Software Design Specification

For Wolf House Points System

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 01/12/19 | 1.0 | Initial Report | Gabriela Braga |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Logical Architecture 5

2.1 Overview 5

2.2 <Package.Name> 5

3. Detailed Description of Components 6

3.1 Class Diagram 6

3.1.1 <Class.Name> 6

3.2 Interaction Diagrams 6

3.2.1 <InteractionDiagram.DiagramType>Diagram 7

3.3 Relational Database Schema 7

3.3.1 <Reliability Requirement One> 7

4. Design Rationale 8

Software Design Specification

# Introduction

[The introduction of the SDS should provide an overview of the entire SDS. It should include the purpose, scope, definitions, acronyms, abbreviations, references and overview of the SDS.]

## Purpose

[Full description of the main objectives of the SDS document.].

## Scope

[A brief description of the software application that the SDS applies to; the feature or other subsystem grouping; what Class Diagram(s) it is associated with, and anything else that is affected or influenced by this document.]

## Definitions, Acronyms and Abbreviations

[This subsection should provide the definitions of all terms, acronyms, and abbreviations required to interpret properly the SDS.  This information may be provided by reference to the project Glossary. Alphabetize.]

## References

[This subsection should provide a complete list of all documents referenced elsewhere in the SDS. Each document should be identified by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document. This should at least reference the SRS.]

## Overview

[This subsection should describe what the rest of the SDS contains and explain how the SDS is organized.]

# Logical Architecture

[NOTE: The main focus in this section is the high-level design. The reader should come away with a good view of exactly how your solution is to be organized.]

## Overview

[This subsection will introduce the various components and subsystems. Insert here your main use case diagram.]

## <Package.Name>

[Insert here the class packages. Repeat sub-section 2.2 for as many packages as your system has. For each package, please provide the package documentation and package diagram.]

# Detailed Description of Components

[NOTE: The main focus in this section is the detailed design. This section will provide most of the basis for implementing the product.]

## Class Diagram

[Insert your class diagram here.]

*Figure X: <ClassDiagram.Name>*

### <Class.Name>

This sub-subsection should be repeated for as many classes as your diagram has so that in here the documentation, all attributes and all operations of each class are documented.]

#### <Class.Name> Attributes

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| <Attribute.Name> |  | <Attribute.Documentation> |
|  |  |  |
|  |  |  |
|  |  |  |

#### <Class.Name> Operations

|  |  |  |
| --- | --- | --- |
| Name | Returns | Description |
| <Operation.Name> |  | <Operation.Documentation> |
|  |  |  |
|  |  |  |

#### <Class.Name> Design Specification/Constraints

[If any, describe them here.]

#### <Class.Name> States and Transitions

[Not all classes will need to have a diagram in here.]

*Figure K: <ClassName.StateActivityDiagram.Name>*

## Interaction Diagrams

[Insert here the collaboration and sequence diagrams.]

### <InteractionDiagram.DiagramType> Diagram

*Figure Y:<InteractionDiagram.Name>*

## Relational Database Schema

### <DatabaseSchema> Diagram

*Figure Z:<DatabaseSchema*>

# Design Rationale

[This section will be used to explain your design from a critical standpoint. This section can also capture good ideas that were abandoned and the reasons for leaving them out of the design.

Use this section to motivate any decisions that will help the reader understand the design that your team is using. No design is perfect. Thus you are being asked to explain the decisions made to support your team’s design. You should concentrate on the tradeoffs you may have made. Here are some guidelines:

A design that is very easy to maintain, may not be the most efficient. Explain this tradeoff in the context of your team’s design.

You may have chosen to store information as opposed to re-generating it on demand (or the other way around.) Why was this decision made? Was it for efficiency? Was it for storage considerations?

When allocating responsibilities among all classes, you may have had a hard time deciding to which specific class assign certain responsibilities. These are important design decisions that need to be explained.]