HSC Software Design and Development

Major Project Specification, 2011

Aims of the Project

This project must be done on an individual basis. The aims of this project are as follows:

- To model the processes of the software development cycle. By turning the theory of software development into actions we gain a fuller understanding of these processes.
- To develop communication skills
- To develop a software product of relevance to high school students or their teachers. The product developed must be a computer game that should serve some educative purpose of relevance to students
- To address relevant social, legal and ethical issues that will arise in this process

Project Organisation

- A significant product is required to be produced, so some class time will be allocated for project work. Two periods will be set aside each week to enable students to develop and code their software, as well as developing the required documentation to support the software development process.
- The Year 12 Software design class will have a permanent booking for project work in the A11 lab. However, some of the work for this project will have to be completed outside of class time.
- Each student will be required to complete a weekly project blogg. Each week the blogg will contain an entry describing
    - the work carried out on the project during the week,
    - where you plan to go in following weeks and
    - any significant issues that may have arisen.
    - Your supervising teacher/s will be browsing your blogg entries to check on your individual progress. At the end of the project students' bloggs and reflections will be graded.

Project Deadlines

There are two deadlines to correspond with varies stages of the Software Development Cycle

Task	Description	Weighting	Deadline
Defining and understanding the problem. Planning and Designing (Part 1)	Investigation, presentation & report on the first stage of major project. Report on planning and design major project.	20%	Monday, Week 10, Term 1, 2011
Implementing, Testing, Evaluation and Maintenance	Submission of final product.. Testing and Evaluation of major project	30%	Friday, Week 10, Term 2, 2011

NOTE

* These dates are only indicative and will be finalised early in term1, 2011

Defining, planning and designing

- For this component of the major task, students are expected to investigate a problem that requires solving or a need that is currently not being met.
- Your first task is create a document called "Project Plan". This document will contain the following subsections:
    - Defining the Problem and Its Solution
    - Planning and Design

  The contents of each subsection and the tasks you have to complete is described below

Defining the Problem and Its Solution

- Brainstorm ideas for a project that you think will satisfy the above criteria and that you feel confident in achieving. This should be done with other students in the class group. All students should take notes of the discussion. Under

the heading "Initial Ideas", summarise the discussion that took place and list some of the ideas proposed. Describe a product that you think you want to construct. Why does this project appeal to you?.

- Investigate
    - The needs of users – this will be done by conducting a survey of students at the school. The survey should be structured in a way that meaningful analysis of the results should be possible. Hence concentrate on using closed questions. Survey at least 12 -15 students across at least 3 year groups. Your teacher will arrange for any photocopying you require. Under the heading "The needs of Users" include a copy of the survey, along with your analysis of the survey results (Length: At least 2 paragraphs, H4.1)
    - Determine the objectives of the proposed software product. Objectives are a description of what you are trying to achieve with the product and most importantly, they should be measurable. For example: *" The game should give users effective feedback about their progress"*. Under the heading "Software Objectives" list at least 4 bullet points that describe the objectives of this software (H5.1)

- Think about the student survey responses and your objectives. Think of at least one alternative solution that would still address the needs of the students you have determined. Under the heading "Proposed Solutions", describe both proposed solutions that you believe will address the student's needs (Length: At least 1 paragraph - H6.3)

- Complete a thorough Feasibility Study for both proposed solutions. The Feasibility study must address the following:
    - Is the solution technically feasible?
    - Is it operationally feasible?
    - Is the project financially feasible (Imagine a budget of $20000 for this assignment)?
    - Can it be completed in the required time frame (schedule feasibility)
    - Are their any social and ethical issues to address

  Under the heading Feasibility study, address each of the dot points for both of the solutions that you have determined. Each dot point should be about a paragraph long (H5.2).

- Under the heading "Recommendation" make a recommendation for the development of a particular solution, with a clear justification for your choice and a justification as to the rejection of the alternative solution. (Length: At least 1 paragraph - H4.2)

Planning and Design

- Under the heading "Time Plan" Produce a Gantt chart for all of the tasks to be undertaken to bring the entire project to completion by the date for stage 2. The Gantt Chart must show each of the task names and the duration in weeks. This can be constructed using Excel. (H5.2)

- Under the heading "Interface designs", create a series of screen dumps or diagrams that clearly shows the nature and features of the user interface. Justify the placement of icons, the use of colours, the style of text used, etc (Length: At least 1 paragraph - H5.2)

- Think about the environment of the software product. The environment consists of those items that are not part of the system, but nonetheless influence the system. For example, DET policy prohibiting publication of student photos on a web site. What are the boundaries of the software solution? Under the heading "Software Environment and Boundaries", develop a series of dot points that describe the software boundaries and the software environment.

