CS 200 **Project 3** **100 points**
Fall 2023 **Nov. 3, 2023 by Noon**

Practical Information

- Ask your questions on Slack or make an appointment to meet with the TA via Zoom.
- Teams: As (Re)Assigned.
- Each team submits only *one full solution* in Blackboard.
- The submission should consist of a simple **HTML file** (called **Report.html**) and any additional file required (.pdf, txt, .jpg, png, and whichever other files you want to submit). It should have explicit references to all submitted files as well as inline inclusion of images. In the **Header table** of the report file, you must specify: the name of the student, student's CWID, student's email address and the time spent on the assignment (in hours, for statistical purposes). Your solution should be entirely included in the body of the report file. Specify any extra assumptions or your own clarifications if needed. Then for each task, describe your solution in the form of a report.
- The Report.html must be a complete report. Failure to produce this report will result in an additional penalty of 5% on your total grade.
- All submitted files should be referred to from within the report file, otherwise they will not be accessible during grading! Images not accessible or visible will have points deducted. Create a ZIP folder with all your files. Submit the ZIP file in Blackboard.
- We will use group submission in this project. **Only one team member (different than Project 1 or Project 2)** shall submit the full solution (ZIP file). Identify the submitter in the Header table of the Report.html.
- Additionally, the Report.html must include a **Task Distribution table**. This section should outline all the tasks that were performed per member and what percentage each team member contributed. If the balance is not evenly spread among all, the grades for each team member will differ based on the percentages.
- Each non-submitting team member **must** have their own HTML file (using the format of FirstNameLastNameTeamXReport.html). This report should be put into the team's ZIP solution., but with only the **Header table**, **Task Distribution table** (affirming that you all agree on it), and **their system Class diagram**.
- The submission tool is Blackboard.
- You may re-submit as many times as you wish before the deadline. The final submission will be graded.

Goals

This assignment will make you familiar with the design of a software system and UML Class & Sequence diagrams. The grading scheme is as follows:

- Image of the Class diagram using UMLentino (40%)
- Image of the Sequence diagrams using UMLentino (45%)
- Discussion of your Design/Architecture based upon lecture slides (10%)
- Bitbucket statistics (5%)
- You can be deducted additional points for missing items, no updates to Projects 1 and 2, Participation %, as well as evenly distributed Bitbucket commits.

- Upload all images, datafiles, source files and result files to Blackboard and provide links to all of them from your Report.html file. Also include any additional information the grader might require to grade the assignment. Your submission should consist of one directory (folder) called: **TeamX_Design** (Ex. Team13_Design). The final submitted **TeamX_Design** ZIP folder should contain:
  - Report.html
  - All Image files
  - All MemberTeamXReport.html files
  - Project 1 folder (corrected to date)
  - Project 2 folder (corrected to date)

Assignments
Given the analysis of the requirements you have performed for the Chocoholics Anonymous (ChocAn) project, **in this assignment, you will perform the design of the system.**

Task 1
Draw the complete **UML Class diagram of the ChocAn system** by **specifying all classes of the system and their relationships**. This will typically include more classes than you had previously identified during the Analysis phase. For each class, **show all attributes** with their type and visibility modifier. You should also **show all methods** that you identified necessary with their full signature (with parameters and return types) and their visibility modifiers. All associations must be clearly marked with name, multiplicities, role names, and directions. Inheritance and composition relationships must be used in your design. Each use case must be initiated by a distinct public method in your class diagram, which may invoke other private ones. Note that if the class diagram is too big, you may use aliases and packages to simplify the reading of the diagram. Write any assumption that you feel will help understand your design decisions.

Task 2
Draw one **UML Sequence diagram per Use Case** you have identified from previous projects. Remember that a **sequence diagram corresponding to a use case is initiated by an actor** who triggers the corresponding public method. Also, recall that a sequence diagram is not a full implementation, but only shows an overview of the methods and messages sent between objects. **Make sure** the sequence diagrams are consistent with your class diagram, meaning any message you send to an object should be a method of the receiver object.

Task 3
To verify your design, **examine the coupling and cohesion issues**. Discuss the coupling and cohesion of your design. Also, **propose and persuade which architecture type** you feel would be best for your project and why. Refer to "*An Introduction to Software Architecture*" reading discussed in lecture. Argue in a paragraph of **300 words (+/-10%) why your design and architecture are "good"**.

Don't forget to use your Bitbucket repository. Commits will be checked in each repo.
Remember to commit/push all your files and diagrams.