**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 has been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report.

(**You do not need to include this page on your document**)

Due on: 9/04/2019 (Tuesday, Lab Time)

For: Both Group A and B

**(Project Title)**

**Media Player**

**Software Design Document**

Name (s):

1. Pankaj Chandra Kar; Roll: 49
2. Sanjoy Kumar Mahato; Roll: 55

**TABLE OF CONTENTS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1.** | **INTRODUCTION** | | | |  |
|  | 1.1 | *Purpose* | | |  |
|  | 1.2 | *Scope* | | |  |
|  | 1.3 | *Overview* | | |  |
|  | 1.4 | *Reference Material* | | |  |
|  | 1.5 | *Definitions and Acronyms* | | |  |
|  |  |  | | |  |
| **2.** | **SYSTEM OVERVIEW** | | | |  |
|  |  |  | | |  |
| **3.** | **SYSTEM ARCHITECTURE** | | | |  |
|  | 3.1 | *Architectural Design* | | |  |
|  | 3.2 | *Design Rationale* | | |  |
|  | 3.3 | *Architectural Goals and Constraint* | | |  |
|  |  | *3.3.1* | | *Security* |  |
|  |  | *3.3.2* | | *Persistence* |  |
|  |  | *3.3.3* | | *Performance* |  |
|  |  |  | | |  |
| **4.** | **DATA DESIGN** | | | |  |
|  | 4.1 | *Data Description* | | |  |
|  |  | *4.1.1* | *Logical Data Model* | |  |
|  | 4.2 | *Data Flow* | | |  |
|  |  |  | | |  |
| **5.** | **COMPONENT DESIGN** | | | |  |
|  | 5.1 | *Analysis Classes* | | |  |
|  |  | *5.1.1* | *Class diagram* | |  |
|  |  | *5.1.2* | *Activity Diagram* | |  |
|  |  |  |  | |  |
| **6.** | **HUMAN INTERFACE DESIGN** | | | |  |
|  | 6.1 | *Screen Images* | | |  |
|  | 6.2 | *Screen Objects and Actions* | | |  |
|  |  |  | | |  |
| **7.** | **USER REQUIREMENT VS COMPONENT TRACEABILITY MATRIX** | | | |  |
|  |  |  | | |  |
|  |  |  | | |  |

