**Software Design Document (SDD) Template**

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions has been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report.

(**You do not need to include this page on your document**)

Due on: 9/04/2019 (Tuesday, Lab Time)

For: Both Group A and B

**(Project Title)**

**Software Design Document**

Name (s):

**TABLE OF CONTENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1.** | **INTRODUCTION** | | | |  |
|  | 1.1 | *Purpose* | | |  |
|  | 1.2 | *Scope* | | |  |
|  | 1.3 | *Overview* | | |  |
|  | 1.4 | *Reference Material* | | |  |
|  | 1.5 | *Definitions and Acronyms* | | |  |
|  |  |  | | |  |
| **2.** | **SYSTEM OVERVIEW** | | | |  |
|  |  |  | | |  |
| **3.** | **SYSTEM ARCHITECTURE** | | | |  |
|  | 3.1 | *Architectural Design* | | |  |
|  | 3.2 | *Design Rationale* | | |  |
|  | 3.3 | *Architectural Goals and Constraint* | | |  |
|  |  | *3.3.1* | | *Security* |  |
|  |  | *3.3.2* | | *Persistence* |  |
|  |  | *3.3.3* | | *Reliability/Availability* |  |
|  |  | *3.3.4* | | *Performance* |  |
|  | 3.4 | *System Logical View* | | |  |
|  | 3.5 | *System Process View* | | |  |
|  | 3.6 | *Modular Decomposition View* | | |  |
|  |  |  | | |  |
| **4.** | **DATA DESIGN** | | | |  |
|  | 4.1 | *Data Description* | | |  |
|  |  | *4.1.1* | *Logical Data Model* | |  |
|  |  | *4.1.2* | *Physical Data Model* | |  |
|  | 4.2 | *Data Flow* | | |  |
|  | 4.3 | *Data Dictionary* | | |  |
|  |  |  | | |  |
| **5.** | **COMPONENT DESIGN** | | | |  |
|  | 5.1 | *Analysis Classes* | | |  |
|  |  | *5.1.1* | *Class diagram* | |  |
|  |  | *5.1.2* | *Collaboration diagram* | |  |
|  |  | *5.1.3* | *Sequence Diagram* | |  |
|  |  | *5.1.4* | *State diagram* | |  |
|  |  | *5.1.5* | *Activity Diagram* | |  |
|  | 5.2 | *Design classes (GUI, OS, Data)* | | |  |
|  |  | *5.2.1* | *Class Diagram* | |  |
|  |  | *5.2.2* | *Interface* | |  |
|  |  | *5.2.3* | *Resource* | |  |
|  |  |  |  | |  |
| **6.** | **HUMAN INTERFACE DESIGN** | | | |  |
|  | 6.1 | *Overview of User Interface* | | |  |
|  | 6.2 | *Screen Images* | | |  |
|  | 6.3 | *Screen Objects and Actions* | | |  |
|  |  |  | | |  |
| **7.** | **USER REQUIREMENT VS COMPONENT TRACEABILITY MATRIX** | | | |  |
|  |  |  | | |  |
|  |  |  | | |  |

