Project 5 Report

Team Members: Willie Chen, Artem Kolomiiets, Pranati Patchigolla and Yixun Lu

CS180

**Part 1:**

Our group, which consisted of four team members: Willie Chen, Artem Kolomiiets, Yixun Lu, and Pranati Patchigolla selected option 3, which was the official marketplace option. The official marketplace provides a platform where sellers can list their products and buyers can check the availability and make a purchase through the marketplace. Customers can interact with the GUI concurrently using network IO to view the overall marketplace to search for products, interact with the marketplace and make purchases, while multiple sellers create, edit, and delete products on their stores all while viewing each customers individual details and revenues from sale. For the selections, we picked to implement the File and Statistics options. We selected these as we knew it would provide a user a good user experience when using the program, as user would receive valuable information such as personalizing their selection between transactions. We made it clean and organized by proving users with a detailed list of all products available in the store along with the product’s purchase history. Products are also organized using number of products purchased by each customer and can also be sorted using the sales volume of the sellers. While implementing these features, we also made sure to implement the optional features we had discussed earlier, the first and third optional features. The first optional feature we selected made sure that the Seller can hold sales and their prices until a specific unit of products are sold. Through this implementation, the customers will be able to see the original price and the reduced sales price. Coming to the other optional feature, through our implementation, sellers will be able to set a bound per product to be able to be sold, which prevents the user from buying more than the limit. Through this implementation, the customer will not be able to buy more of the product with the bound unless it is removed, or the limit is edited to accommodate more. Through the implementation of these optional features, the official marketplace becomes more advanced and customizable to the seller. In perspective of the customers, it becomes more transparent and open in terms of the information regarding the products. Through this implementation, the official marketplace promotes equality and collaboration between the customers and the sellers. This was the main reason we picked these optional features as it provides more clarity to both the seller and the customer while using the platform.

Coming to the individual components of the marketplace, the Main.java class is the most important class for our project. It is used to create user accounts and used to run and open the program. The Store.java class is required to provide a platform to implement then entire marketplaces functionalities, such as editing products. Seller.java is the mains base for seller to list their products, sort and edit. Customers.java plays the critical role of providing a platform for multiple other branches of code to implement off of. These include the User.java class and the UserBase.java class. These classes implement and contain information necessary for the Customers.java class.

While implementing the project, we split the project into several components of the marketplace including Server, MarketPlace, Store, Main, Exception, Seller, UserBase, Customers, Sellers, AllProducts, AllStore, and Client classes to make sure that our marketplace stayed organized and make editing and debugging the code easy. The Seller class was responsible for making sure all the sellers information was correct and was used to add and delete sellers as well. The UserBase was implemented off of Sellers. Userbase is used through the implementation of the GUIs to create user account and differentiate the functionalities between both. Sellers can view, edit and delete their list of products through this class, which would appear in the store class. The Store.java class is required to serve as a platform for all the products that the sellers list to provide a platform to implement the entire marketplaces functionalities through the main class. Customers.java plays an important role of providing a userbase for customers as well as providing a platform for multiple other classes to implement off of such as User.java class and the UserBase.java class. These implemented classes contain important information which would be required to be used in the Customers.java class. This also made it so that everyone could contribute in an organized way which would also make it easy to differentiate the designation of work. Our program updates data every time a user changes their input, or the seller changes their product. Another important implementation we did was to follow the Client Server model. We wanted to make sure everything ran smoothly and that all the data was store in the server side, so we reworked a lot of code we thought was failing to achieve this, as well as rework logic.

We used this design style of implementing different classes to make it easier to debug, as well as understand where we could implement more methods. It was more clear when coming to designated work as we could easily split up the work depending on the classes we could assign our team members. We were able to fix errors that came up with the Seller.java class in project 4. We tested our code to make sure everything was running smoothly. We were able to also improve the market transaction process to make it smoother and more detailed. We felt as though through this implementation we were able to run things smoothly and figure out where we were lacking in some respects. We felt as though this was the perfect approach in terms of what we could’ve done as we were able to combat everything we were concerned about through this implementation. We overall tried to make sure that everything ran smoothly, and the users have a comfortable time using our interface through this implementation. Through the use of this implementation of segregated classes, we were able to effectively implement our official marketplace. We were satisfied with our output and overall, we go to make sure that we receive positive input and constructive feedback from our project.