**Details**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1.** | **INTRODUCTION** | | | | |
|  | **1.1** | ***Purpose*** | | | |
|  |  | Our goal is to create a user friendly light weight media player. Which can select any media file from mobile and play it. It will support all the media files. Besides a will be media list option by which user can save their favourite song lists. | | | |
|  | **1.2** | ***Scope*** | | | |
|  |  | This project is an android based project. So only android user can use this product. User can listen songs and watch videos in their free time with this app. There are many attractive features so that user can feel comfortable to use this app. Features, goals, cost of the project are described below. | | | |
|  | **1.3** | ***Overview*** | | | |
|  |  | This project is an android based project. So only android user can use this product. User can listen songs and watch videos in their free time with this app. There are many attractive features so that user can feel comfortable to use this app. Features, goals, cost of the project are described below. | | | |
|  | **1.4** | ***Reference Material*** | | | |
|  |  | * Functional requirements are written by Sanjoy Kumar Mahato and * Non-functional requirements are written by Pankaj Chandra kar. | | | |
|  | **1.5** | ***Definitions and Acronyms*** | | | |
|  |  | Java programming language and use FXML for designing the layout of the app. | | | |
|  |  |  | | | |
| **2.** | **SYSTEM OVERVIEW** | | | | |
|  |  | Our project is android based. So this media player will be available for most of the people in the world. There will be basic functions of a media player. We will design the user interface using FXML and there will be java(android) code for backend. | | | |
|  |  |  | | | |
| **3.** | **SYSTEM ARCHITECTURE** | | | | |
|  | **3.1** | **Architectural Design** | | | |
|  |  | The main architecture will contain the list, playing media, volume and seeking and shuffle. Besides there are many other functions. But there are the critical one will need to manage. | | | |
|  |  | 1. **Media List:** This subsystem will contain the list of the media files in a specific folder. There will be another list which is known as ‘Favorites’.   Besides there will be a option for selecting media files from the memory.   1. **Playing Media:** This subsystem is responsible for playing selected media files. There can be an option like ‘Loop’ which will play the same media file. 2. **Volume and Seek Control:** This subsystem will be responsible for increasing or decreasing volume. Besides there will be a seek option. Which will change the media files playing time. 3. **Shuffle:** This subsystem will help to shuffle the media list. It is random.   \*\*There is no specific direction. Because user can go to any options from any menu. | | | |
|  | **3.2** | **Design Rationale** | | | |
|  |  | We didn’t choose the other materials like home, search, add music etc. Because our main purpose is to play a media file. For we need playing the media option. Besides Shuffle is an important issue here. It will work in the text file. | | | |
|  | **3.3** | **Architectural Goals and Constraint** | | | |
|  |  |  | | | There are some key requirements and system constraints that have a significant bearing on the architecture. |
|  |  | **3.3.1** | | | **Security** |
|  |  |  | | | As a media player there is not enough security options. But there will be pop up window which will ask user to give access to the Phone Memory. |
|  |  | **3.3.2** | | | **Reliability/Availability** |
|  |  |  | | | The system is reliable. There is possibility of meeting the specifications is 95%.  Moreover, 90% of the time it will take less than 10 seconds to restart the software. |
|  |  | **3.3.3** | | | **Performance** |
|  |  |  | | | As we will mainly focus on playing media we would make it light weight. So it will be faster than other media player. |
|  |  |  | | | |
|  |  |  | | | |
| **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. | | | |
|  |  | **4.1.1** | | **Logical Data Model** | |
|  |  |  | | *A logical data model provides all the information about the various entities and the relationships between the entities present in a database (ER diagram or object-oriented classes).*  As there is no need of any extra Database Management System for Creating Media Player, but there will be recent playlist which we can save as text files. | |
|  |  |  | | | |
|  | **4.2** | **Data Flow** | | | |
|  |  | As our main database it text file there is no complex flow. But here is diagram which will help to understand how we will save media locations and how the list is created. How can a new song be added in the list. | | | |
|  |  |  | | | |
| **5.** | **COMPONENT DESIGN** | | | | |
|  | *Component-level design is elaborative in nature. It transforms information from requirements into a design representation that provides sufficient detail to guide the construction (coding and testing) activity.* | | | | |
|  |  | | Here is our main activities and what will be inside the components. From the diagram we can see all the functions what will be added in the media player. It will be easy for coding according to the diagram below. | | |
|  | 5.1 | |  | | |
|  |  | | | | |
| **6.** | **HUMAN INTERFACE DESIGN** | | | | |
|  | **6.1** | **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.*  The screens will look similar like there images. It will not be 100%. But It is the basics what we need to do. | | | |
|  | **6.3** | **Screen Objects and Actions** | | | |
|  |  | *A discussion of screen objects and actions associated with those objects.*   1. Firstly, there is a loading screen. 2. In the home screen there is a option for search. 3. There are play, pause, open, exit button in the home screen. 4. There is a list in the home screen. 5. There is a media list screen. 6. There is the main play screen. 7. In the play screen there are play, pause, next, previous button. 8. There is option for shuffle. | | | |
|  |  |  | | | |
| **7.** | **USER REQUIREMENT VS COMPONENT TRACEABILITY MATRIX** | | | | |
|  | Give a table that cross references architectural components (based on defined component identifiers) to user requirements numbered in the SRS. It may be appropriate to omit references to non-functional user requirements that are not code related (e.g. legal requirements).   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  | UR1 | UR2 | UR3 | UR4 | UR5 | … | URn | | AR1 | X | X |  |  |  |  |  | | AR2 |  |  | X |  |  |  |  | | AR3 |  |  | X |  |  |  |  | | AR4 |  | X |  |  |  |  | X | | AR5 | X |  |  |  |  |  | X | | AR6 |  |  |  | X |  |  |  | | AR7 |  |  |  |  | X |  |  | | | | | |