

# CEG 4110/6110 Final Project

Fall 2019

Assigned: November 7, 2019

Due: November 17, 2019

## Part 1 - UML Class Diagram:

With the Analysis Specification document completed, you are now tasked with beginning the process documentation, planning, and design of the “Quick-Reports” project. Your first task will be to utilize your team’s Analysis Specification (specifically the high-level modular design) to begin the formal design. You will refine if necessary any modules and then utilize them to create one or more UML Class diagrams. That is, you will create a UML Class diagram for the overall system, or multiple smaller UML Class diagrams which represent sub-systems that connect together.

You will need to think of any functions and their corresponding names that will appear in your system. ***You do not need to include any helper methods, getters and setters, or unnecessary attributes.*** Only include attributes and methods that convey the essential functionality of the class. Where needed, include abstract classes and methods. Accompany the UML Class diagram with a brief explanation of the design.

Furthermore, think about how you will present the UML Class diagram as they may not easily fit on a single page or PowerPoint slide. If they cannot be read/understood, they cannot be graded. When your team is evaluating your UML Class diagram(s), discuss amongst yourselves if any reasonable software engineer is able to understand the classes and purpose of this system?

It is ***NOT*** recommended to have one person of the team create the entirety of the UML Class diagram(s). As there are multiple interconnections of major parts of the “Quick-Reports” software, it is a good idea to have everyone contribute.

***Note:*** These diagrams are not a contract for how your code will look. They serve as an initial approximation of the classes used within the code. The point of these diagrams is to make you think about this problem significantly before ever writing code. If at first you feel like you have no idea where to begin, this is okay and happens frequently! Utilize the provided in-class time and more outside of class if needed to discuss with your group to create, iterate, and refine the UML Class diagram(s).

## Part 2 - Development Plan:

In the second part of the Design document, you will create a plan of action for the implementation of the software as follows:

1. Create a responsibility list for your team. What will each team member be responsible for? What is the communication plan to discuss questions about these responsibilities (email, text message, etc.)?
2. A brief project schedule with the milestones of the project. When will the UML Class diagram be completed? When will the database be finished and tested? Do this for all primary items. Include the tasks associated with these primary items and who is responsible for each task (if known). Flexibility is okay with respect to the due dates.

3. A documentation plan. What supporting documentation will you write (e.g., instruction manual, developer documentation, etc.) and when will the documentation process begin? How will you document your code (e.g., JavaDoc or Prologue comments)? Specifically state what information will be included in each document, including the code.

### **Deliverable Requirements:**

A single word or PDF document of the Design is to be submitted to the corresponding pilot dropbox. The document must be intuitively labeled with the two parts discussed here. Furthermore, you must include a cover page listing the name of your software and group members.

### **Grading:**

This portion of the final project be graded as follows: 60% for the UML Class diagram of Part 1 and 40% for the development plan of Part 2.