**System and Software Design Description (SSDD)**

**<Put your project name>**

**<put your team name>**

**Prepared by <Author>**

**<Organization>**

**<Date created or modified>**

# Version History

| Date | Author | Version | Changes made | Rationale |
| --- | --- | --- | --- | --- |
| <mm/dd/yy> | <name> | <version number> | <put where you modify> | <write the reason for changes> |
| <mm/dd/yy> | <name> | <version number> | <put where you modify> | <write the reason for changes> |
|  |  |  |  |  |
|  |  |  |  |  |

# Table of Contents

[Version History 2](#_Toc67408661)

[Table of Contents 3](#_Toc67408662)

[Table of Figures 5](#_Toc67408663)

[Table of Tables 6](#_Toc67408664)

[1. Introduction 7](#_Toc67408665)

[1.1 Purpose 7](#_Toc67408666)

[1.2 Overview 7](#_Toc67408667)

[2. System Overview 8](#_Toc67408668)

[2.1 Design Consideration 8](#_Toc67408669)

[2.1.1 Assumptions 8](#_Toc67408670)

[2.1.2 Constraints 8](#_Toc67408671)

[2.2 System Context 8](#_Toc67408672)

[2.2.1 System Modeling 9](#_Toc67408673)

[2.2.2 Modes of Operation 9](#_Toc67408674)

[3. Architecture 10](#_Toc67408675)

[3.1 System Composition 10](#_Toc67408676)

[3.1.1 Hardware Composition 11](#_Toc67408677)

[3.1.2 Software Composition 11](#_Toc67408678)

[3.2 Design Rationale 11](#_Toc67408679)

[3.2.1 Software Component Design 11](#_Toc67408680)

[3.3 Architectural Styles, Patterns and Frameworks 11](#_Toc67408681)

[4. System Design 12](#_Toc67408682)

[4.1 Structural View 12](#_Toc67408683)

[4.2 Logical View 12](#_Toc67408684)

[4.3 Interaction 12](#_Toc67408685)

[4.4 Interface 13](#_Toc67408686)

[4.4.1 User Interface Design 13](#_Toc67408687)

[4.4.2 Component Interface Design 13](#_Toc67408688)

[4.4.3 External Interfaces 13](#_Toc67408689)

[4.5 Information 13](#_Toc67408690)

[4.5.1 Data 13](#_Toc67408691)

[4.6 Algorithm Design 13](#_Toc67408692)

# Table of Figures

[Figure 1: Use Case Diagram 9](#_Toc67408693)

[Figure 2: Hardware Component Diagram 10](#_Toc67408694)

[Figure 3: Software Component Diagram 10](#_Toc67408695)

[Figure 4: UML Package Diagram 10](#_Toc67408696)

[Figure 5: UML Class Diagram 12](#_Toc67408697)

[Figure 6: UML Object Diagram 12](#_Toc67408698)

[Figure 7: UML Sequence Diagram 12](#_Toc67408699)

[Figure 8: UML Communication Diagram 12](#_Toc67408700)

# Table of Tables

[Table 1: Hardware Component Description 10](#_Toc67408701)

[Table 2: Software Component Description 10](#_Toc67408702)

[Table 3: Architectural Styles, Patterns, and Frameworks 11](#_Toc67408703)

### Introduction

#### Purpose

<< The purpose of this SSDD and its intended audience. (e.g. “This software design document describes the architecture and system design of XX. ....”) >>

#### Overview

<< Provide a description and scope of the software system. >>

### System Overview

<< Provide a general description of the system’s functionalities, context, design and background information. >>

#### Design Consideration

<< This section discusses all the possible design considerations. >>

##### Assumptions

<< Describe any assumptions regarding the software. These may concern such assumptions as:

* Platform: software or hardware, operating systems
* End-user characteristics>>

##### Constraints

<< Describe any global limitations or constraints that have a significant impact on the design of the system's software (and describe the associated impact). Consider the following items as possible constraints:

* Hardware or software environment
* End-user environment
* Availability or volatility of resources
* Standards compliance
* Interoperability
* Interface/protocol
* Data repository and distribution
* Security requirements (or other such regulations)
* Memory and other capacity limitations
* Performance
* Network communications
* Other means of addressing quality goals

>>

#### System Context

<< The context provides a “black box” (with internal decisions hidden) perspective on the design subject. That context is defined by reference to actors that include users and other stakeholders, which interact with the design subject in its environment. It can use the anticipated cases of use of the design subject (hence “use cases” in UML). The design subject’s services may also be associated with actors through information flows. Drawing a boundary separating a design subject from its environment, determining a set of services to be provided, and the information flows between design subject and its environment, is typically a key design decision.

