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 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 1016­1998](http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf)1 for the full IEEE Recommended Practice for Software Design Descriptions.

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](#_bookmark0)
   1. [Purpose 2](#_bookmark1)
   2. [Scope 2](#_bookmark2)
   3. [Overview 2](#_bookmark3)
   4. [Reference Material 2](#_bookmark4)
   5. [Definitions and Acronyms 2](#_bookmark5)
2. [SYSTEM OVERVIEW 2](#_bookmark6)
3. [SYSTEM ARCHITECTURE 2](#_bookmark7)
   1. [Architectural Design 2](#_bookmark8)
   2. [Decomposition Description 3](#_bookmark9)
   3. [Design Rationale 3](#_bookmark10)
4. [DATA DESIGN 3](#_bookmark11)
   1. [Data Description 3](#_bookmark12)
   2. [Data Dictionary 3](#_bookmark13)
5. [COMPONENT DESIGN 3](#_bookmark14)
6. [HUMAN INTERFACE DESIGN 4](#_bookmark15)
   1. [Overview of User Interface 4](#_bookmark16)
   2. [Screen Images 4](#_bookmark17)
   3. [Screen Objects and Actions 4](#_bookmark18)
7. [REQUIREMENTS MATRIX 4](#_bookmark19)
8. [APPENDICES 4](#_bookmark20)

### INTRODUCTION

## Purpose

The purpose of this document is to give general look of the project.

## Scope

The main goal is to help restaurants, companies, nutritionists, individuals and more to achieve better schedule in their meal plans. Our project will have comprehensive functionality which will be beneficial for the users because they will be able to gain all the information they need for a meal plan (such as the cost, the ingredients, the nutrition values, instructions of how to store specific ingredients, and more (is detailed in the document " Software Requirements Specification" under project-scope)).

## Overview

Provide an overview of this document and its organization.

### SYSTEM OVERVIEW

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

### SYSTEM ARCHITECTURE

## Architectural Design – database, the application, one will store, gui, calculate cost, שאיבת מידע על מתכונים במערכת need to separate in order to realize how it will all work together

## תקינות קלט- זיהוי תמונה

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.

In the PowerPoint " Tables in database " we specified the tables we need in order to store all the data for the application.

## Decomposition Description

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 object­oriented description. For a functional description, put top­level 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.

## Design Rationale

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.

### DATA DESIGN

## 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.  
  
For this project we will used database of MySQL and php to communicate with it.  
This kind of database stores its information in tables. Each component will have its own table, linking it to small tables to store information of a kind (the linking will help us to extract information more easily (by foreign key)).  
In order to use the information from the database, we will pull the information and create an object. It will help us to use it in the code, we will be able to perform other functions on this object according to the user request from the application.  
User table – will have user details to store id, email, and other necessary information  
(for example, by the user id, we can extract user self-ingredient/recipes and display it in the target place).   
Ingredient table – will store all the ingredients in the system  
Recipes table - will store all the recipes in the system

## Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

### COMPONENT DESIGN

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.

### HUMAN INTERFACE DESIGN

## 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.

## Screen Images

Listed in the PowerPoint "User interface".

## Screen Objects and Actions

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

### REQUIREMENTS MATRIX

Provide a cross­reference 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.

### APPENDICES

We recommend looking at the PowerPoints - User interface, Separation to classes, Use-case for previous knowledge.