CSE 120: Software Engineering Course — Fall 2022

Midterm Exam — Total points possible: 100

Due Date: 06-NOV, 11:59pm

Please note the following:

- This is a take-home exam with no proctoring. You can start at any time once you get access to this exam. You must submit your solution by the due date through CatCourses (Midterm Exam under Assignments). You can resubmit any number of times until the deadline. If there are any technical issues with uploading, it is extremely important to immediately contact the instructor or the TA.
- This is an open-book exam. The **only resources you can use during the exam** are all **course materials** uploaded to CatCourses plus the **textbooks**. This means that you are NOT allowed to search for solutions on the internet. The ONLY exception is to refer to links provided in Lab 02 handout (as they count towards to course materials uploaded to CatCourses).
- **You must take the exam by yourself**. You may NOT collaborate with anyone else in or outside the class (including your teammates).
- There are 3 tasks in total. You can earn some partial points if you show progress even if you do not solve the tasks to completion.
- Please answer precisely. If there is something in the question that you believe is open to interpretation, then please go ahead and interpret, but state your assumptions in your answer.
- If you have questions, you can email the instructor and the TA. Please make sure to send your questions before 03-NOV 11:59pm.
- You are required to take the following honor pledge prior to taking the exam. Please print this text verbatim at the beginning of your submission file right below the title (see the Midterm Exam assignment page on CatCourses to see how to title your submission).

> ***By completing this exam, I acknowledge and confirm that I will not give or receive any unauthorized assistance on this examination. I will conduct myself within the guidelines of the university academic integrity guidelines.***

The Office of Student Life of a university has hired you to design a mobile app to help students ease their frustrations when it is time to register for classes.

Below is a list of features that your client would like to include in the app:

- Generate up to 50 different schedules that fit your classes.
- Filters to tailor the schedule to your life such as focusing on late classes, minimizing gaps, and even input your work schedule and meeting times.
- Port your created schedule to Google Calendar with the click of a button.
- Follow classes and receive a notification when a spot opens up.

Task 1: Create user stories for each of the features mentioned in the list above. You may choose to have more than one user story for each feature. Each user story must also include the acceptance tests/criteria. Note that for each feature, you must show at least one (user story, acceptance criteria) pair.

Task 2: For **each** of the user stories from Task 1, list the following use case information (i.e., you must have exactly one use case for each user story):

- Name
- Actors
- Entry Conditions
- Exit Conditions
- Event Flow (showing basic and alternate flows)
- Special Requirements (if any)

Task 3: Create a UML use-case diagram for this app. There should be only one diagram that includes all the use cases described. The use case diagram should clearly show the actors, subject (with boundary clearly shown), use cases, associations between actors and use cases, multiplicity of associations (if any) and relationships between use cases (if any).