

COMPX341-23A Assignment One: Software Design Specifications (SDS)

Due date: 11:59pm Friday 7th April, 2023
Submission: LaTeX PDF via Moodle
Weight in overall grade: 20%

Abstract

You work as a Software Engineer for SoftFlux, a software engineering company. A new and emerging Smart Home development company, called Encost, has approached your company with a proposed project called The Encost Smart Graph Project (ESGP). ESGP is a software system that enables the visualisation of Encost's devices using a graph data structure.

Introduction

Your company (SoftFlux) and the client (Encost) have already agreed on a Software Requirements Specifications (SRS) document. Your task is to continue with the Software Design. Because of the urgency with which the SRS was created, many of the requirements have not been elaborated in detail and you are expected to fill in the gaps to the best of your ability. Your COMPX341 lecturers will play the role of client. You can elicit/clarify requirements with the client using the course's Moodle forum.

A LaTeX template has been provided. Your task is to use the template to compile an engineering report for the project's Software Design Specification (SDS).

Ensure you use appropriate figures, graphs, diagrams, algorithms, tables, etc., so that other divisions of your company (such as construction or testing) can complete their tasks without your direct involvement. Additionally, you need to discuss how your design addresses the outlined software requirements—professional technical reports must always contain discussion; presenting data, algorithms and figures without explanation is not sufficient! Your report also needs to contain a title, a brief introduction that explains the purpose of your document (expanding on required background as needed) and a conclusion. In your design, ensure you explain the functionality and reasoning of any off-the-shelf technologies you propose using as well as competing technologies you discarded and why.

Overall, in the assignment, you are required to do the following.

1 Complete an SDS

Complete an SDS covering the following topics & structure:

- Contents table and clearly separated and titled sections
- Intro/Purpose
- Extra/Specialized Requirements Specification
- Software Architecture:
 - Ensure you include component architecture, process and use case diagrams, and relevant justification;
 - Ensure you clearly explain, with diagrams, the interactions among the components.
- Component Design:
 - Detailed design of each component you outlined in Software Architecture
- A brief conclusion

2 Disambiguate ambiguities

Because of the urgency with which the SRS was created, many of the requirements have not been elaborated in detail or may not be covered. You are encouraged to fill in the gaps to the best of your ability. Your COMPX341 lecturers will play the role of client. You can elicit/clarify requirements with the client using the course's Moodle forum. Should you find any incompleteness or ambiguity you should (1) state what that is and (2) say how you have decided to resolve it. Bonus marks will be given if extra requirement specifications are identified and justified (no more than 10 marks). Note that this assignment is marked out of 100. Bonus marks will only be used to offset the deducted marks.

3 Submit a LaTeX PDF

Submit a PDF document which should be written in LaTeX (in Overleaf) using the SDS template. Your document should not be more than 6000 words—and it does not have to be necessarily that long; that is a limit, not a target. The shortest document that covers ALL the ground is the best document.

Ensure you discuss how each of your design decisions covers the functional and/or nonfunctional requirements of the SRS.

Notes

1. As mentioned in class, five of the stronger SDS submissions will be selected for use with Assignment Two. All selected submissions will be anonymised before being made available to other students.
2. You need to put enough information in the SDS for someone to develop a test plan from, and to develop an implementation from. Put yourself in their position (which you will be in yourself soon enough) and write with those people in mind.
3. Make sure you use a good, clear and consistent naming convention throughout. The names you chose in this document will be those used by the testers and implementors in their work.