**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1.** | **INTRODUCTION** | | | | |
|  | **1.1** | ***Purpose*** | | | |
|  |  | Identify the purpose of this SDD and its intended audience. (e.g. “This software design document describes the architecture and system design of XX. ….”). | | | |
|  | **1.2** | ***Scope*** | | | |
|  |  | Provide a description and scope of the software and explain the goals, objectives and benefits of your project. This will provide the basis for the brief description of your product. | | | |
|  | **1.3** | ***Overview*** | | | |
|  |  | Provide an overview of this document and its organization. | | | |
|  | **1.4** | ***Reference Material*** | | | |
|  |  | List all documents, if any, which were used as sources of information for the test plan. | | | |
|  | **1.5** | ***Definitions and Acronyms*** | | | |
|  |  | Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience. | | | |
|  |  |  | | | |
| **2.** | **SYSTEM OVERVIEW** | | | | |
|  |  | Give a general description of the functionality, context and design of your project. Provide background information if necessary. | | | |
|  |  |  | | | |
| **3.** | **SYSTEM ARCHITECTURE** | | | | |
|  | **3.1** | **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 overview 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. | | | |
|  | **3.2** | **Design Rationale** | | | |
|  |  | Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that was considered. You may discuss other architectures that were considered, provided that you explain why you didn’t choose them. | | | |
|  | **3.3** | **Architectural Goals and Constraint** | | | |
|  |  |  | | | There are some key requirements and system constraints that have a significant bearing on the architecture. |
|  |  | **3.3.1** | | | **Security** |
|  |  |  | | | Security architecture is the design artifacts that describe how the security controls (security countermeasures) are positioned and how they relate to the overall systems architecture. |
|  |  | **3.3.2** | | | **Persistence** |
|  |  |  | | | Persistence refers to the characteristic of [state](https://en.wikipedia.org/wiki/State_(computer_science)) that outlives the [process](https://en.wikipedia.org/wiki/Process_(computing)) that created it. This is achieved in practice by storing the state as data in  [storage](https://en.wikipedia.org/wiki/Computer_data_storage). Programs have to transfer data to and from storage devices and have to provide mappings from the native [programming-language](https://en.wikipedia.org/wiki/Programming_language) [data structures](https://en.wikipedia.org/wiki/Data_structure) to the storage device data structures. |
|  |  | **3.3.3** | | | **Reliability/Availability** |
|  |  |  | | | The term [reliability](https://whatis.techtarget.com/definition/reliability) refers to the ability of a software component to consistently perform according to its specifications. Availability is the ratio of time a system or component is functional to the total time it is required or expected to function. |
|  |  | **3.3.4** | | | **Performance** |
|  |  |  | | | Performance of software is examined on its responsiveness, stability, scalability, reliability, speed and resource usage of infrastructure. |
|  |  |  | | | |
|  | **3.4** | **System Logical View** | | | |
|  |  | Description of the logical view of architecture. Describes the most important classes, their organization in service packages and subsystems, and the organization of these subsystems into layers. Also describes the most important use-case realizations, for example, the dynamic aspects of the architecture. Class diagrams may be included to illustrate the relationships between architecturally significant classes, subsystems, packages and layers | | | |
|  | **3.5** | **System Process View** | | | |
|  |  | A description of the process view of the architecture. Describes the tasks (processes and threads) involved in the system's execution, their interactions and configurations. Also describes the allocation of objects and classes to tasks. | | | |
|  | **3.6** | **Modular Decomposition View** | | | |
|  |  | Provide a decompositionofthe subsystemsinthe architecturaldesign. Supplement with text as needed. You may choose to give a functional description or an object oriented description. For a functional description, put top­ level data flow diagram (DFD) and structural decomposition diagrams. Also include state transition diagram, sequence diagram, activity diagram whenever appropriate. For an OO description, put subsystem model, object diagrams, generalization hierarchy diagram(s) (if any), aggregation hierarchy diagram(s) (if any), interface specifications, and sequence diagrams here. | | | |
|  |  |  | | | |
| **4.** | **DATA DESIGN** | | | | |
|  | **4.1** | **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. | | | |
|  |  | **4.1.1** | | **Logical Data Model** | |
|  |  |  | | A logical data model provides all the information about the various entities and the relationships between the entities present in a database (ER diagram or object-oriented classes). | |
|  |  | **4.1.2** | | **Physical Data Model** | |
|  |  |  | | A physical data model is a database-specific model that represents relational data objects (for example, tables, columns, primary and foreign keys) and their relationships. | |
|  |  |  | | | |
|  | **4.2** | **Data Flow** | | | |
|  |  | A Data Flow is visual representation of the information flows within a system. It shows how information is [input](https://www.computerhope.com/jargon/i/input.htm) to and [output](https://www.computerhope.com/jargon/o/output.htm) from the system, the sources and destinations of that information, and where that information is [stored](https://www.computerhope.com/jargon/s/stordevi.htm). | | | |
|  | **4.3** | **Data Dictionary** | | | |
|  |  | Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.1, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters. | | | |
|  |  |  | | | |
| **5.** | **COMPONENT DESIGN** | | | | |
|  | Component-level design is elaborative in nature. It transforms information from requirements into a design representation that provides sufficient detail to guide the construction (coding and testing) activity. | | | | |
|  | 5.1 | | Using the requirements, each analysis class and architectural component is elaborated | | |
|  |  | |  | | |
|  | 5.2 | | Classes and components in this category include. GUI components (often available as reusable components), Operating system components,  Object and data management components | | |
|  |  | | | | |
| **6.** | **HUMAN INTERFACE DESIGN** | | | | |
|  | **6.1** | **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. | | | |
|  | **6.2** | **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. | | | |
|  | **6.3** | **Screen Objects and Actions** | | | |
|  |  | A discussion of screen objects and actions associated with those objects. | | | |
|  |  |  | | | |
| **7.** | **USER REQUIREMENT VS COMPONENT TRACEABILITY MATRIX** | | | | |
|  | Give a table that cross references architectural components (based on defined component identifiers) to user requirements numbered in the SRS. It may be appropriate to omit references to non-functional user requirements that are not code related (e.g. legal requirements).   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | UR1 | UR2 | UR3 | UR4 | UR5 | … | URn | | AR1 | X | X |  |  |  |  |  | | AR2 |  |  | X |  |  |  |  | | AR3 |  |  | X |  |  |  |  | | AR4 |  | X |  |  |  |  | X | | AR5 | X |  |  |  |  |  | X | | AR6 |  |  |  | X |  |  |  | | AR7 |  |  |  |  | X |  |  | | | | | |