Homework 3: Design

Start Assignment

Due Wednesday by 6pm **Points** 25 **Submitting** a file upload **File Types** pdf

** This is a group project - please find a partner (or two if we have odd numbers) and come see me during class to indicate who your partner is! **

Case Study: A Class Registration System

Problem Statement

On SU Online, students are able to search and register for classes. In order to add a course, students need to meet the prerequisites and the class may not be full. Students may also drop a class they decide not to take before the registration deadline. Students can only take a maximum of 18 credits if undergrad or 12 credits if grad student.

Students must be logged into SU Online before attempting to add or drop, but can search for classes either logged in or not. If the class is full, students must send an email to their advisor with their name, SUID, and the course they want to take; students will be put on a waitlist for the requested course. If space allows, students will be automatically enrolled in the order they are listed on the waitlist.

Determine Use Cases

From the problem statement, determine use cases for the class registration system. Note that only a list of high-level use cases are required for this assignment.

Finding Candidate Classes

Using the noun extraction approach, underline all noun phrases in the problem description.

Steps for Creating CRC cards

1. Identify classes
   - Based on the nouns in the problem statement, find initial candidate class list
     - Divide into “probably classes,” “possibly classes,” and “rejected as classes”
       - Remove any instances of classes (i.e. objects) or duplicate classes
       - Group similar classes together
     - Determine if there are any "hidden" classes that are not specified in your candidate class list
     - Filter candidate list.
   - Once you've solidified the list of classes,
     - Write the name(s) of the class on the index card
     - Underline the class name
     - On the back of the index card, write an informal description of the class
2. Identify responsibilities (for each class)
   - Include
     - The actions an object performs
     - The knowledge an object maintains
     - Major decisions an object makes that affect others
3. Identify collaborators (for each class)
   - Include
     - Collaborations are requests from one object to another
     - Collaborations between a group of objects working together making requests on one another

Submission

On Canvas, submit a single PDF which includes:

- A list of use cases. These can be high-level and do not need to include all steps, failures, and variations.
- Analysis of “probably classes,” “possibly classes,” and “rejected as classes”. Use the noun extraction approach.
- Comment on any relationships identified between classes (i.e. dependency, aggregation, inheritance).
- List of classes, responsibilities, and collaborators. Create CRC cards using the **CRC Card Maker** **(https://echeung.me/crcmaker/)** .

HW 3: Design

Criteria	Ratings			Pts
A list of use cases These can be high-level and do not need to include all steps, failures, and variations.	**5 pts** **Full Marks**	**3 pts** **Partial Marks**	**0 pts** **No Marks**	5 pts
Analysis of “probably classes,” “possibly classes,” and “rejected as classes”.	**5 pts** **Full Marks**	**3 pts** **Partial Marks**	**0 pts** **No Marks**	5 pts
Comment on any relationships identified between classes (i.e. dependency, aggregation, inheritance).	**1 pts** **Full Marks**		**0 pts** **No Marks**	1 pts
List of classes, responsibilities, and collaborators. Create CRC cards using the CRC Card Maker	**14 pts** **Full Marks**		**0 pts** **No Marks**	14 pts
				Total Points: 25