**Your Project Title**

**Solution Approach**

Your Sponsor



Sponsor logo (if any)

**Mentor(s):**

(Your mentor’s name(s) )

**Your Team’s Name**

(Provide a list of team members)

Course: CptS 421 Software Design Project I

Instructor: Aaron S. Crandall

**Note**: Recall that this writing assignment says:

Length = 5+ pages text + appendixes as needed. Cover page, table of contents, pictures, tables, images, diagrams do not count for the 5 pages text.

As shown in the Syllabus, this assignment carries a weight of 20% for the final course score.

Posted as a single self‐contained file (no links to outside resources.)

Posted as a PDF file.

Typed single‐spaced.

Typed with black text.

Typed with #11 font size.

Typed using Arial font.

Typed with one inch margins on sides, top and bottom.

**Please erase this page in your final document.**

**TABLE OF CONTENTS**

[I. Introduction 3](#_Toc401658281)

[II. System Overview 3](#_Toc401658282)

[III. Architecture Design 3](#_Toc401658283)

[III.1. Overview 3](#_Toc401658284)

[III.2. Subsystem Decomposition 3](#_Toc401658285)

[III.2.1. [Subsystem Name] 4](#_Toc401658286)

[III.2.2. [Include sections III.2, III.3, etc., for other subsystems] 4](#_Toc401658287)

[IV. Data design 4](#_Toc401658288)

[V. User Interface Design 4](#_Toc401658289)

[VI. Summary of the State of This Project 5](#_Toc401658290)

[VII. Future Work for This Semester 5](#_Toc401658291)

[VIII. Glossary 5](#_Toc401658292)

[IX. References 5](#_Toc401658293)

# Introduction

The introduction begins by stating the purpose of the document. Explain the purpose for providing this design document and specify the intended audience for it. If this is a revision of an earlier document, please make sure to summarize what changes have been made during the revision (keep this discussion brief). Then provide a brief description of your project and state your project goal. .

At the end of the introduction, provide an overview of the document outline.

# System Overview

The system overview contains a general description of the functionality and design of the project. The overview will only briefly describe the overall design considerations and the comprehensive explanations will be done in the sections to follow. The overview should serve as an introduction to these sections.

# Architecture Design

## Overview

This section should describe the overall architecture of your software. The architecture provides the top level design view of a system and provides a basis for more detailed design work. This will be the initial draft of your software architecture. Next semester you will revise this draft and finalize your design.

* Provide a bird’s-eye view of your software architecture. Mention the architectural pattern you adopted in your software and briefly discuss the rationale for using the proposed architecture (i.e., why that pattern fits well for your system).
* Please refer to Chapter 6 in the CptS323 textbook (Dutoit, 2010) to refresh your knowledge on system decomposition and software architectural patterns.
* Briefly describe each layer/component in the architecture and explain its responsibilities.
* Provide a block diagram (e.g., UML component diagram) that illustrates the proposed architecture. The block diagram should show all major subsystems and identify the layers/components in the architecture.

## Subsystem Decomposition

This section explains how you decomposed your system into subsystems. A subsystem typically corresponds to the amount of work that a single developer can tackle. You will show your system decomposition, identify the major subsystems, describe the assignment of functionality to each subsystem, and define the interfaces between them. When you decompose your system into subsystems, you need to take into account the dependencies within and between the subsystems, i.e. cohesion and coupling measures.

* Briefly explain how you decomposed your system into subsystems.
* Discuss the rationale for the proposed decomposition in terms of cohesion and coupling.
* Redraw your architecture diagram (in section III.1) and show all the services each subsystem provides and requires (for example, UML component diagram that uses ball-and-socket notation to depict provided and required interfaces).
* For each subsystem in your architecture, include a sub-section.
* To improve clarity, you may provide multiple figures that show different parts of the architecture (illustrating services) and place each figure right before the corresponding subsection.

### [Subsystem Name]

Include the following sub-sections for each subsystem.

#### Description

Describe the subsystem and identify its responsibilities.

#### Concepts and Algorithms Generated

Discuss the concepts, algorithms or solutions generated and considered for this subsystem. Report the selected solution and explain the solution selection process. Include any special considerations and/or trade-offs considered for the solution approach you have chosen.

#### Interface Description

Provide a description of the subsystem interface. Explain the provided services in detail, and give the names of the required services.

Services Provided:

1. Service name:

Service provided to: [list the receiving subsystems here]

Description: [Describe what the service is and what it does. Provide its input and output values. Briefly describe the major functions that the service provides.]

Services Required:

Names of the required services and the subsystems that provide them.

### [Include sections III.2, III.3, etc., for other subsystems]

# Data design

[You may skip this section if your project doesn’t require any data manipulation or storage]

Describe all data structures (including the internal and temporary data structures), and the database(s) created as part of the application. This information is important from the design point of view as it will help the team in properly understanding all the data structures and databases which will be required for the coding.

# User Interface Design

[You may skip this section if your project doesn’t have a GUI component] – but! If the tools is ever to be used by humans (even just starting and stopping it), there’s some form of user interface design. It can be very simple, but it does exist. Make sure you document how you expect people to use your product, even if it’s just:  
 \* Installation  
 \* Configuration file edits  
 \* Launch daemon by running command x

Provide a detailed description of user interface. The information in this section should be accompanied with proper images showing how exactly you vision the interface to be like (for example mock-ups). Make sure to mention which usecases in your “Requirements Specification” document will utilize these interfaces for user interaction.

# Summary of the State of This Project

Briefly summarize the current state of your project and your team’s progress in building and implementing the prototype. Include a list of the goals achieved.

# Future Work for This Semester

Explain the remaining work (the tasks to be achieved) for your prototype and your plans **for the rest of the semester (ignore spring semester for now)**. Use Gantt chart to clearly identify the work plan, completed portions, and timelines. Also report about how you partition the responsibilities among the team members.

# Glossary

Define technical terms used in the document.

# References

(Dutoit, 2010) [Object Oriented Software Engineering Using UML, Patterns and Java](http://www.amazon.com/dp/0136061257), 3rd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice Hall, 2010.

Cite your references here.

For the papers you cite give the authors, the title of the article, the journal name, journal volume number, date of publication and inclusive page numbers. Giving only the URL for the journal is not appropriate.

For the websites, give the title, author (if applicable) and the website URL.