Software Development Team Project

As a team, you will define a problem and design, program, and document a marketable solution. There should be a market, or user demand, for your software. You will submit a thorough report of your development process along with your software, which you will present to the class on completion.

Assessment and Evaluation of Teamwork

The final mark for the assignment or project will be assigned to each team member. That mark may be subject to a peer evaluation process. Where a team fails to function as a cohesive unit, the instructor may intervene and determine another course of mark allocation.

If the project is completed normally, then an agreed upon marking scheme is applied, and students will be evaluated accordingly. In cases where projects fail to complete due to circumstances outside the control of the team then the instructor can apply some other marking scheme to evaluate the participating students based on the completed work.

Each student will be evaluated based on the completed project and must provide evidence of the portion of the work completed by them. To this end, **each student should keep one or more of the following**. This also serves to protect the student's right for a fair assessment should things not go as planned.

1. A log book or diary of activities, like meetings attended, work completed, etc.
2. Source code.
3. Revision information for code and written documents, printouts of unfinished programs or documents.
4. Source documents, like papers, journals.
5. Anything else which would or could convince the instructor that the student has contributed positively to the project. E-mailed documents (dated by a system such as Sandcastle), are very helpful.

**Software Development Life Cycle**

We will loosely follow the Software Development Life-Cycle, in particular, the Waterfall software development model:

* Requirements Analysis (**Problem Definition and Analysis**)
* System Design (**Planning**)
* Implementation & Testing (**Programming, Documentation, Testing**)
* Deployment and Maintenance (**Class Demo/Promotion**)

**Problem Definition (/10)** *Due:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Brainstorm a few ideas, and do some research to help you select the one that is most feasible. Record your project title, project goal, target audience (what demand exists for your proposed program?), and next steps.

**Problem Analysis (/20)** *Due:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. **UML Diagrams, system flowcharts**
2. **Mockups of the user interface**

Understanding the characteristics of your intended users will help you to design a good user interface. You may want to do some research and survey potential users. Diagram all relevent screens.

1. **You will likely need to supplement your models and mockups with a written analysis to include:**

The program objectives and end user requirements (EURs). In other words, what your software will be capable of.

Brief descriptions to accompany each model and mockup.

A brief explanation/pseudocode to outline how you’ll approach the coding.

**Planning (/20)** *Due:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Once your project team has decided on what software you will develop, you will begin planning the stages. What tasks need to be completed? What will be the division of labour among team members? Assign team mmber roles and define timelines for completion of tasks. You will use a **Gantt Chart** to plan and manage your project. You may wish to use the free program available at GanttPro.com or something comparable.

**Programming and Development**

Coding of classes, interface

Check-in Meeting Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (preliminary demo, discuss problems and expectations)

**Documentation**

External User Documentation (User guide with installation instructions, training materials, Help files and FAQs, etc.)

Fully documented program code (doc comments, docstrings, block comments, line comments)

**Testing**

Test with a range of test cases/traces. Ensure that your program meets EURs (Modular/unit and integration testing, normal, erroneous, and boundary testing).

Record all test data, corrective action (screencaps or other documentation), and test reports

**Final Submission (and Group Presentation of Product) (/40)**

Submit a final report including all stages of development, test reports, test data, and software evaluation.

Demo functional program for the class

Submit System User Guide and program source files (with proper internal documentation)

Include packaging, and a promotion/marketing plan

Outline plan for software support and maintenance