Software Technical Requirements Document

for Replace with Your Product

Replace with Your CLC Group Members

Grand Canyon University: CST-326

Replace with Due Date   
(Make sure all yellow is removed before submission.)

Document Version <X.X>

<date>

**AUTHORS**

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Department** |
|  |  |  |
|  |  |  |

**DOCUMENT HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Document Revision Description** | **Document Author** |
| 10/12/22 | 1.0 | What is a summary of the revision | Bill Hughes, Sponge Bob |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**APPROVALS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Approval Date** | **Approved Version** | **Approver Role** | **Approver** |
| 10/12/22 | 1.0 |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 4](#_Toc134441706)

[1.1. Purpose of the Document 4](#_Toc134441707)

[1.2. Related Documents 4](#_Toc134441708)

[1.3. Terms/Acronyms and Definitions 4](#_Toc134441709)

[2. Risks and Assumptions 4](#_Toc134441710)

[3. Architecture 4](#_Toc134441711)

[3.1. System Architecture 4](#_Toc134441712)

[3.2. Application Architecture 4](#_Toc134441713)

[4. Requirements 5](#_Toc134441714)

[4.1. Hardware Requirements 5](#_Toc134441715)

[4.2. Software Requirements 5](#_Toc134441716)

[4.3. Documentation/Comments 5](#_Toc134441717)

[4.4. Error Checking/Handling 5](#_Toc134441718)

[4.5. Software Versioning and Storage 6](#_Toc134441719)

[4.6. Data Requirements 6](#_Toc134441720)

[4.7. File Structure 6](#_Toc134441721)

[4.8. Schema Objects 6](#_Toc134441722)

[5. Integration Requirements 6](#_Toc134441723)

[6. Performance 6](#_Toc134441724)

[7. Software Quality 6](#_Toc134441725)

[8. Technical Requirements 6](#_Toc134441726)

[9. Open Issues 7](#_Toc134441727)

[10. Appendix 7](#_Toc134441728)

# Introduction

< Identify and describe the business need(s) or problem(s) this document will address. Include background information here.>

## Purpose of the Document

< Describe what a technical specification document is and its intended purpose for the audience.>

## Related Documents

*<*Add any related documentation that is relevant and related to the TSD. An example is the Functional Specs.>

## Terms/Acronyms and Definitions

<State any terms and their definitions that are employed in the functional specifications. Also include any acronyms that are mentioned in the document.>

|  |  |  |
| --- | --- | --- |
| **Term/Acronym** | **Definition** | **Description** |
| FSD | Functional Spec Document |  |

# Risks and Assumptions

<List any assumed factors and identified risks that could affect the technical design of the system. Included can be third party or commercial components that will be used, issues around the operating environment, or any constraints. Product potential solutions.>

# Architecture

<Provide a short summary description of the hardware and software architectures.>

## System Architecture

<Provide any appropriate graphical representations that are relevant to the system and project. Add as many as needed. For example, data flow diagram, sitemap, process flow that relates to the technical flow of data, etc.>

## Application Architecture

<Describe the patterns and techniques used to design and build the software application.>

# Requirements

<Provide a short summary description of the hardware and software requirements.>

## Hardware Requirements

<List and identify all hardware related items for the project. For example, what are the minimum requirements of the computer that this project will run under (Windows 10, 16GB Ram etc.)? If there is a hosting environment, what are the hosting requirements? What are the targeted hardware platforms (Windows, Mac, Unix)? Make sure to address server and client-side requirements.>

## Software Requirements

<List and identify all software related items this project will require. What is/are the software that will be used to design this project? This ensures everyone is using the same platform with the same version number. Also, when the project is complete, what are, if any, the software requirements that need to be on the hosted platform to ensure this project runs correctly? Make sure to address servers and client-side requirements.>

## Documentation/Comments

<What type of documentation and comments are required by each designer?>

Example: There can never be too much documentation/too many comments when writing your code. Best case scenario is that every line of code should be documented. Don't assume the next person looking at your code will have the same thought process you had during the design phase.

## Error Checking/Handling

<This is where you can explain the error conditions/exceptions that normally happen in interfaces or cross-flow system integrations. Explain the nature of exception, error id, root cause of the error, and also the strategy to handle the scenario. You can also indicate if there are any concurrent programs designed to automatically handle the error records or error conditions. State if there are any error reports generated or notifications utilized to alarm the support teams and systems administrators during interface failures or outages.>

Example: All software must handle exceptions and errors to prevent app crashes.

Use exception handling code (try/catch blocks) appropriately.

All data going into a database table should be checked to confirm the data is in the appropriate format. If it is not, try and convert it. If error or exception still occurs, flag the user of the error.

## Software Versioning and Storage

<Will the team be using Git or some other repository to manage software storage and versions?>

## Data Requirements

<Define the requirements for any data this application will use. Will it be MySQL, MSSQL, Mongo, etc.? >

## File Structure

<How should the files be structured for the app?>

## Schema Objects

<Provide the database schema.>

# Integration Requirements

<Identify the integration needs and state all required interfaces with anything external to this solution, including hardware, software, and users. Include architectural overview diagrams, high-level data flow diagrams, table structures and schema, interface protocols, APIs, error conditions, error validations and messaging needs, auto processing requirements, etc. You can optionally state hardware and software dependencies, upgrade requirements, compatibility issues with existing frameworks and solutions, etc.>

# Performance

<Specify performance requirements this app must meet.>

# Software Quality

<List other characteristics crucial to the success of this product. List each by describing its relation to the product, being quantitative, specific, and verifiable.>

# Technical Requirements

| **FR**  **ID** | **TR**  **ID** | **Technical Requirement** |
| --- | --- | --- |
| FR | TR-1 |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Open Issues

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Issue ID** | **Issue** | **Raised By** | **Raised On** | **Solution/ Decision** | **Resolved By** | **Resolved On** | **Status** |
|  |  |  |  |  |  |  |  |

# Appendix