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]](#footnote-1) for the full IEEE

Recommended Practice for Software Design Descriptions.

Sticks and Stones

**World Wide Banking**

Software Design Document

Neghea Serban Elian

Perisoara Mihai Cezar

Popa Mihai Alexandru

Date: (03/23/2020)

**TABLE OF CONTENTS**

[1. INTRODUCTION 2](#_Toc4987)

[1.1 Purpose 2](#_Toc4988)

[1.2 Scope 2](#_Toc4989)

[1.3 Overview 2](#_Toc4990)

[1.4 Reference Material 2](#_Toc4991)

[1.5 Definitions and Acronyms 2](#_Toc4992)

[2. SYSTEM OVERVIEW 2](#_Toc4993)

[3. SYSTEM ARCHITECTURE 2](#_Toc4994)

[3.1 Architectural Design 2](#_Toc4995)

[3.2 Decomposition Description 3](#_Toc4996)

[3.3 Design Rationale 3](#_Toc4997)

[4. DATA DESIGN 3](#_Toc4998)

[4.1 Data Description 3](#_Toc4999)

[4.2 Data Dictionary 3](#_Toc5000)

[5. COMPONENT DESIGN 3](#_Toc5001)

[6. HUMAN INTERFACE DESIGN 4](#_Toc5002)

[6.1 Overview of User Interface 4](#_Toc5003)

[6.2 Screen Images 4](#_Toc5004)

[6.3 Screen Objects and Actions 4](#_Toc5005)

[7. REQUIREMENTS MATRIX 4](#_Toc5006)

[8. APPENDICES 4](#_Toc5007)

# INTRODUCTION

## Purpose

This software design document describes the architecture, implementation and system design of an online banking system presented in the IEE Software Requirements Specifications document. This is an online banking system, designed for online payment and actions such as viewing balances, check transactions, transfer money and make payments.

## Scope

Our product, entitled “World Wide Banking”, is an online payment system that allows our customers to conduct financial transactions. The customers can check at any time their account balances and view data about their account, can check their last transactions, can transfer money at any time from one account to another and they are also able to make payments to various services (electricity, telephone, internet and so on).

All of the information from above is stored in a database, all of the customers’ data about their accounts plus other data that is not available to a customer, like last successful login and the date when they activated their online banking service. The scope is to have a secured place where a person can keep their money and do different actions with it, but also aims to provide information about keeping and using the money for the clients, using the help and contact sections. The application has a FAQ section where the most asked questions by the users are asked.

The banking application aspires to keep growing, having more users that use it, increasing the popularity and profits. The interface needs to be maintained to a simple aspect so every type of user is able to use it.

## Overview

This document a written description of a software product, which is written in order to give a overall guidance to the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. Practically, it is required to coordinate a large team under a single vision, needs to be a stable reference, and outline all parts of the software and how they will work.

## Reference Material

[About SDD](https://en.wikipedia.org/wiki/Software_design_description)

[Wiki Online Banking](https://en.wikipedia.org/wiki/Online_banking)

## Definitions and Acronyms

SSD = Software Design Document

FAQ = Frequently Asked Questions

# SYSTEM OVERVIEW

The online banking application utilizes a database, which stores everything from transactions, to user data and also data that is not available to the clients.

The online banking application is designed with the help of ASP.NET using the C# language, implementing HTML and CSS elements and also the databases from before, the interface is kept to a minimum to be easily used by every type of client.

The banking application has the following features:

* A fast and reliable feedback form which allows the client to ask any question, regarding the banking application, which will go to the contact team.
* A minimal aspect which is made for every type of user of the application which provides a simpler access and understanding.
* A banking account management which provides the user any type of action he needs for managing the money.

# SYSTEM ARCHITECTURE

## 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.

## 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.

## 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

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.)

## 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

*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.

1. http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee­SDD.pdf [↑](#footnote-ref-1)