**Software Requirements Specification (SRS)**

**for Alarm and Music Player Project**

*Baseline version 0.1*

*Issued on : Juli 27, 2015*

Issued by :

Yuwono

Rosabella

Yordan

Albert Rahardjo

Issued for :

Irene Lazaruli   
(Human-Computer Interaction Project)

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 0.1 | Juli 12, 2014 | Yuwono, Rosabella, Yordan | initial version |
|  |  |  |  |

**Document Approval**

The following Software Requirements Specification has been accepted and approved by the following :

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date** | **Signature** |
| Yuwono | Project Leader | August |  |
| Rosabella, Yordan | Project Design & Planner | August |  |
|  |  |  |  |

**Table of Contents**

Table of Contents page.

Change History ii

Document Approval ii

List of Figures iv

List of Tables v

**Chapter 1 1-19**

**INTRODUCTION**

1.1 Purpose 1

1.2 Scope 1

1.3 Definitons, Acronyms, and Abbrevations 1

1.4 References 1

1.5 Overview 1

**Chapter 2**

**GENERAL DESCRIPTION**

2.1 Product Perspective 3

2.2 Producr Functions 3

2.3 User Characteristics 3

2.4 General Constraints 3

2.5 Assumption and Dependencies 3

**Chapter 3**

**SPECIFIC REQUIREMENTS**

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Use Cases 6

3.4 Class/Objects 7

3.5 Non-Functional Requirements 7

3.6 Design Constraints 7

3.7 Other Requirements 7

**Chapter 4**

**CHANGE MANAGEMENT PROCESS** .......................................................... 8

**List of Figures**

Product Perspective 3

MP3 Player 5

Clock 6

Alarm 6

Stopwatch 7

Use Case 8

**Chapter 1**

**INRODUCTION**

* 1. **Purpose**

The purpose of this project is to see an interaction between human and the system. The system is then equipped with clock and music player to trigger some basic interaction between human and system, alarm program can be combine with Music Player with Netbeans program

* 1. **Scope**

They scope of this Alarm and Music Player project is used to make the works of an alarm and music player in general that can be compiled as an interaction of human and system. The goal is how the alarm can be used and combined with the music player yet they still have each of their responsibility with their own tasks.

* 1. **Definitions, Acronyms, and Abbreviations**

*UPH : Universitas Pelita Harapan.*

*IT : Information Technology.*

*GUI : Graphic User Interface.*

*SDS : Software Design Specification.*

*SRS : Software Requirement Documentation.*

*SPMP : Software Project Management Plan.*

*STR : Software Testing Report.*

* 1. **References**

*Farre, Joyce. 2013. "Java Programming". Edisi ke-7. Boston: Cengage Learning.*

*Malik, D.S. 2011. "C++ Programming : From Problem Analysis To Program Design". Edisi ke-5. Boston: Cengage Learning.*

* 1. **Overview**

This system is designed for :

• Help people to listen to music with added feature such as clock, alarm, and stopwatch.

• Music player can play any song in the user’s database (preferably MP3 format).

• Alarm and stopwatch is able to be set, reset, and snoozed according to the user’s wishes.

Low-level design (LLD) architecture of our system is the result of multiple sub-systems. Class diagram, with all the methods available is an example of LLD. Our LLD system described every function from the algorithm of our system, meanwhile our system high-level design (HLD) is a decomposition to some models or function, and represent into an interface that we made and available to use for user. To model our system, we use major design artifatcs such as use case diagram, class diagram, sequences diagram, and state-transition diagram.

**Chapter 2**

**GENERAL DESCRIPTION**

1. **Product Perspective**

This product is a totally independent and self-contained purpose. There is no need significant hardware. We just use laptops with NetBeans (GUI) software installed for auditing, designing, and LCD laptop for showing the result.

Final State

( Survey)

NetBeans

(GUI)

Project Planning

1. **Product Functions**

This system is designed for :

• Help people to listen to music with added feature such as clock, alarm, and stopwatch.

• Music player can play any song in the user’s database (preferably MP3 format).

• Alarm and stopwatch is able to be set, reset, and snoozed according to the user’s wishes.

1. **User Characteristics**

Users give significant rules of the working in the project. In our survey, some of the instructions need user intention. The understanding of user doing every steps in instruction are needed to make the project easy to use. Some of manual instructions by using keyboard are already given in the program, so user will able use all features without ease the confusion.

1. **General Constraints**

Here is the list of all constraints that are made :

* Budget to build this project is Rp 50.000,00 to spread the quesioner for making a survey
* Time needed to complete this project is about 3 months.

1. **Assumptions and Dependencies**

Here is the list of all Assumptions and Dependencies that are made :

* GUI is highly scored based on the how the user interface
* Time can be used as alarm, stopwatch, timer
* Music player can play from browsing a file
* Music player have 5 list of songs
* Music can be paused and play
* All the buttons can be use properly (Next, Previous, Play, Pause, Stop) for Music.
* All the buttons can be use properly (Set) for Alarm & (Set, Reset) for Stopwatch.