This section should contain:

* One or more UML Use-Case Diagrams showing the processes whereby actors and the system interact in order to accomplish a goal that benefits at least one of the actors.
* For each use case shown in the UML use case diagram(s):
* The assigned identifier (name) of the use case
* The purpose of the use case
* The requirement(s) listed in the SSRD that are (completely or partially) covered by the use case.
* Descriptions of the use case's.

<<UML Use-Case Diagram>>

Figure 1: Use Case Diagram

>>

##### System Modeling

<< A use case diagram to show the overall interactions between actors and systems>>

##### Modes of Operation

<< Please describe all the possible modes that system will operate on. Please also describe each mode in details. Create sub-index to describe each mode if needed.>>

### Architecture

<< This section should describe the architectural view of the software system. The sub-sections should describe composition of the system. The system might contain sub-systems, components, or any COTS products. It often contains the following diagrams

* UML hardware component class diagram
* UML software component class diagram

It also provides descriptions of the hardware components and software components for the system architecture. Any design rationale and architecture styles or framework should also be written.>>

#### System Composition

<<Types of constituents of a system: subsystems, components, modules; ports and (provided and required) interfaces; also libraries, frameworks, software repositories, catalogs, and templates. This section can also discuss possible functional (logical) decomposition, and runtime (physical) decomposition.

<<Hardware Component Diagram>>

Figure 2: Hardware Component Diagram

<<Software Component Diagram>>

Figure 3: Software Component Diagram

<<UML Package Diagram>>

Figure 4: UML Package Diagram

Table 1: Hardware Component Description

|  |  |
| --- | --- |
| Hardware Component | Description |
|  |  |
|  |  |
|  |  |

Table 2: Software Component Description

|  |  |
| --- | --- |
| Software Component | Description |
|  |  |
|  |  |
|  |  |

>>

##### Hardware Composition

<< Use a UML diagram to illustrate the hardware components and its configuration. Please also describe the interactions between actors and the hardware. >>

##### Software Composition

<< Use a UML diagram to illustrate the software components and its configuration. Please also describe the interactions between actors and the software. Please also describe each software component in details. >>

#### Design Rationale

<< Describe any design rationale in this section>>

##### Software Component Design

<< Please describe the design details that contain the following aspects

* Component Name
* Purpose of the component
* Interfaces of the component
* Behaviors of the component

Above items can be described by using UML diagrams.

>>

#### Architectural Styles, Patterns and Frameworks

<< Describe any implementation architecture styles (e.g. 3-tier architecture), patterns (e.g. client-server), or frameworks (e.g. Java and CORBA) used to describe the system architecture.

Table 3: Architectural Styles, Patterns, and Frameworks

|  |  |  |
| --- | --- | --- |
| Name | Description | Benefits, Costs, and Limitations |
|  |  |  |
|  |  |  |
|  |  |  |

>>

### System Design

#### Structural View

<< Static structure (classes, interfaces, and their relationships) Reuse of types and implementations (classes, data types). The section should use UML class diagrams or UML object diagrams to describe its logical viewpoint.

<<UML Class Diagram>>

Figure 5: UML Class Diagram

<<UML Object Diagram>>

Figure 6: UML Object Diagram

>>

#### Logical View

<< Please describe the logical view for each component. This can be done by using UML sequence diagram

<<UML Sequence Diagram>>

Figure 7: UML Sequence Diagram

>>

#### Interaction

<<This section describes all the communications between modules or sub-systems. This might contain messaging, data, and…etc. It should use the UML sequence diagram or UML communication diagram to describe the interactions.

<<UML Communication Diagram>>

Figure 8: UML Communication Diagram

>>

#### Interface

##### User Interface Design

<< Describe interface user and the system. This interface design should show layout of the user interfaces. This section suggests to use a screenshot of prototype or a wireframe of the UI to show the design.>>

##### Component Interface Design

<< Describe interface design between components. This might include (not limit to) the following.

* Function (message) interface
* API
* Protocol

>>

##### External Interfaces

<< Describe any interfaces that exist with external systems that are not within the scope of the system being designed>>

#### Information

##### Data

###### Data Description

<< Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.>>

###### Data Dictionary

<< If you provided an OO description, list the objects and its attributes, methods and method parameters. You might also need to use an ER diagram or a schema if a database is used.>>

#### Algorithm Design

<< Discuss any algorithm design that is specific to the system and requirement. >>