- Under the heading "Hardware", describe the minimum hardware configuration required for the software product. Justify the configuration you have described. (Length: At least 1 paragraph – H1.1)

- Create a new heading "Data Dictionary" (H 1.3). Create a table that lists
    - All of the variables to be used in the program
    - There data type / structure
    - The purpose of the variable

- Create a new heading "Data Validation" (H 1.3). Choose from the Data Dictionary a selection of variables that are used to store input from the user and describe the validation processes that will be applied to the variables to make sure that the user input is correct and that the software is stable

- The use of top/down design and modularisation is strongly recommended. Under the heading "Structure Diagram", create a structure diagram for the proposed software product. If necessary use a large sheet of butcher's paper to create the structure diagram. This can be hand drawn or created on a computer, however it must be neat and clearly labelled.
- Under the heading "Data Flow Diagram", create a context diagram and a data flow diagram for the entire project. Since you will be using top/down design, you will need to create a new DFD on a separate page for each level process of the design. This can be hand drawn or created on a computer, however it must be neat and clearly labelled.
- Create a new heading "Algorithm" (H4.3). Under this heading document all of the main algorithms used for the modules in your product. Use a mix of flowcharts and pseudocode for all algorithms. If an algorithm is substantially repeated in another module, please note this on the page detailing the algorithm.
- Under the heading "Anticipated Social and Ethical Issues" discuss the issues that may arise as a result of the development and deployment of your software (Length: At least 2 paragraphs – H3.2)

Submission of Work

- Submit a printed and bound copy of the final report that thoroughly documents all of the above work carried out to this point(H6.2)

Advice to Students

Fancy "bells and whistles" will not necessarily guarantee a good grade for the major project. What is important is that you create a software product that

- Satisfies the above criteria
- Is within your ability and experience as a programmer
- Is substantial in nature
- Is based on top/down design
- Is stable
- Uses suitable and effective data validation
- Is complete
- Most importantly, that it works as it is intended

How you are graded for this task will be based on my expectations of you as a programmer and on the demonstrable efforts you have undertaken to complete the task. My advice is to do at least 1-2 hours of work on the project per week. Research and revision of class work related to the project may be necessary. However, it is consistency which is the key. DON'T LEAVE EVERYTHING TO THE LAST MINUTE.

Finally, don't leave the coding of the project to after the first deadline. As soon as you feel confident about the structure charts and algorithms, you should start your coding. And one final tip: code defensively, that is, try to imagine all the ridiculous thing that a complete idiot could do to your software and try to code to prevent these things causing problems.

Remember for stage 1 of the project a printed; hard copy of the report is the only thing that I will accept. CD's, Thumb drives etc, will not be accepted at this stage.

Marking Criteria: Student

Task Description	Criteria	Mark
Initial ideas (H4.1)	• A comprehensive summary of the brainstorming and discussion. Preferred option clearly stated with suitable justification (5 Marks) • Satisfactory summary of brainstorming and discussion. Preferred option is stated though weakly justified (3 Marks) • Missing summary of brainstorming and discussion. Preferred option is vague or poorly justified (1 Mark) • Not documented (0 Marks)	
The needs of users (H4.1)	• The needs of the user are fully and clearly explained and based on evidence gathered from interview or survey (5 Marks) • The needs of the user are explained. Indications that some data gathering occurred (3 Marks) • The needs of the user are poorly explained. Seemingly not based on any evidence (1 Mark) • Not documented (0 Marks)	
Software Objectives (H5.1)	• Clearly defined and measurable objectives appropriate to the problem being solved (5 Marks) • Some objectives described with limited capability for being measured. Some relevance to problem being solved (3 Marks) • Poorly described objectives of little relevance to the problem being solved (1 Mark) • Not documented (0 Marks)	
Proposed Solutions (H6.3)	• Three alternative solutions to the problem clearly described – creative and imaginative (5 Marks) • Three alternative solutions described (3 Marks) • Less then three solution documented or poorly described (1 Mark) • Not documented (0 Marks)	
Feasibility Study (H5.2)	• Thorough and well documented Feasibility Study, evidencing careful thought and planning by the team (10 Marks) • Substantial Feasibility Study, evidencing thought and planning by the team (7 Marks) • Adequate attempt at Feasibility Study. Need for more thought and planning. Some component of the study missing or poorly written. Not enough alternative solutions considered (4 Marks) • Poor attempt at the Feasibility Study; insufficient number of alternative solutions considered. Evidence of little or none thought and planning (1 Mark) • Not documented (0 Marks)	
Recommendation (H4.2)	• Clearly indicated with well expressed justification (3 Marks) • Clearly indicated with some justification (2 Marks) • A solution is selected with limited or poorly expressed justification (1 Mark) • Not documented (0 Marks)	
Time Plan - Gantt chart (5.2)	• Well presented, with reasonable time frames and clear indication of responsibilities (3 Marks) • Well presented with acceptable time frames and some indication of responsibilities (2 Marks) • Poorly presented; unreasonable time frames; Little or no indication of responsibilities (1 Mark) • Not documented (0 Marks)	
Interface Designs)	• Clearly presented screen dumps. Interface designs make creative and interesting use of screen design principles. Justification of the design is thorough and clear. (5 Marks) • Screen dumps are present. Some minor problems with the interface designs that can be easily rectified before proceeding to the design stage. Justification of the design is present and is adequately expressed (3 Marks) • The interfaces are poorly designed. The dumps are not present or are poorly presented.Major redesign of the user interface will be required before proceeding to the design stage of the solution. Justification for the design is superficial or irrelevant. Some aspects of the requirement are not present. (1 Mark) • Not documented (0 Marks)	
Software Environment and Boundaries (5.3)	• Clearly established the software environment and boundaries to be used and provided a suitable justification (5 Marks) • Established the software environment and boundaries to be used and provided some justification for the choice (3 Marks) • The software environment and boundaries is not clearly described. Poor justification of the environment / boundaries. Some elements missing (1 Mark) • Not documented (0 Marks)	

