System Design Specification

# TeamName

Student – 1

Student – 2

Student – 3

Date of Submission

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Section/Sub-section Title** | **Author**  **[person with primary responsibility for this section]** | **Page #** |
| **1. Overview**  **1.1 Specification Purpose**  **1.2 Project Scope**  **1.3 Deployment Diagram** |  |  |
| **2. System Architecture**  **2.1 Architectural Design**  **2.2 Architectural Style(s) & Trade-off Analysis**  **2.3 Decomposition Approach**  **2.4 Sequence Diagrams** |  |  |
| **3. Data Design** |  |  |
|  |  |  |
| **4. Software Design**  **4.1 Class Title**  **4.2 Title of Next Class (and so on)** |  |  |
| **5. User Interface Design**  **5.1 User Interface Design Overview**  **5.2 User Interface Navigation**  **5.3 Use Cases Interface**  **5.3.1 Use Case Title**  **5.3.2 Next Use Case (and so on)** |  |  |
| **Appendix - Requirements Traceability Matrix** |  |  |

### General Instructions

* This document must contain all the following sections and subsections.
* Maintain the form and format of the template - font, margins, paging, etc.
* This is a generic template and some sections will not apply to this course project. When this is the case, simply comment why a section is not applicable to this project.
* Use bullets, figures, & tables as much as possible to make the presentation concise and readable.
* All diagrams, tables, & figures need to be numbered by section [i.e. Table 4.1] and have either a title or caption. Be consistent in the use of a title or caption.
* Ensure your terms are used consistently throughout the document. For example, don’t use task leader in one section and have the same role called team leader in another section.
* Do not use ambiguous terms such as user. Specify a particular role.

### Remove ALL

### highlighted instructions and

### instructions delimited by < > from your final deliverable.

# Overview

## Specification Purpose

*< Describe this System Design Specification* ***document’s*** *purpose and its intended audience.>*

* + 1. **Purpose**
    2. **Audience**

## Project Scope

*< Describe the scope of the* ***software*** *to be produced in terms of:* [*Goals-vs-objectives*](https://dl.dropboxusercontent.com/u/60888300/CurrentCourses-Public/Common-Deliverables/ResearchPaper/Goals-vs-objectives.pdf)*>*

* + 1. **Goals**
    2. **Objectives**
    3. **Benefits**

## Deployment Diagram

*< Provide a*[***deployment***](http://www.agilemodeling.com/artifacts/deploymentDiagram.htm) *diagram* ***with description*** *which shows the system’s physical layout.   
In other words, indicating which software components run on which pieces of hardware>*

# System Architecture

## Architectural Design

*< Provide a* ***diagram*** *showing the*

* *major software subsystems,*
* *data repositories and*
* *their interconnections.*

*Explain the diagram. The main purpose is to gain a general understanding of how the system was decomposed, and how the individual parts work together. >*

## Architectural Style(s) & Trade-off Analysis

*< Using the* [*architectural style(s)*](https://www.dropbox.com/s/jqqoz2i8b0weqrb/P%26A%20Ch-05.pdf?dl=0) *scoring matrix below identify the architectural style(s) selected for each major subsystem from the section above (if each subsystem is different) or the single style used for the system as a whole.*

*In each cell* ***score*** *the style and* ***explain*** *the rationale for selecting or not selecting the architecture including*

* *critical issues and*
* *trade/offs that were considered. >*

|  |  |  |  |
| --- | --- | --- | --- |
| **Styles** | **Major Subsystem Name** | **…** | **Major Subsystem Name** |
| Pipes and Filters |  |  |  |
| Client-Server |  |  |  |
| Peer-to-Peer |  |  |  |
| Publish-Subscribe |  |  |  |
| Repositories |  |  |  |
| Layering |  |  |  |

*Score from 1-5 with 1 the least fit and 5 the best fit*

## Decomposition Approach

*<Explain the* [*decomposition approach*](https://www.dropbox.com/s/jqqoz2i8b0weqrb/P%26A%20Ch-05.pdf?dl=0) *used to decompose the system into the major subsystems from the section above. Why was this approach used.***>**

## Sequence Diagrams

*<Develop* ***and describe*** *a System-level* [***sequence diagram***](http://www.agilemodeling.com/artifacts/sequenceDiagram.htm) *[interaction among the Use Cases.>*

