**Music Player Using PYTHON**

Divyang Yadav (259743)

**PURPOSE 3**

SCOPE 3

PROBLEM STATEMENT 3

OVERVIEW 4

**OVERALL DESCRIPTION 4**

PROJECT PERSPECTIVE 4

PROJECT FUNCTIONALITIES/REQUIREMENTS 4

ASSUMPTIONS AND DEPENDENCIES 4

**SPECIFIC REQUIREMENT 5**

FUNCTIONAL REQUIREMENTS 5

USER REQUIREMENTS 5

DESIGN AND IMPLEMENTATION AND CONSTRAINTS 5

NON-FUNCTIONAL REQUIREMENTS 5

**MODULE OUTPUT 6**

**PURPOSE**

The main purpose of this documentation is to give a detailed view of project called Music Player using Python. Along with it we are provided requirements both functional and non-functional and also specification of this project.

1. **SCOPE** **:**

Scope of this project Music Player Using Python is to provide a platform where the user can:

1. Add Music From List
2. Play Music From List
3. Pause Running Music
4. Start Paused Music / Start New Music Form List
5. Stop Running Music
6. Increase / Decrease Volume of Music
7. **PROBLEM STATEMENT :**

The system aims to providing the functionality to an individual to Open or Select Song from the Computer or Personal Laptop and add it to the list and then play any song form this list anytime. A user also use it flexibly like whenever they need to pause music they use pause button and when they need to start it again they use start button to start music from where they pause and if they want to stop music then use stop button and play next song from added music list. The Users/Individuals also increase/decrease the volume of song as their requirement using volume bar provided in window. They also delete unwanted music songs from list which they not need in future.

1. **OVERVIEW :**

The system aims to provide the functionality to individual of select any music file and user will be able to play the song they can Start, Pause, Stop and Forward the song and can Increase or Decrease the volume as per the requirement. Duration of the song time will be also displayed and current Duration will be showed in window. The user also add songs form menu bar where add music from open menu which is placed inside File menu and they also Exit from application File menu Exit button. They also known about application and creator of that from about us button which is under Help Menu.

# **OVERALL DESCRIPTION**

# **PROJECT PERSPECTIVE :**

# The design of the software in this document follows the concept of providing the instant response in terms of playing the user selected song. Also the user start, stop, pause, forward and can also increase or decrease the volume and the duration so time will also been displayed.

# **PROJECT FUNCTIONALITIES / REQUIREMENTS :**

1. Select Song
2. Start, Pause, Stop, Forward.
3. Display the Total Duration and Current Duration of Song
4. Volume Increase or Decrease
5. Add or Delete song
6. **ASSUMPTIONS AND DEPENDENCIES :**

The proposed system has an python dependencies install on the client side. The product will be supported only by the below mentioned Operating System and Tools need to be installed.

**Supported Operating System :**

* Windows
* Linux
* Mac

**Tools need to be Installed on Client Machine:**

* Python 2.7 and Higher version
* Tkinter Dependencies
* Mutagen Dependencies
* Pygame Dependencies

**SPECIFIC REQUIREMENT**

1. **FUNCTIONAL REQUIREMENTS**

**USER REQUIREMENTS :**

* Select Song
* Start, Stop, Pause, Forward
* Increase or Decrease volume
* Display Current and Total Duration
* Add or Delete song

**DESIGN AND IMPLEMENTATION AND CONSTRAINTS**

**Hardware Requirements:**

* Processor: Qualcomm Or Higher Version
* RAM: 1.5 GB or Higher
* Disk Space: 100MB (Minimum)

**Software & Dependencies Requirements:**

* Operating System: Windows, Linux, Mac.
* Python 2.7 and Higher version
* Tkinter Dependencies
* Mutagen Dependencies
* Pygame Dependencies

**Constraints:**

* The Local database of music libraries needs to be having on Hosting Computer.

1. **NON-FUNCTIONAL REQUIREMENTS**

**Authorization:**

* Only admin has authority to make changes in Media player. Users can only select or play song.

**Availability:**

* 24 X 7 Availability will ensured by using an always Desktop app Service.

**Maintainability:**

* Modular System approach to be used to update or replace system components quickly without incurring significant downtime losses

**MODULE OUTPUT**







