**Comp 730 + 830 Final Software Project Update 2**

Team 1: Sean Burwen, Greeshma Srinidhi

**Past Project Progress:**

In the last update, the project was brought up to a “proof of concept” state which demonstrated the basic listing browser and individual listing functionality. This week, additional functionality and the required backend components were added to move the project forward. The first addition was the creation of an SQLite database which contains one table used to store all listings. The table is made up of 7 columns: ListingID, Title, Description, Price, Quantity, Image, and Seller. The database is interacted with through a separate class called Database.java; so far it has methods for determining the size of the table, pulling Listing information by ListingID, and Inserting listing data. Browser.java acts as the “main” java file and it interacts with the database through an object of the Database class. The “create listing” button in Browser.java now has functionality as well. When the button is clicked, it will open a dialog window which utilizes a file called CreateListing.java for its Jframe design and functionality. The frame contains fields for the user to fill, which correspond to each field of a listing and column of a table. This creates a new listing object which is then added to the screen of Browser.java and to the database table.

**Sean’s Activities:**

* Created Database.java
  + Moved existing database functions created by Greeshma to this file.
  + Created fetchTableSize() method to find the amount of items in the table.
  + Created insertListingData() method to add new listings to the table.
* Improved Listing.java
  + Added fields and methods for “Seller”.
* Created CreateListing.java
  + Created the visual design. This includes text fields to fill for creating the listing and buttons to interact with.
  + Implemented the submit button which creates a new listing based on text in the fields. There is also a cancel button which will close the dialog window.
* Updated Browser.java
  + Added action listener for the “create listing” button. This opens CreateListing.java as a dialog window and returns a new listing object. The listing object gets added to the screen and to the database here.

**Greeshma’s Activities:**

* Database Creation
* Created new Database name Listings.
* Created a table in the Database.
* Inserted sample data to check functionality.
* Updated Browser.java
* Written code for connecting SQLite Database in java.
* Written code which displays the inserted sample data in output.

**Next Steps:**

1. Begin work on the cart functionality.
   1. The cart page will look like 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.
2. Begin work on the checkout page. (this depends on the cart being done)
   1. The checkout page should be a dialog box which acts similarly to the create listings dialog box.
   2. Once the checkout page has been completed, a receipt should be generated and returned. Functionality per-user will come later.
   3. Once the checkout is complete, all purchased items should have their quantity reduced by 1 and the cart should be cleared.
3. Begin work on user authentication.
   1. A login button should be added to the menu bar in Browser.java. Once the user has logged in, the login button should change to become a “log out” button.
   2. The rest of the buttons should not be interactable until a user has logged in.
   3. User account information should be stored in an additional database table. (this is not very secure, but it’s just proof of concept).
   4. We can keep this simple for ourselves by keeping most of the visual elements not visible until a user has logged in. When the user logs out, we can make them invisible again and clear any relevant data.

**Difficulties:**

One major difficulty was with the file path for accessing the database. Since an absolute path is required for the connection, a dynamic way of determining that absolute path was required. Another difficulty was getting Browser.java to pause until the popup window for creating a listing was filled out. It took a lot of researching to find modal dialog boxes as the solution to this problem. It was also difficult to figure out how to insert new data into the database, this took some time to research as well.

One major difficulty is determining the appropriate path for creating the database file. Depending on the environment and project structure, constructing the SQL syntax for creating a table with appropriate columns and data types and Inserting sample data involves understanding the correct SQL syntax for insertion and ensuring that the data types and values match the table schema.