Algebra, Year=2000, Price=10.45]

Enter Quantity then click:

Total: \$919.6

Thank you for your purchase

-- By Baidi, Liu liubd@my.yorku.ca

At this time, a new record should be inserted into our yrb\_purchase table, let's check it by using some simple SQL query. We can see the record has already been inserted into the table.

```
db2 => SELECT * FROM yrb_purchase WHERE cid = 1
```

CID	CLUB	TITLE	YEAR	WHEN	QNTY
1	Basic	Will Snoopy find Lucy?	1985	2001-12-01-11.59.00.000000	1
1	Readers Digest	Flibber Gibber	2000	2001-12-01-11.59.00.000000	1
1	Readers Digest	Getting into Snork U.	2000	2001-12-01-11.59.00.000000	1
1	Readers Digest	Relational Algebra	2000	2017-12-03-21.39.42.415000	88
1	Readers Digest	Yon-juu Hachi	1948	1999-04-20-12.12.00.000000	1
1	W&M Club	Nothing but Steak	1991	2001-12-01-11.59.00.000000	1

6 record(s) selected.

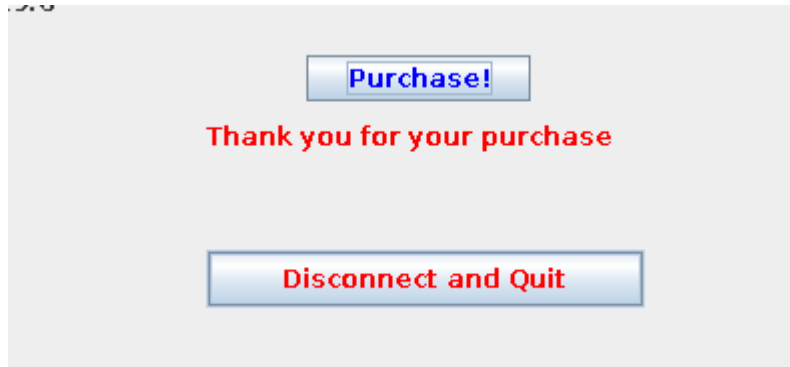
```
db2 =>
```

## insert\_purchase method

When the user clicks the "Purchase!" button then confirm by clicking "yes", this method will be called. It takes a Purchase object as argument and insert the information by SQL INSERT command and use executeUpdate method to execute the command and commit to the database when the method is done. The Purchase object is created with all the necessary information in the AppGui class when the confirmation button is hit and pass it as a parameter into the insert\_purchase method.

## 5, Disconnect and Quit

At last, the user can click the Disconnect and Quit button to disconnect with the database and quit the program.



### Classes

I created seven classes for this project.

AppGui.java – GUI, run the program with it.

JdbcUse.java – Store all the methods for the SQL update and queries

Customer.java, Category.java, Book.java, Offer.java, Purchase.java

- Objects according to the tables in the database.

### Tools

I used Java Swing to build the GUI.

I created seven classes