Guidelines for Final Project

Deadlines: Initial Design **(3%):** 4/14, Final Project **(7%):** 4/28
This is a group project (group of 3 members); JOIN your group on Canvas

Objective:

Design, develop and implement the working software system in Java.

Problem Statement – Online Marketplace System

The project will mimic an online marketplace system like Amazon which allows sellers to provide an inventory of items to be sold; buyers to be able to purchase those items; and the Marketplace to keep track of shipped items. This simplified system ('mini-Amazon') can be visualized in three different ways:

- **Seller's Perspective:** The sellers should be able to register themselves with the Marketplace (and they will be issued unique seller ID, passwords, and contact email address). Once registered, seller will be able to upload the inventory of items available for sale. Each item will have the following attributes: unique item number, item name, brief description, seller ID, quantity, price, etc. Seller should be able to manage the inventory (i.e., show current inventory, increase or decrease quantity, change price, update description of item, but not the seller ID).
- **Buyer's Perspective:** Buyer my be able to register their accounts with the market place and each buyer will have a unique ID, name, email. Buyers should be able to view the current inventory available for sale and should also be able to search for a particular item by name or ID. Buyer is allowed to purchase any item (if the quantity is available).
- **Marketplace's Perspective:** The marketplace system is responsible to make sure that the inventory is properly maintained. It also ensures that the item is shipped upon the sale of the item. At any time, Marketplace Administrator should be able to check any attribute regarding sellers, buyers or items.

The software development cycle includes requirement specification, design and analysis, coding and testing phases. You will also prepare a report documenting each step used in the software design cycle and will use UML diagrams to properly represent all classes used in your design. (You may use free software called StarUML which can be downloaded from http://staruml.sourceforge.net/en/download.php)

System Functionalities:

The Marketplace system that you are designing offers a variety of design choices. You are encouraged to design the software system that provides the most realistic user experience. Here are some of the required functionalities:

- Seller should be able to register their accounts with the marketplace. Upon registration, Marketplace system will issue a unique seller ID to the Seller.
- Seller should be able to use their username and password to login to the marketplace system and update their registration information (except the Seller

ID). Marketplace will verify the correct credentials and will offer different options to manage seller account details (such as Update name, Update email, etc.)

- Seller can also update the inventory list on the marketplace (after user logs in). The inventory will consist of the list of items available for sale (each item has a number, name, description, category, price, etc.). The list may be entered by the user or read from a file. The system should offer the seller a number of options to manage the inventory such as add inventory item(s), delete inventory item(s), update item details, etc.
- The Marketplace inventory system can be divided into different categories and sellers are allowed to add items in each category. Different sellers may have different item names, descriptions and prices but each item will have unique item number and will be part of some category of items.
- Buyers should be able to register on the Marketplace. The buyer registration process is somewhat similar to the seller registration process. Buyer should also be able to update their credentials.
- Buyers should be able to use the marketplace by the use of available menu options. For example, the buyer should be able to see the list of items available for sale (in each category) and should be able to purchase the item if the desired quantity is available. Buyer should also be able to *search* for the item using item number or name. If multiples search results are found, buyer should be able to buy the item of choice. If item is out of stock, seller should be notified by the marketplace. This notification can be displayed to the seller once he logs in to the system. Once the item is purchased the seller will ship the item and will notify the buyer about the shipping after the payment is made. This notification can be shows to the buyer once he logs in. Marketplace will keep track of all transaction (i.e., all sold items, their payment status and shipping status). Inventory list is updated after any successful transaction.
- Marketplace administrator should be able to see the list of buyers, sellers, and generate reports like inventory by selected seller, buyer purchase history, sales history (on any specific day or within last week or month), shipping status of any item, etc. Administrator should also have access to update any seller and buyer account details as well as update any inventory item.
- Make your system user friendly by providing sufficient guidelines and help to use the system (For example: if your system is expecting any input in a specific format, be sure to specify that in the instructions as well as the Testing part of your report)
- **Note:** Your Marketplace system should have **state persistence** by saving the necessary data to files. State persistence allows your system to outlive the process that created it. For example, if you add a few users (sellers and buyers) to the system when you initially run it and then close the execution before running the system again, the system must be able to remember the previous users and its state. This will allow you to use the previously entered information without having to create the buyers, seller, inventories, etc. again and again.
- You have tremendous amount of freedom in designing this system. *You are welcome to add any additional functionality to your marketplace system. Just imagine that you are designing your own website rivaling eBay or Amazon.*
- *Note: The design of a Graphical User Interface is optional. You can also develop a command-line based system*

System Development:

The final project includes two major components: Project Design and Implementation.

Design (3%): The design component will consist of the software design for the project and will include the list of all classes (member variables and functions) to be used, their relationships & collaboration, as well as databases/files. The initial design must precede any implementation. You will start by analyzing the requirements of the project and by identifying the classes required along with attributes and functionalities. A doc/docx document listing the design along with UML diagrams (Class diagrams, activity diagrams, etc.) must be submitted by April 14th (1159pm). Please include the following:

- Skeleton Code with all classes (with member attributes and member functions)
- What are class hierarchies and relationships? (in your report)
- All other data structures or files to be used (in your report)
- Peer evaluations (Groups **and** individual) are REQUIRED

It would be beneficial for you to analyze your system thoroughly and provide detailed UML diagrams. Here are some of the questions, answering those will help you put on the good path:

- How many classes will you have and how will they interact?
- How will you store using seller names, passwords, items and their details?
- How you ensure that all seller items are visible to the buyers
- How will buyer activities (buying or selling) impact the seller inventory?
- How will you deal with multiple sellers and multiple buyers?
- How will admin be able to access seller and buyer information?

Implementation (7%): Once you complete the initial design, start implementing the buyer, seller and system functionalities individually. Test them separately with individual drivers. Be sure to add the appropriate user-friendly menu options that would allow the user to select various actions easily. Ensure that the menu options only work when correct input is provided by the user. Test each option individually to ensure that all available functionality is properly implemented.

Now imagine yourself as the user of the system (as Administrator, Seller and then Buyer) and observe the behavior of the system (and see how it changes). Reflect on the efficiency of your choices. Review your initial design and make the necessary changes to remove any unnecessary attributes and functions. You may have to add some more functionality that you might have missed in your initial design. Implement the updated design and report the updates to the initial design along with a discussion regarding the efficiency of your choices. A doc/docx document listing the final design along with UML diagrams must be submitted along with working Source Code by April 28th (1159pm). See the **checklist** given below.

Testing: Develop and build unit tests for each operator. Create different drivers for testing each component of the system as a Seller, Buyer and Administrator separately and then integrate the entire system and test with a separate main driver.

Please ensure that the complete system is properly tested before submission

Deliverables:

The following items must be submitted:

- **All source and header files** related to the system implementation.
- **All drivers** that are part of the testing as Buyer, Seller and Administrator separated in different folders
- A **README** file with any information regarding compilation and testing I need to know in order to successfully compile and run your system. Include **any other files** needed to compile or test
- **Source code must be properly organized, readable, and must use proper best coding practices.**
- **The report file** with the initial design, updated design with discussion, and testing activities. Feel free to show additional testing you performed in the report
- Peer evaluations (Groups **and** individual) are REQUIRED

Project Checklist:

You can use the following checklist to ensure that you have submitted everything required for the project:

- Did you submit evidence of successful compilation or testing (screen captures)?
- Did you submit the instructions required to compile your code?
- Did you submit all files including any new header files used for compilation?
- Did you implement buyer’s features: password, unique ID, inventory, etc.?
- Did you submit a report with your designs (and UML diagrams), evidence of successful testing of all features?