**Chapter 3**

**SPECIFIC REQUIREMENTS**

1. **External Interface Requirements**

User Interface

* User must be able to see and have an basic knowledge about using the music player, alarm & stopwatch in general software
* User can do on click with mouse or manually with using keyboard press
* The instruction of using keyboard manually attach in the help’s box, that user can read before use them or if user get any confuse, they can see it in a couple time.
* User must be click or press on the function keys to make the program run their features.
* This program has some features that are same as in general of using music player, alarm & stopwatch. There are some buttons features : start, stop, play, pause, snooze, set and reset.

Find Tracks Button

Music’s Tittle

Close Button



Minimize Button

Change to Alarm & Stopwatch directories

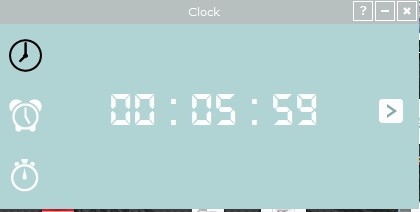
Help Button

Next Track Button

Previous Track Button

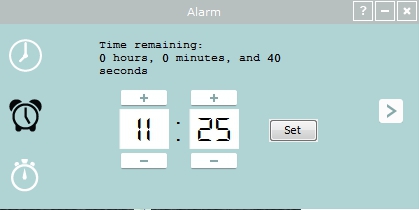
Play Button

Stop Button



Clock Directiories

Set Alarm button



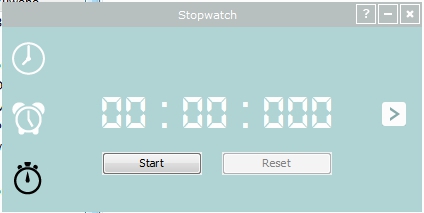
Alarm Directories



Dismiss Alarm button

Reset Stopwatch

Start Button



Stopwatch Directories



Lap (Record) Button

Hardware Interface

* The software can give the interaction to hardware through the initialization of the code program, from when and what source code that can make the user able to do their part (by click or press the function keys/button)

Software Interface

* This program can be run in Windows, MAC OS
* NetBeans must be installed and good tested to make the program running well

1. **Functional Requirements**

* A user should be able to play song from their own computer directory
* A user should be able to skip, return, play and pause their song
* A user should be able to set timer for their needs
* A user should be get the exact clock a user should be able to use stopwatch
* A user should be able to switch from clock to music player
* The system should be able to get the clock data from the computer
* The system should be able to resume the song when the user pause the song

<<include>>

Stop Music

1. **Use Cases**

<<include>>

Play Music

Next Track

Browse Music File

<<include>>

Previous Track

Change Directories

<<include>>

Stop Alarm

Set Alarm

<<include>>

Dismiss Alarm

<<include>>

Start Stopwatch

Reset Stopwatch

<<include>>

Record Stopwatch

1. **Classes/Objects**
2. MainGui

This class is build to make the function of alarm & stopwatch. All the detail of JPanel, JFrame, JButton are already explain well in the coding of this class.

1. MainClass

This class is build to make the function of music. How to play, stop, next & previous, browse track are already given in the coding and explain well.

1. **Non-Functional Requirements**

* This project only can be executed by OS with NetBeans software installed
* Coding only can be compile in NetBeans software or by clicking twice the .jar package
* Project can be revised by revised the coding (if needed) by user if they can understand the work of the coding.
* If NetBeans is not successfully installed, the project cannot be running perfectly.

1. **Design Constraints**

Standard for design :

* We build the software to make user can click the button for starting the actions.

The actions are :

Play button -> user able to play music in media player

Stop button -> user able to stop music in media player

Previous / Next button -> user able to next and go back the media player

Set button -> user able to set the alarm & stopwatch

Dismiss button -> user able to dismiss the alarm when ring

Reset & Lap button -> user able to reset the alarm/ stopwatch

Requirements for space capacity or performance, this project only can be execute by OS that installed by NetBeans program.

1. **Other Requirements**

Security of this software doesn’t a main point in our project, because the software is like general software and there’s no special requirement to use this software. This software can be used for all people and all age, when they need to use this software they can use it. For safety, this software is already give the help box in right upper, if people touch and click in any other button that doesn’t have any function, this software will not respond well. And be careful we must to wait a moment and don’t really quick to change the directories between each other, the software doesn’t respond as quick as you wish.

**Chapter 4**

**CHANGE MANAGEMENT PROCESS**

To update the SRS, as needed, we must to know what the main and the goal when you change the program. We must to see clearly and also know what’s the main idea we change the code of our program, so we can change the code exactly in the main point of the code. The new code can be save as the user wish when they already compile them and there is no error. The code will have no error if and only if they have good combination and collaboration between classes. It is technically can be success by user experience when try this software, so they will know what the things that must change, and where code must be added to implement the changes of the program. When there’s no error pop up as warning, we can conclude that the changes already approved.