Final Project Overview Document
EC327 Introduction to Software Engineering, Fall 2022

Assigned: November 9, 2022, Wednesday -- **Due:** December 12, 2021, Monday at 11:59 pm

Total points: 100
Weight: 15% of your final grade

Goals

The goals of this **team project** are to create a unique, team-driven software project with the following characteristics:

- Technically proficient
- Exhibits strong software engineering principles
- Excellently motivated, explained, and documented
- Available to the larger community as an open-source project

Team Size

Teams should be composed of 4-5 students.

Team Roles

The project has **five team roles**:

1. *Project Lead* - This person coordinates all efforts of the project.
   - The project lead is responsible for organizing and maintaining the schedule of all work being done for the project. This includes:
     - Setting up the timeline, milestones, and objectives.
     - Interfacing with all team members to maintain progress during development, including calling and organizing meetings.
     - Completing all required work that other members fail to do (i.e. a catch-all for ambiguous tasks).
   - **Submitted documentation for timelines, objectives, and meetings**. – ***ProjectTimeline.doc***

2. *Specification Lead* – This person is responsible for the specification of the project in terms of performance, functionality, and software architecture. This includes:
   - Checking all design specifications for system-level problems and performing system-level integrated testing.
   - Commenting and readability of the code.
   - Testing regime - thoroughly test your code and provide comments/code for easy testing.
   - **Overall software architecture description and images** – ***ProjectArchitecture.ppt***

3. *Interface Lead* – This person is responsible for how users interact with the software. They will be responsible for the following areas that will be graded as part of the project:
   - Crisp and intuitive - how easy is it for a user to understand what to do; is the feedback (e.g. error messages) clear and complete? ***Scale of 1-10 evaluated by entire course staff.***
   - Robust - is it possible to crash the GUI or have it do inappropriate things (e.g., leave windows or icons on the screen, perform incorrect actions)? ***Scale of 1-10 evaluated by entire course staff.***

- Visually pleasant - does it properly utilize icons, backdrops, and other artifacts to improve the user's experience? Keep in mind that sometimes less is more (and sometimes it isn't). ***Scale of 1-10 evaluated by entire course staff.***

4. *Technical Lead* – This person is responsible for the algorithms, data structures, and core technical aspects of the project. They will be responsible for the following areas that will be graded as part of the project:
   - Figuring out what data should be passed between the GUI and the backend code (if applicable). and implementing the routines that will do it.
   - Testing routines to demonstrate the proper passage of data.
   - Efficiency - the implementation should be efficient enough to provide a good user experience. ***Scale of 1-10 evaluated by select course staff.***
   - Robustness - the back-end should never crash or enter an infinite loop; it should also always provide reasonable data to the other components. ***Scale of 1-10 evaluated by select course staff.***
   - Basic security - there should not be reasonably obvious ways to get the project to do something wrong, - ***Scale of 1-10 evaluated by select course staff.***

5. *Documentation Lead* - This person manages all documentation and marketing for the project. This includes:
   - Setting up a version control system for maintaining all code and documentation used by the project. ***Provide link to course staff showing your repo (GitHub).***
   - Packaging all code into one application, with appropriate resources, and distributing it.
   - Putting together (and writing, if necessary) all documentation for the project, both front-end and back-end, in an organized, concise, and appealing fashion. ***ProjectDocumentation.doc***
   - Getting all team members to agree to a statement of work at the end of the project, describing what each member contributed. ***StatementOfWork.doc***
   - Putting together a short video presentation of the project, including motivation for why the project is useful, who will use it, how they will use, and why they will use it. ***Video on course YouTube page.***

Each task can be handled by one or more people. In four people teams, the recommended organization is to share the documentation task across all members. You can also arrange it such that two people jointly do two tasks, e.g., interface lead and technical lead by two people or project lead and specification lead by two people.

Task Specifications and Grading

Your project will be graded equally on the following three criteria:

1. Front-end

The user-interface for the project must be:

- crisp and intuitive (i.e., no instruction manual required), providing appropriate and understandable feedback,
- robust to all sorts of behavior, normal or otherwise, meaning your project should neither crash nor leave strange elements on the screen,
- visually pleasant, properly utilizing icons, backdrops, and graphical-user artifacts.

2. Back-end

The back-end code should be:

- efficient; it does not have to be optimal, but the user should not suffer a negative experience (e.g. long wait times, dropped information, etc.),
- robust to all manner of input from the user interface. The back-end should never crash or enter an infinite loop,
- secure (at a basic level), in that there should not be reasonably obvious ways to get the back-end to do something malicious,
- well documented (e.g. commented) and easy to understand by programmers with your level of expertise.

3. Marketability

The basic question you should ask yourself is: "Would people use the project?"

- You should specify who might want to use the project, and for what purposes.
- It should be clear how the design decisions in the project conform to the potential use/users of the project.
- Your project should include a concise and clear description targeted toward the class of potential users.
- **Students should make sure to stay away from licensed or copyrighted content (e.g. pictures, games).**

Deliverables

Project teams are to submit:

- All code for all aspects of the project, including instructions for how to compile and run the code from scratch. Include a sufficient number of test cases as part of the instructions.
- **ProjectTimeline.doc**
- **ProjectDocumentation.doc**
- **StatementOfWork.doc**
- **ProjectArchitecture.ppt**
- **Source code repository URL.**
- A **maximum 5-minute** video presentation of the project in action, including its target audience and use, as described above. **These should be uploaded to the course YouTube site (details to come).**