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

Instructor: Song Wang
Release Date: March 10, 2021

Due: 11:59 PM, Wednesday, April 15, 2021

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.
- 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.

[1] https://eclass.yorku.ca/eclass/mod/url/view.php?id=536469

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 Toronto Parking Authority operates over 17,500 parking spots across Toronto. Approximately, 10,000 of these spaces are by 2,000 ticket meters and the additional 7,500 by single space meters. Currently, a customer either purchases a ticket from the machine and places it on the car's dashboard or inserts coins into a single spot meter. Employees have to walk weekly to collect money from these 9,500 locations. Additionally, the Toronto Police Service offers parking enforcement via dispatched personnel who randomly select areas to monitor. These personnel need to check tickets or parking meters manually and issue tickets as needed. In order to minimize the number of locations to collect money, the Toronto Parking Authority wants to develop an online platform to process payment for select spaces where single parking meters are currently used. The spaces will be numbered and the registered customer can select the space number from their device online to initiate payment. Additionally, parking enforcement will have a separate login into the application which will allow them to see whether an occupied space has been paid for or manage the parking locations.

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.

4 You Tasks

4.1 Implement the System

- You are expected to write 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.
- Your system should have basic 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.**
- 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

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.)

[2] https://gradle.org/

- 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[3] 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

- Compile and print off a report including: names and your CSE logins.
- The 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.
- You have finished an initial design for the project in the midterm exam. During the implementation of the project, have your strictly follow your initial design? **If not, what are the problems in your initial design?**
- **Please comment the difference between the class diagrams of your final implementation and the initial design in the midterm (If applicable).**
- At most 10 pages (A4, double spaces, font size: 11pt).

5 Submission

To get ready to submit:

- Close Eclipse
- Zip your project with name 'EECS3311-Project.zip'.

By the due date, submit via the following command:

```
submit 3311 proj EECS3311-Project.zip
```

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