COP 2535: Data Structures, Term Project

Step 01, Iteration 1, Requirements

1 Introduction

The COP 2535 Term Project will give you the opportunity to build an application using one of the data structures we will learn about. You may in fact use several. This project is not intended to be a heavy-weight project, and I don't expect you to make an extraordinary on it. However, this could be part of your portfolio, so you could decide to make more of an effort if you choose to do so.

This project follows an iterative, incremental development model. It is *iterative* in that you will repeat the same series of steps four times through the term. It is *incremental* in that you will add to the functionality of your project during each of the four iterations. Each iteration will consist of the same four phases.

Requirements Analysis In this phase, you will determine the functional requirements to be implemented in this iteration. *Functional requirements* refers to what the software will actually do. This is probably the most important phase, because you cannot build anything unless you know what it is that you are building. Your deliverable will typically be one or more Use Cases, or a Functional Requirements Specification.

Design In this phase, you will design the software to be implemented. In some respects, this is the hardest phase. Thhis is where you will make decisions as to *how* to implement the requirements. Your deliverable will typically be in the form of UML diagrams. We will mostly be using Activty Diagrams, which are similar to flow charts.

Implementation In this phase, you will implement your design in code. The deliverable is your source code.

Testing In this phase, you will test your implementation against the requirements. That is, you will answer the question, "Does the code do what the requirements expect?" Your deliverable will typically be a text document containing the output of the code when run.

2 Instructions

Choose a project. We will discuss some examples in class. Your project should meet the following requirements:

- It should be very simple. You will not have very much time to spend working on it, so you should not be too ambitious.
- It should have a text user interface. We will not be doing qraphical user interfaces, so your users will use a text-based interface.
- It should focus on maintaining, creating, modifying, deleting, and retrieving data. The purpose of a data structure is to hold data, so you should choose a project that uses data.

Here are some projects you might want to think about:

- A check book balancer
- A music/book/gun collection

- A daily/weekly/monthly scheduler
- A family tree
- A job application tracker

3 Deliverable

Your deliverable for Iteration 1, Requirement Analysis will be a Use Case. I will cover this in more detail in class and give you some examples. Generally, a Use Case has the following parts:

Actors anyone or anything that performs a behavior (who is using the system)

Stakeholder someone or something with vested interests in the behavior of the system under discussion (SUD)

Primary Actor stakeholder who initiates an interaction with the system to achieve a goal

Preconditions what must be true or happen before and after the use case runs.

Triggers this is the event that causes the use case to be initiated.

Main success scenarios (Basic Flow) use case in which nothing goes wrong.

Alternative paths (Alternative Flow) these paths are a variation on the main theme. These exceptions are what happen when things go wrong at the system level.

A Use Case is a text document. No particular formality is required. The purpose is to (1) describe the function of the software so that it can be tested, and (2) to pass on to the design team so that they can design the software. Here are some useful links.

- `https://www.usability.gov/how-to-and-tools/methods/use-cases.html`
- `https://www.wrike.com/blog/what-is-a-use-case/`
- `https://www.indeed.com/career-advice/career-development/list-of-use-cases-examples`