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 and system design of an online banking system.

## 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.

## Overview

Provide an overview of this document and its organization.

## Reference Material

*This section is optional.*

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

## Definitions and Acronyms

*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.

# 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

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

The interface was designed to be easy to understand, easy to use, simple and modern. The interface consists in simple containers to display the informations, few buttons to choose in order not to confuse the client.

As the client starts using the applications, he is prompted to enter login credentials. Once logged in, a list view of accounts is displayed. The list view was chosen so it is clear even to the most unprepared user where everything is, making it easy to know where to look for something. Each account has it’s own buttons, clearly labeled, under the container which displays the vital informations of the account (amount of money and transactions history). It it’s the first time the client uses the application, instead of the overview of the account, there is an information panel explaining how to make an account.

The main feature of the simplicity of this application is the fact that any action is a two-step process at most, many of the important actions even being a one-step process.

## Screen Images

## Screen Objects and Actions

Upon entering the application, the user will be greeted by a page asking him to choose to log in or register. In case the user encounters any difficulties, there are two more buttons on the bottom right of the page, one for commonly asked questions (whose purpose is obvious) and one to contact the support team.

When clicking a button, the respective page will be opened. The login one will have only 2 fields, the username and password. If the credentials entered are not correct, the outline of the field will be outlined in red and the user will be notified. The register page is more complex, consisting in 13 fields with different sets of rules as follows: the username will only consist in alpha-numeric characters; the password will require at least one capital letter, a number and a non-alpha-numerical character; the CNP, phone and postal code fields will only allow numbers to be entered; the mail field will check the syntax of the string entered to match; the country, city and district will also be checked. If any of the ones mentioned above is not checked or if the passwords don’t match, their respective fields will be outlined red and the user will be notified. The username, phone number and CNP will be not less than 8, 10 and 13 characters long respectively. Theese two pages also have the “Contact Us” and “FAQ” buttons on the bottom right, same as the first one.

The main page will have a few elements, the exact number is up to the customer (will be explained below, of course). The main body consists of 4 buttons and 3 containers.

The containers always displays how much money is in that account. Also in this container there is a section where the user can see the exchange rates for EUR and USD. The biggest of the sections within this container is the transactions history which will display the last 10 transactions of this account with the amount that was spent, the name of the business the money was sent to and the exact time of the transaction.

The buttons serve the following roles: The “Send Money” button, obviously, will prompt the user to enter the information required in order to send. The next button is “Create Account”. Upon clicking this button, the user will be asked for a PIN and a currency. After creating the account the new account will form a new body on the main page, making it a list view. If there is no account created yet, there will only be the buttons. The next button is the “Delete Account” button, whose purpose is also obvious. The action must be confirmed by entering the PIN of the account.

The last button is the gear icon button which is only a drop down button for more account management options: change pin, card renewal, card lost, and request statement. The change PIN button will ask the user for the old and the new PINs and will modify them upon confirming the new PIN. The card renewal button will present the terms of use of the product. The only thing the user needs to do is to agree to the terms and confirm the renewal process. The card lost button will instantly lock any transaction associated with that card and will inform the user on how to get a replacement, ensuring him that the money on the lost card is safe. The statement request is as straight forward as it seems, the user will be notified about the request being received and the statement being finished.

# REQUIREMENTS MATRIX

|  |  |  |
| --- | --- | --- |
| ID | Functional Requirements | System Components Purpose |
| A0 | User log in | Checks login credentials in the DB |
| B0 | User log out | Unloads the RAM of the login credentials |
| C0 | Contact us | Forwards user message to the support team through the DB |
| D0 | Register | Checks syntax and uploads data to the DB |
| E0 | Create banking account | Creates new data in the DB with the login credentials as key |
| F0 | Delete banking account | Deletes the data created by the function with ID E0 |
| G0 | Exchange rates | Computes the difference between currencies and stores it |
| H0 | Transfer money | Modifies the data of the recipient and sender in the DB |
| I0 | Receive money | Modifies the data of the sender and recipient in the DB |
| J0 | Change PIN | Modifies account information in the DB |
| K0 | Card renewal | Requests new space allocation in the DB |
| L0 | Declaring card lost | Requests blocking of banking account credentials usage |
| M0 | Taking statement | Requests pull of large data file from the DB |
| N0 | Help | Displays questions containing most used key words from DB |

# 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)