EECS 3311
Software Design
Project (100 points), Version 1

Instructor: Song & Ilir
Release Date: March 14, 2022

Due: 11:59 PM, Thursday, April 7, 2021

One of the efficient way to evaluate a software design is to examine its effectiveness and usefulness in development. This project will provide you a different type of experience, i.e., developing software under the guidance of a given design (from your Midterm). Specifically, you will be asked to further develop the SmartShoppers project based on your design from Midterm. **Note that, the evaluation of this project will be on 1. the completeness of your implementation for the requirements and 2. lessons learned regarding the pros and cons of your design during the development of the project.** Note that, your design WILL NOT BE EVALUATED in this course project.

All your lab and project submissions must be compilable on the department machines. It is then crucial that should you choose to work on your own machine, you are responsible for testing your project before submitting it for grading. This project is intended to help you practice design principles we have introduced in this course.

Check the Amendments section of this document regularly for changes, fixes, and clarifications.

Ask questions on our slack channel [1].

1 Policies

- Your (submitted or un-submitted) solution to this assignment (which is not revealed to the public) remains the property of the EECS department. Do not distribute or share your code in any public media (e.g., a non-private Github repository) in any way, shape, or form before you get the permission from your instructors.

- You are required to **work on your own for this project**. No group partners are allowed.
- When you submit your solution, you claim that it is solely your work. Therefore, it is considered as an violation of academic integrity if you copy or share any parts of your code or documentation.
- When assessing your submission, the instructor and TA may examine your doc/code, and suspicious submissions will be reported to the department/faculty if necessary. We do not tolerate academic dishonesty, so please obey this policy strictly.

- You are entirely responsible for making your submission in time.

- You may submit multiple times prior to the deadline: only the last submission before the deadline will be graded.
- Practice submitting your project early even before it is in its final form.
- No excuses will be accepted for failing to submit shortly before the deadline.

[1] https://join.slack.com/t/eecs3311-winter-2022/shared_invite/zt-13rn5czkh-AYms0i7a1i~MACbqJfhOJA

- Back up your work periodically, so as to minimize the damage should any sort of computer failures occur. You can use a **private** Github repository for your labs/projects.
- The deadline is strict with no excuses.
- Emailing your solutions to the instruction or TAs will not be acceptable.

Amendments

- so far so good

2 Problem Description

In this project, you are expected to implement a software system for the project we introduced in the Midterm exam.

Background: The SmartShoppers system is an online system that allows customers to find products in the physical retail stores with a greater precision. The main goal of the software system is to provide the customers with a faster and smooth shopping experience in store. SmartShoppers would allow customers to find the products that are available in a specific store and its location in the store. This product will have many benefits to the customer and the SmartShoppers Inc. The customer will have financial benefits as they are able to save money by being provided all the on-sale items. The customer also is able to find recommended products which they might not have had a chance to find in store. The customer saves time shopping because they know the exact locations of the products which they would like to purchase. ShoppersLand Inc. would benefit financially as they will sell more products because of the SmartShoppers online system. ShoppersLand's employees would also be able to work efficiently as they will be asked fewer questions by the customers regarding the products.

Specifications: The detailed specification is in **Software-Requirement-Specification.pdf**.

3 Getting Started

- There is no template for the course project. You should create your own Eclipse project named **EECS3311-project**.
- You can use maven or gradle[2] to maintain the libraries and dependencies of your project.
- You are suggested to follow MVC[3] model (i.e., Model–view–controller) to separate meta models, business logic, and GUI components during developing the project.
- Your project must be a Java GUI project.

4 You Tasks

4.1 Implement the System (50%)

- You are expected to develop valid implementations and meet **all the listed requirements** in the software requirement specification document.
- You can design and implement your system with any design patterns that are applicable (can be different from these in your Midterm solution).
- Your system should have basic Java GUI interfaces. You can use any third-party GUI libraries in your project. Database is not required, you can use text files (or csv file) to mimic the database. **Good GUI implementation will get 5 bonus points.**

[2] https://gradle.org/
[3] `https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller`

- You could make any assumption for any requirement (scenario, business logic, etc), while your assumptions should be reasonable and justified in your report.

4.2 Write Test Cases to Test Your Project (20%)

You are required to add as many tests as you judge necessary to test the correctness of your implementation.

- You must add **at least 2 test cases for each requirement**, and all of them must pass. (In fact, you should write as many as you think is necessary.) **Let's skip the GUI related classes when writing tests.**
- You will be assessed by the quality of your tests via the code coverage. For the non-GUI classes, the average code coverage should be larger than 80% (i.e., after executing your test cases, more than 80% code should be covered). You can use JaCoCo[4] to measure the code coverage of your test cases. JaCoCo has an Eclipse plugin which is available in the Eclipse Marketplace.

4.3 Project Report (20%)

- The **latest** class diagrams for your implementation (`https://app.diagrams.net/`); You must also include the **draw.io XML source file of your class diagram and its exported PDF** in the docs directory when you make your electronic submission.
- During the implementation of the project, **what are the problems in your initial design from Midterm?**
- **Please comment the difference between the class diagrams of your final implementation and the initial design in the midterm.**
- At most 10 pages (A4, double spaces, font size: 11pt).

4.4 Project Demo (10%)

Please shoot a 5-mins demo video of your project, in which you can show how a user or admin can use your software with the developed features.

5 Submission

To get ready to submit:

- Close Eclipse
- Zip your project, report, demo video together with name 'EECS3311-Project.zip'.

By the due date, submit on the eClass:

[4] https://www.eclemma.org/jacoco/