# Data Design

*<Provide* ***Data Flow Diagrams*** *- [*[*Example 1*](http://www.agilemodeling.com/artifacts/dataFlowDiagram.htm) *&* [*Example 2*](https://www.visual-paradigm.com/tutorials/data-flow-diagram-example-food-ordering-system.jsp)*]* ***with descriptions*** *to show what data is operated on by which process as the data passes through the system.*

*Provide a Level 0 and a Level 1 DFD. Any notation format is acceptable.>*

# Software Design

* 1. **Class Overview**  
     *<if your software design includes multiple classes, at this point provide a diagram* **with description** *that shows the relationship(s), such as inheritance, among/between these classes. This diagram* ***need not show*** *a complete set of attributes and methods for each class, but rather only enough information to explain the class relationships. The subsections below will include diagrams that provide a complete description of each class.>*
  2. **Individual Class Title**

*<Provide a class diagram* **with descriptions** *of the attributes and methods.>*

* + 1. **Class Description**
    2. **Attributes**
    3. **Methods**

*Note: Class diagrams in a SRS are conceptual in nature [shows that a class exists and what are its’ attributes]. SDS class diagrams, however, provide* ***enough detail to implement the class*** *[generally this means a fleshing out of the operations/methods].*

*Note: Attributes are the information stored about an object (or at least information temporarily maintained about an object), while methods are the activities an object or class perform.*

*[See* [*http://www.agilemodeling.com/artifacts/classDiagram.htm#DesignClassDiagrams*](http://www.agilemodeling.com/artifacts/classDiagram.htm#DesignClassDiagrams) *-Section 3 to provide more info about Class Diagrams]*

*Remember Roles are not necessarily Classes, it depends on your project.*

*However, you will certainly use Roles in Section 5.*

**4.3** Title of Next Class (and so on) …

# User Interface Design

## User Interface Design Overview

*<Provide a high-level description of the user interface for this software application.>*

## User Interface Navigation

*<Provide a diagram and a* ***description*** *of the navigation hierarchy that illustrates how a user moves through the user interface [format this diagram as you wish]. In the lower right-hand corner* ***of each screen*** *in this diagram, place the subsection in 5.3 where this screen is fully described [i.e. 5.3.1.2].>*

## Use Case(s) Interface

*<Customize the following subsections to document each use case that requires an interface.>*

*Note: The various use case interfaces listed and described in section 5.3* **must match** *the screens shown in the Section 5.2 Navigation Hierarchy.*

### Use Case Title

### 5.3.1.1 First user interface for Use Case 5.3.1

*< For each user interface in this use case provide*

1. *a* ***description****,*
2. *a* ***screen image****,*
3. *a* ***fields table*** *that includes descriptive information for each field that appears on the user interface screen. If the data is selected from a pick list, include the list of possible values or their description. If the content of a field is derived from client-side calculations using other fields or values, then specify the algorithm for the calculation. If the content of a field is derived from server-side calculations or lookups, then specify the source of that calculation (e.g., the class or stored procedure where the calculation occurs).*
4. ***specify the error messages*** *to be displayed when the input does not meet requirements for the field. >*

**5.3.2** Next Use Case Title (and so on) …

# Appendix - Requirements Traceability Matrix

*<Develop a table that maps* ***all*** *the system requirements documented in the SRS Requirements* *Traceability Matrix to the design elements documented in the SDS via the Requirements Number.>.*

***Do not embed*** the matrix table in this document, but rather ***link*** to the ***separate document*** created for the SRS that contains the matrix table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Priority**  **Required=1, Desired=2, Optional=3** | **Requirement # by Category** | **Description** | **SRS Section** | **SDS Section** |
|  |  |  |  | **[not required till Detailed SDS]** |

Grading Criteria

**To *maximize* your grade ensure you can answer yes to all these questions**

* Is the Deliverable organized & structured as required by the template– **YES/NO**
* Was the Deliverable Template used and used correctly -- **YES/NO**
* Was the Table of Contents compete **– YES/NO**
* Were all sections present and adequate – **YES/NO**
* Were **ALL** the requirements listed in the SRS Requirements Traceability Matrix included in the SDS Requirements Traceability Matrix – **YES/NO**