

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.

Corresponding to V1 report

Corresponding to V2 report

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 have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to [IEEE Std 1016-1998](#)¹ for the full IEEE Recommended Practice for Software Design Descriptions.

The link seems broken. The attached pdf is a newer version: IEEE Std 1016-2009.

¹ <http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf>

(Team Name)

(Project Title)

Software Design Document

Name (s):

Lab Section:

Workstation:

Date: (mm/dd/yyyy)

TABLE OF CONTENTS

1.	INTRODUCTION	2
1.1	Purpose	2
1.2	Scope	2
1.3	Overview	2
1.4	Reference Material	2
1.5	Definitions and Acronyms	2
2.	SYSTEM OVERVIEW	2
3.	SYSTEM ARCHITECTURE	2
3.1	Architectural Design	2
3.2	Decomposition Description	3
3.3	Design Rationale	3
4.	DATA DESIGN	3
4.1	Data Description	3
4.2	Data Dictionary	3
5.	COMPONENT DESIGN	3
6.	HUMAN INTERFACE DESIGN	4
6.1	Overview of User Interface	4
6.2	Screen Images	4
6.3	Screen Objects and Actions	4
7.	REQUIREMENTS MATRIX	4
8.	APPENDICES	4

1. INTRODUCTION

1.1 Purpose Required

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 Required

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 Optional

Provide an overview of this document and its organization.

1.4 Reference Material Optional

This section is optional.

List any documents, if any, which were used as sources of information for the test plan.

1.5 Definitions and Acronyms Optional

This section is optional.

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 Required

Give a general description of the functionality, context and design of your project. Provide any background information if necessary.

3. SYSTEM ARCHITECTURE

3.1 Architectural Design Required

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
如可行，请将整个系统划分成功能模块，介绍各个模块的功能以及相互关系。

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 Decomposition Description Optional

Provide a decomposition of the subsystems in the architectural design. 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. 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.

Search and learn the different diagrams by Google if needed.
For OO description, you can use UML, you can also try
Rational Rose software if you like (not required).

3.3 Design Rationale Optional

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn't choose them.

4. DATA DESIGN

4.1 Data Description Required

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.

4.2 Data Dictionary Optional

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, 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 Optional

In this section, we take a closer look at what each component does in a more systematic way. If

you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

6. HUMAN INTERFACE DESIGN

6.1 Overview of User Interface Required

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 Required

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.)

6.3 Screen Objects and Actions Optional

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

7. Describe design patterns and/or polymorphism features used in the project.

即使不是典型的设计模式，如果采用了多态，也可以列出。

~~7. REQUIREMENTS MATRIX~~ Remove this part

Provide a cross-reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

8. Describe other programming skills you learned in the project.

~~8. APPENDICES~~ Optional

This section is optional.

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.