Hardware (H1.1)	• The minimum and ideal hardware environments are well described and evidences thought and understanding of the nature of the development software (5 Marks) • The minimum and ideal hardware environments are described and evidences some thought and understanding of the nature of the development software (3 Marks) • The minimum and ideal hardware environment is superficially I described and evidences little or no thought and understanding of the nature of the development software (1 Marks) • Not documented (0 Marks)	
Data dictionary (H1.3)	• The data dictionary is complete and correctly formatted (4 Marks) • The data dictionary is complete and formatted (2 Marks) • The data dictionary is incomplete and/or not formatted (1 Marks) • Not documented (0 Marks)	
Data Validation (H1.3)	• Clearly described and functional data validation rules that will maintain software stability (5 Marks) • Data validation rules are described and will mostly maintain software stability (3 Marks) • Data validation rules are superficially described and / or will not maintain software stability (1 Mark) • Not documented (0 Marks)	
Structure Diagram (H4.1)	• Structure diagram demonstrates a thorough understanding of top / down design. Diagram fully documents the system and is complete and clearly labelled (10 Marks) • Structure diagram demonstrates a satisfactory understanding of top / down design. Diagram documents most of the system and is mostly complete and labelled (6 Marks) • Structure diagram demonstrates a poor understanding of top / down design. Diagram is confusing or is incomplete or poorly labelled (2 Marks) • Not submitted (0 Marks)	
DFD and Context Diagram (H4.1)	• Context diagram is complete and clearly labelled. DFD follows all conventions and effectively represents the flow of data. Evidence of top/down design influence DFD (10 Marks) • Context diagram is complete and labelled. DFD follows most conventions and represents the flow of data. Evidence of some influence of top/down design on DFD (6 Marks) • Context diagram is incomplete or poorly labelled. DFD does not follows all conventions and does not correctly represent the flow of data. Little or no evidence of top/down design influence DFD (2 Marks) • Not submitted (0 Marks)	
Algorithms (H4.3)	• Clearly structured algorithms that will perform the required tasks of the software in the most efficient manner (10 Marks) • Well structured algorithms that will perform all of the tasks of the software (7 Marks) • Satisfactorily structured algorithms that will perform most of the tasks of the software (5 Marks) • Poorly structured algorithms that will perform some of the tasks of the software (3 Marks) • A superficial attempt at the algorithms that does not address any of the requirements of the software (1 Mark) • Not documented (0 Marks)	
Anticipated social, legal and ethical issues (H3.2)	There is substantial evidence of deep knowledge and understanding of social, legal and ethical issues that may arise as a result of the development of the software product (5 Marks) There is evidence of some knowledge and understanding of social, legal and ethical issues that may arise as a result of the development of the software product (3 Marks) There is limited evidence of knowledge and understanding of social, legal and ethical issues that may arise as a result of the development of the software product (1 Marks) Not documented (0 Marks)	
Final Report (H5.2)	• High quality written communication skills. Superior layout and presentation (5 Marks) • Substantial communication skills with a sound layout and presentation (3 Marks) • Poorly written, with evidence of little or no planning (1 Mark) • Not documented (0 Marks)	
	Final Mark (100)	