**Comp 730 + 830 Final Software Project Update 1**

Team 1: Sean Burwen, Greeshma Srinidhi

**Past Project Progress:**

In the first few weeks of the project, the goal has been to get the basic functionality of the program working in a “proof of concept” state. This means that the Listings Browser (browser.java) and the Listing object (listing.java) were the first steps to complete as they are the basis for the rest of the program. So far, these two java files have been created with their intended visual design and with placeholders for future functionality. Currently, browser.java is essentially a blank panel with a scrollbar which has a loop to dynamically add each listing as a panel to the screen. It also has a menu bar at the top which will serve to contain the “create listings” button, “edit listings” button, and search features later. Listing.java contains the visual elements for every listing and can take ListingID, Title, Description, Price, Quantity, and Image as inputs for its constructor. It can also update any of these through a set of setters and getters. The listings also have an “Add to cart” button as a placeholder for later.

Next comes the linking of database which includes creating the table and inserting data into the table. After successful creation of table, the database gets linked to the project. The data has to be retrieved in a way that it gets reflected in Listing.java and Browser.java

**Sean’s Past Activities:**

* Created Browser.java
  + Implemented the scroll pane with functionality for adding each listing to it with a loop. These listings are stored in an array called “listings array”.
  + Created the menu bar which will have create, edit, and search features in the future. The buttons are just placeholders for now.
* Created Listing.java
  + Determined and created the visual design of the listings. Each visual object’s text is tied to a variable that can be edited.
  + Implemented constructor with all relevant attributes as inputs.
  + Implemented setters and getters to update and retrieve attributes from listings.

**Greeshma’s Past Activities:**

* Database Linking
* Installed and set up SQLite database.
* Configured Build path.
* Downloaded JDBC driver and SQLite JAR files.
* Added external JAR files in project.
* Working on linking database to project.

**Next Steps:**

The immediate goal is to reach the specified 50% completion threshold. The following steps will take the project up to that point:

1. Connect Browser.java to the database so that it can read in listings from the table. This replaces the placeholder of 40 example listings that currently exist.
2. Add functionality for creating new listings.
   1. Any created listings will need to be added as entries into the database table and to the array in Browser.java. This can be done using the observer pattern.
   2. When a listing is created, it should also be added to the browser visually.
3. Create Cart.java to begin work on the cart functionality.
   1. The cart page will look similar to the listings browser but will only show items that have been added to the cart. If a listing is shown in the cart, it should have a “remove from cart” button in place of the “add to cart” button.
   2. The cart page will calculate the total price of the items.
   3. The cart page will have a “proceed to checkout” button.
   4. The “add to cart” button should be given its intended functionality. When an item is added to the cart, the quantity of the item will be reduced by 1.

**Difficulties:**

The majority of difficulties experienced so far have been with configuring the visual elements of Browser.java. For example, getting each listing panel to populate in the browser without overlapping each other was difficult to figure out. Once that part worked; getting the scrollbar to actually function was also very difficult. Lastly, getting the starting view of the page to be at the top was also difficult to figure out, since it was automatically putting the view at the bottom originally.

Another difficulty faced was while configuring the classpath in Eclipse to include the JDBC driver JAR file. As Java database programming is new for us, ensuring having the correct JDBC driver for database is crucial and it was confusing to download and configure the right driver. All these difficulties were eventually sorted out through a lot of googling.