<PRODUCT NAME>

A

Software Design Document (SDD)

On

"<Product Name>"

Version X.X approved

Month-YEAR

Contents

1. Introduction

1.1. Purpose

<Identify the product whose design details are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SDD, particularly if this SDD describes only part of the system or a single subsystem.>

1.2. Document Conversion

<Describe any standards or typographical conventions that were followed when writing this SDD, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level design segments are assumed to be inherited by detailed Modules, or whether every design statement is to have its own priority.>

1.3. Intended Audience and reading suggestions

Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SDD contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4. Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SDD that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

1.5. Document Copyrights

1.6. References

<List any other documents or Web addresses to which this SDD refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. System Architecture

2.1. Architectural Design

<Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level of how responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don't go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.>

2.2. Functional Design

<Develop a use case diagram and each function in the Use case diagram are connected with the other function and the main architecture of the module and then to the other modules. Do not go into so much details about the functions. Just mention the function with the connectivity to the other functions and how they are connected to the software itself.>

2.3. Data Design

<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.>

2.4. Component Design

<In this section, we take a closer look at what each component does in a more systematic view. Give a detailed description for each module as mentioned below.>

2.4.1. <Module X or "Name"

2.4.2. <Module X or "Name" …etc.

3. Human Interface Design

3.1. Overview of User Interface

<Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback Information that will be displayed for the user.>

3.2. Screen Images

<Display screenshots showing the interface from the user's perspective. These can be hand drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)>

3.3. Screen Objects and Actions

<A discussion of screen objects and actions associated with those objects.>

4. Requirements Matrix

<Provide a tabled information that matches each of the system components with each of the requirements with no explanation. You can use the SRS document for the project to get all of the requirements.>