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 [I EE E St d 1016­ 1998](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.** | 1.1 | **INTRODUCTION**  Purpose | **2**  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.** | 4.1 | **DATA DESIGN**  Data Description | **3**  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**

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**

*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 !!**

**XAMPP** - a free and open-source cross-platform web server solution stack package developed by Apache Friends consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

**PHP** - a popular general-purpose scripting language that is especially suited to web development.

**MySQL** - an open-source relational database management system.

**HTML**- Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser.

**CSS**- Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

**2. SYSTEM OVERVIEW**

Movieteca is a website with a futuristic design, ready for everyday user to rent his favorite movies. This website will use the latest MsSQL Database in order to offer to our clients the best and the fastest experience possible on a website.

The website is designed to be compatible with services and infrastructure existing in the filming industry. We are negotiate with movie producers, filming studios, in order to have all the policies and security protocols in order.

The Movieteca will provide the following capabilities:

* User account management, which shows a detailed view of the client’s choices, rents status, rents return time and other options.
* Integration of all maintenance data which will allows the rental to be in real-time.
* Enhanced and additional user interface for a smooth navigation and easy to use method
* Feedback form which allows the user to leave a comment or a review to a movie, and also to tell how prompt were we in our provided services.

**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 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.

**3.3 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.

**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.

**4.2 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.

**5. 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.

**6. HUMAN INTERFACE DESIGN**

**6.1 Overview of User Interface**

Based on the user reviews, upon accessing our products, you need to login, the login page is simple, all you have to do is to enter your **Username** and **Password**, and select what you are, an user or an admin. If the user does not have an account, he will be able to register. Unsuccessful registration/logins, will lead to a warning. Failing to complete the recommended fields, will lead to a warning. After the login process is done, the user will be able to browse: “Home”, “Movies”, “Contact” pages. Also for a quick search we have added search bar. After our user have decided what movie to pick, he can add it to his cart then proceed to checkout, also he can leave a to the movie that he have seen. If the user get something unclear, he can message us through live-chat functionality.

User interface share following qualities or characteristics:

**Clarity**: The interface avoids ambiguity by making everything clear through language, flow, hierarchy and metaphors for visual elements.

**Responsiveness**: This means a couple of things. First, responsiveness means speed: a good interface should not feel sluggish. Secondly, the interface should provide good feedback to the user about what’s happening and whether the user’s input is being successfully processed.

**Aesthetics**: While you don’t need to make an interface attractive for it to do its job, making something look good will make the time your users spend using your application more enjoyable; and happier users can only be a good thing.

**Efficiency**: Time is money, and a great interface should make the user more productive through shortcuts and good design.

.

**6.2 Screen Images**

**6.3 Screen Objects and Actions**

**7. REQUIREMENTS MATRIX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **REQUIREMENTS**  **TRACEABILITY MATRIX** | | | | | | |
| Project Name: Car Rental Website | | | | | | |
| User class ID# | User Class Name | Functional Requirement ID# | Functional Requirement Name/Use Case | System Component | Priority | Test Case ID# |
| UC\_1 | Customer User | 1.1 | Website access | Link | High | #TC001 |
|  |  | 1.2 | Update notification | Newsletter | Medium | #TC002 |
|  |  | 1.3 | User Registration | Login/Register | High | #TC003 |
|  |  | 1.4 | User Login | Login/Register | High | #TC004 |
|  |  | 1.5 | Password retrieval | Login/Register | High | #TC005 |
|  |  | 1.6 | Rental search | Search module | High | #TC006 |
|  |  | 1.7 | Rental selection & checkout | Checkout module | High | #TC007 |
|  |  | 1.8 | Profile update | Profile module | High | #TC008 |
|  |  | 1.9 | Check Rented Movies | Profile module | High | #TC009 |
|  |  | 1.10 | Post feedback | Profile module | High | #TC010 |
|  |  | 1.11 | Sign out and Log back in | Login/Register | High | #TC011 |
| UC\_2 | Administrator User | 2.1 | Admin Login | Super Login module | High | #TC012 |
|  |  | 2.2 | Create/Delete movies listing | Admin Panel  module | High | #TC013 |
|  |  | 2.3 | Post movies listing | Admin Panel  module | High | #TC014 |
|  |  | 2.4 | Confirm/Cancel customer checkout | Admin Panel  module | High | #TC015 |
|  |  | 2.5 | Manage feedback | Admin Panel  module | High | #TC016 |
|  |  | 2.6 | Contact us live-chat management | Admin Panel  module | High | #TC017 |
|  |  | 2.7 | Check User Details | Admin Panel  module | High | #TC018 |
|  |  | 2.8 | Update public details | Admin Panel  module | High | #TC019 |
|  |  | 2.9 | Manage newsletter subscribers | Admin Panel  module | High | #TC020 |
|  |  | 2.10 | Dashboard menu statistics | Admin Panel  module | High | #TC021 |
|  |  | 2.11 | Change admin password | Admin Panel  module | High | #TC022 |
|  |  | 2.12 | Log out of Admin Panel | Admin Panel  module | High | #TC023 |
| UC\_3 | Guest User | 3.1 | Explore website | Link | High | #TC024 |
|  |  | 3.2 | Needs to create an account | Login/Register  prompt | High | #TC025 |

**8. 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.