**Part 2:**

1. **Yixun Lu**

In Project 5, my mission was to add network IO to the program we designed in Project 4. This allows users to use our program not only just on a single machine but also through a server that stores user information, with all user interactions occurring on this server via Client.java. When starting to use our program, users can choose to 1. Log in, 2. Create an account, or 3. Exist. When creating an account, users need to enter their email, password, nickname, and buyer or seller status. When logging in, users must enter their email and password to access their account. Users can edit their account information, search for stores and products of interest, view their purchase history, and buy or refund products. In addition, when searching for products of interest, users can sort by price in ascending or descending order, purchase quantity, or display the default market order. This aligns with our design philosophy and principles, aiming to create a simple and easy-to-use program with diverse features. This experience was unforgettable for me, and it was the first time I built such a large network IO. I was a little overwhelmed when I first got the content of Project 5, but after discussing with my teammates and carefully teaching them, I gradually found my direction. This project experience is valuable and cherished and it will have a great influence on my computer science learning life in the future.

If given the chance to do it again, I would try to streamline our code further. Sometimes, I could replace many lines of code with a single function, but I am not very familiar with how to apply that function, so I usually write more lines of code to substitute for the unfamiliar function model. In fact, I would like to use more unfamiliar methods in my program, as familiarizing myself with more functions and techniques would lead to further breakthroughs in my Java skills which is one of my goals. Furthermore, I was actually more interested in the direction of option 1 at first, and I also thought that the option 1 program looked more interesting, but our group chose option 3 in the end, which is also a small pity of this project.

1. **Artem Kolomiiets**

In project 5, I was responsible for designing and implementing all of the GUI into the program. For the most part, this consisted of countless different simple GUI all with different functionalities, such as: information messages, error messages, warning messages and question messages. Our program contains a plethora of different functionality for both the customers and sellers, which meant a large amount of unique GUI had to be created. In some instances, I deemed simple GUI to not be the most efficient solution, and created complex GUI, that had several functions or combined multiple simple GUI in one for those cases. This ensured a smoother user experience. After creating the GUI, I was tasked with implementing it into the program in the correct place.

In addition to all the coding, I produced the demo for the presentation, which in a couple of screenshots shows some of the main features of the program. This would be sufficient to visually demonstrate and help understand the main functionality of the application, for someone like a possible investor or user. I then came up with possible questions and corresponding answers that a recruiter or manager might have concerning the project, such as the skills that were developed from participation in the project and the possibility of developing the application further.

If I was given the chance to do something different, I would not change anything in particular concerning project 5. The changes would all be in project 4, where  the team suffered from not enough organisation and assignment of deadlines, which ultimately led to us having to rush some fixes on the last day. This led to our code not being completely correct and had a few errors, that we did not have time to fix. This could have been avoided by better and more clear planning of both the general structure of the program and the work to be done by each member and appropriate deadlines. This could have been avoided by better and more clear planning of both the general structure of the program and the work to be done by each member and appropriate deadlines. I believe we as a team all understood this, and worked better during project 5, completing it earlier as everyone was more clear with what they had to do and by which deadline.

1. **Pranati Patchigolla**

In project 5, I was responsible for implementing and fixing up the Seller.java file and seller related files that had been causing us errors in project 4. I made the code for the Seller class, Store class, and Product class. Since our official marketplace needed to in incorporate the additional features that we had initially decided, I had to code in these additional features through the use of Arraylists and “instanceof” to make sure everything ran smoothly with ease. I was able to start early so I was able to make sure that out code was the most efficient, organized and consistent with the extra time. I was able to edit individual classes such as the seller extended class UserBase class, which was used to create sellers through the main class to add on to the marketplace on our database. In order to incorporate the optional features we had to where to create methods to hold sale, and check if the produc has been purchased in order to update the number of sales done. Finally I had to make sure the code compiled and the methods could be incorporated into the main class code. To be able to make sure the code I wrote was effectively implemented into the main clas, I thought it was important to split the seller side functions into different classes to maintain efficiency.

If I was given the chance to try doing something differently for project 5, I would say I wouldn’t change anything about it. We all learned from our mistakes in Project 4, where we ended up starting at the last minute so we did not have time to debug, and took that into consideration and started much earlier. We were able to finish the code earlier than expected as everyone was able to contribute this time, leading to a more clear understanding of the assignment and the deadlines.

1. **Willie Chen**