

Course Project

This document describes the term project for ECE450S. The software that is the subject of this project is described in a separate set of documents. The purpose of this project is to give you experience in the design and implementation of software using a team approach and modern software development tools. It attempts to simulate many aspects of real world software engineering.

Project Teams

The project will be done by teams of 5 students. (Teams with more or less than 5 members will be allowed only under exceptional circumstances). All students in a team will get the same mark for the work they do unless they unanimously agree (in writing) to an unequal division. The instructor will be the final arbiter with regard to any disputes arising from the composition of teams and/or the distribution of marks to the members of a team. All students on a team must be taking the course for credit.

Project Phases

The project will consist of the following three phases:

Phase A - Understanding a Legacy System (Reverse Engineering) (15%)

The instructor will describe a problem for which a software solution is desired. Throughout the project, the instructor or his designated assistant (the teaching assistants) will serve as the final client of the system. Each team will prepare a document containing the architecture description of the legacy system, the requirements specification for the software system and a preliminary design for the software. Your specification and design documents should include at least the following information:

Design recovery of the legacy system A description of the components of the legacy system, the logical and physical relations between the components.

Requirements specification A specification of the requirements, including environment assumptions, objectives, feasibility and risks, alternatives and costs comparison, etc. A proposal for the structure of your software in terms of modules, functions and procedures or any other building blocks that you deem to be appropriate.

Phase B - Module Implementation Phase (15%)

Each Team will implement the part of the software designated as Phase B Deliverables based on their specification and design. Reports on why a requirement is not met or validated must be part of the deliverables.

Phase C - Swap and Integration Phase (20%)

Each team will select software produced by SOME OTHER TEAM in Phase B. They will continue the implementation of the system adding the Phase C Deliverables to complete the initial implementation of the system.

During this phase, each team should still maintain their exported module to fix bugs fired by the users. The team also tests on the imported modules to find their bugs. The bugs found and fixed will be scored as bonus to the project mark.

Evaluation

Each team will be evaluated on the quality of the software and documentation that they produce in each phase. Errors in the software will be taken as an indication of negative software quality. Each team is responsible for errors in the software they produce independent of whether they created those errors or inherited them.

A small bonus (2.0% of phase's mark) will be awarded each time a team software is selected by some other team at the start of phases C, i.e. the mark for those phases will be calculated as:

$$phaseMark = projectMark \times (1 + \max\{0, 0.02 \times nSelectors - 0.001 \times nNetBugs\})$$

where $nSelectors$ is the number of teams that have selected the software for the next phase, $nNetBugs$ is the number of outstanding bugs found by the users of the component minus the number of bugs you found in imported components. The Instructor reserves the right to modify the project as required to cope with unforeseen circumstances.

Software Swapping

The purpose of swapping is to give you experience in working with software that you did not originate. Each swap will be done on a full information basis. The documentation and source code for each team software will be placed online for inspection by other teams. Each team will be able to examine this information before selecting the software that they will use in the next phase. All decisions concerning software selection will be communicated to the instructor or teaching assistant by email on or before the start of lecture or tutorial on the due date. After all teams have made their selections, all selections will be announced by the instructor. At the start of phases C, each team must select software FROM SOME OTHER TEAM, i.e. they cannot use their own software in the next phase. It is NOT within the spirit or intent of the project to extensively rewrite software that you have selected so that it resembles software you have previously produced.

Project Schedule

It is **VERY IMPORTANT** that all teams adhere to the project schedule given below. Failure to meet a delivery deadline at phase B or C will be penalized at the rate of 10% of the mark achieved **PER DAY** that a team software and documentation is late.

Phase	Due	Swap
A	Feb 11	
B	Feb 25	March 11
C	April 8	

KISS - Keep It Simple Stupid

One of the best strategies for a successful software project is to keep the project goals reasonable relative to the amount of effort available to accomplish those goals. In software engineering this is known as the KISS principle.

DRY - Don't Repeat Yourself

One way to achieve simplicity in the practice is to avoid duplications, known as the DRY principle. Following DRY principle often leads to high reusability and good design.

The instructor has worked hard to keep the software system specified for the project as simple as possible. He understands that a more complicated system would provide greater functionality and user satisfaction. However such a system would also require more time and effort to implement. You are also encouraged to think reasonable in designing and implementing this software system.

Do NOT add functionality beyond what is specified at the expense of large efforts. .