CSPE41 Project Assignment: Design and development of a novel software product

Due: **11:59 pm, 26 April 2023** Negotiable. Focus must be on learning and discovering new stuff. Enjoy!

Objective: The objective of this project is to practice software engineering principles followed in Product and Project companies. Towards this end, we will learn SRS (Software Requirements Specification) document, UML design, Technical and User documentation, Code development and Testing.
Note: I would like to make this project happen just like the way it is run in the industry.

General Instructions:
This assignment can be done in a **team of four** (you choose your group member). I encourage team participation, as it will make you feel working in an organizational environment. It contributes **20%** to your final mark – (18% - Project and 2% - attendance in lectures). This project is system based and

- must be developed using C++/Java with classes and objects.
- must meet software documentation standard in all aspects (coding, directory Maintenance etc.)

Search in google 'how to write a good software documentation', '10 tips to write a software documentation' - this shall help you to produce a good document.
Read the software documentation of 'notepad', 'xfig', 'Texmaker'.
As always, please feel free to approach me in person in office hours or by emails, if you have any questions. I'd be happy to help out.

Marking Criteria: Your submission will be marked using the following criteria.

- Novelty of the problem statement.
- Showing good efforts through completed tasks.
- Exploring best functionalities/features of the software and demonstrating creativity and resourcefulness.
- Showing attention to details through a good quality project report and individual review performance.
- 2% attendance Criteria: [91%, 100%] - 2 points, [81% - 90%] – 1 point.

Copying others contents will be seriously viewed, which will lead to heavy penalty for both the donor and the recipient. I encourage you, not to rush at the last moment.

Tasks

Step 1 Problem Identification

Identify a suitable problem of your choice **but not limited** to social concern/cause, issues which NITT faces, issues of global concerns, educational concerns etc. for the development of your software.

Outcome: An abstract mentioning the topic, team name and id, team members, 10-15 lines description of the software developed which includes motivation, existing products, objective, salient features and future scope.

Step 2 Requirements Elicitation

Identify the software and hardware requirements needed to develop the software. List them clearly through a Software Requirement Specification (SRS) document. A sample SRS can be found in this folder with file name *Sample_Software_Requirements_Specification_for Airline flight booking system.pdf*

Step 3 Design of the product (software)

Note: All diagrams should be done in a standard drawing tool. You are free to choose the tool you like. E.g. Microsoft Visio, Lucidchart and IBM Rational Rose etc.

a. Draw all the UML diagrams we have discussed in the lectures relating to the product you have developed.
b. Draw the Data Flow Diagram (DFD) [Context-level DFD, Level 1 DFD, Level 2 DFD] and Entity Relationship Diagram (ERD) relating to the product you develop.
c. **User Interface Design:** Adhering to the three golden rules suggested by Theo Mandel. Mention clearly on how you have adhered to these rules by showing necessary snapshots from your product with a brief write up.
d. **Component Design:** Adherence to at least three basic design principles for designing class based components. Mention clearly on how you have adhered to these principles by showing necessary snapshots from your product with a brief write up.
e. **Coupling and Cohesion:** Identity the types of Coupling and Cohesion you have used in the development of the software. Identify the type and support your explain with a screenshot.
f. **Webapp Design** (Applicable only for those products which are realized as a web application): Out of the six design strategies, identify which ever you have used in the web application. Mention the design technique and support your explanation with appropriate snapshots.

Step 4 Coding

In a typical software development, modular programming approach is practised. If a product requires 5 modules, the team lead will prepare .h file for each module and make it available for each module team. Suppose team M1 wants to use some subroutine of M2, this can be easily achieved through .h files as it contain the name of the function along with necessary arguments. You must separate function declaration and definition. Use header files (.h file) for declaration and (.cpp/.java) file for definition. i.e., apart from standard header files, there should be XXX.h file which will have member functions. The definition for such functions must be included in YYY.cpp/YYY.java.

Your code should have proper indentation and comments. Along with that,

a. Highlight the do's and don't's of good coding style of the implementation issues which you have followed in your development. It is not necessary you should have followed all of them. For each do's and don't's, mention with an example of a code snippet.
b. **Albrecht's Function Point Model:** Compute the number of external inputs, external outputs, external inquiries, external files, internal files in your product and compute the number of function points developed. For F1 to F14, refer the text book and use your own weights.

Tip: Use debugging tool like 'gdb'-many run time errors can be fixed using gdb..Features like..you can peep In to the contents of stack..etc.

Step 5 Testing

Create the following documents:
a. Test Plan - a complete planning document which contains the scope, approach, resources, schedule, etc. of testing activities.
b. Requirements Traceability Matrix - which connects the requirements to the test cases.
c. Test Scenario
d. Test cases and Test Data
e. Cover sheet for a program unit notebook (refer to the slides of the 'Implementation Issues' lecture.

Outcome: Test Document

Step 6 Creation of User Manual with Installation Instructions

Note: A sample User manual can be referred over the web.

User Documentation (user manual-read me file)-how to use the product... will help you to get an idea.

Outcome: User manual and Installation

Step 7 Conclusions and Future Scope

Briefly write 5 line of conclusions and 5 lines of future scope about the software developed.

Step 8 Contributions, Key Take aways, Acknowledgements and References

If it is a team assignment, mention clearly the individual efforts done by each person in the team. I mean, who has worked on what and so on. Be candid in mentioning your contribution. Exaggeration of efforts will lead to penalty. As a team, mention 10 key take away you have learnt from developing this product as part of this course. Briefly mention the people whom you would like to show your gratitude to. List out all the references you have used as part of the development process.

Other Logistics:

This is to be completed and submitted to Moodle. By the due date, submit one zip file containing:

- Prepare two documents. Software Documentation - this document will be prepared by the developer and it will be retained by the company. This will be useful if the company wants to add some more features into the product using a different set of teams. Remember, in a typical product company, as a developer, you will be either part of Major release or Minor release team but not both. Please note the product you develop will be used by or modified by some other development team. This document contains contents of Step 1 to Step 5.
- User Documentation (user manual-read me file)-how to use the product… This document contains contents of Step 6 to Step 8.
- Include any other reference materials you have used in the development of the product.

Use directory structure. i.e., the name of the main directory is << *Team Name_Reg.No#1_Reg.No#2_ Reg.No#3_ Reg.No#4_CSPE41_Project.zip*>> and it should have subdirectories such as Docs (for user and software doc), source (all .cpp/java files), object (all .o files), include (all .h files). Makefile and a.out must be in parent directory.

What next? The project zip file to be submitted by the due date mentioned earlier. Once done, every team needs to appear for a project review on 29 April or on 1 May, 2023 from 9 am to 6 pm in my chamber. I will float the schedule of the review, later, by choosing the teams in a random way with their date and time.

In review:

- You show a demo of your software along with the test cases. You will be evaluated for your efforts individually.
- As a member of the team, you need to understand the entire details of the product including the implementation. You will be evaluated for your knowledge on the fundamental concepts in the course and how you have applied it in the development of the software.

Duration: 10 minutes demo + 10 minutes Q&A for every student. Totally, (10* 4 = 40) + 10 = 50 minutes approx. for a team. Please do not rush at the last moment and save your precious time without waiting in the queue outside my chamber. Remember, there are no waiting chairs outside my cabin!

END OF ASSIGNMENT