A
PROJECT REPORT
ON

Music Player

Submitted in fulfilment for practical of
Software Engineering

Computer Science

Submitted By: -

Pulkit Goel (16EBKCS072)

Ujjwal Singh (16EBKCS109)

Submitted To:-

Mr. Rakesh Saini

DEPARTMENT OF COMPUTER SCIENCE
B.K. BIRLA INSTITUTE OF ENGINEERING & TECHNOLOGY PILANI (RAJ)
(Affiliated to Rajasthan Technical University, Kota)
2017-18

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we give to our respected Asst. Prof. **RAKESH SAINI** whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project especially in writing this report.

Furthermore I would also like to acknowledge with much appreciation the crucial role of the Director of our institution **Dr. P.S Bhatanagar** for permitting us to utilize all the necessary facilities of the institution.

We are also thankful to all the other faculty and staff members of our branch for their kind co-operation and help.

Pulkit Goel
Ujjwal Singh

INDEX

WORK DETAILS

2.1 INTRODUCTION

Here, we have a software as “Music Player”. This music player will be used to play songs in audio formats like .mp3, .wmv .wav.

A music player is a computer program for playing songs or music. Music player commonly displays standard media control icons such as play, pause, next, previous, fast forward, rewind, shuffle, replay, etc.

All the mainstream operating systems have at least one built-in media player. For example, Windows 10 comes with Groove Music Player and Windows Media Player while Apple OSX comes with Quick Time Player. Linux Distributions may also come with music player such as SM player, Amarok, Audacious, Banshee, Mplayer, Rhythmbon, Totem, VLC media player and Xine. Android OS comes Google Play Music as default.

All the music players have their own distinct features but with some common features also so, here we are developing a music player using all the common features in a different user interface. So all the requirements in the user interface related to our software is discussed below.

2.2 PROPOSED MUSIC PLAYER

In our proposed music player we have the provision for adding the songs with multiple song format support which are as follows:

- .Mp3
- .Wav
- .Wmv

Here, we can add multiple songs at a time to playlist. The objective of the music player is to allow the user to perform various functions such as:

- Play
- Pause
- Next
- Previous
- Shuffle
- Replay

This music player project mainly perform various actions related to a music file. Like when we click the play button the first song in the playlist will start playing. If the songs had been added in the playlist by the option of add songs present in the File menu.

Now when the song has been selected to play and it is been played then at the time of the playing of that song we can perform some operations on the songs in the playlist including the currently playing song.

The operations that can be performed are firstly pause i.e. it will temporarily stops the playing song. Secondly, we can press the next button to go to the next song in the playlist and vice-versa we can press the previous button for going to the previous song in the playlist. Also we have included an option for starting the song from beginning i.e. when the user will press the replay button. Now if the user wish to play any song in the playlist he/she can to go the song and simply select the song and the selected song will start playing. The last feature that had been added is the shuffle which is used to shuffle all the songs in the playlist.

2.3 PRESENT MUSIC PLAYER

Today's audio player can sync your songs with Android phones, some iOS devices, USB drives, and other portable music players. It can also keep an eye on your hard drive for changes and update the music library automatically. They comes with support for various audio formats, including MP3, AAC, WMA, WAV, M4A, FLAC, OGG, APE, TAK, etc.

We can customize Present Music Players with a number of skins and plugins. Their list of features includes support for 15-band equalizer, DSP effects, CD ripping, automatic metadata import, etc.

They have a services like SoundCloud, RockRadio, RadioTunes, etc. You can also configure Clementine to access your music stored on cloud drives, including Box, Dropbox, Google Drive, OneDrive, Amazon Cloud Drive, etc.

Their settings menu is quite well-decked too, allowing users to control variables like audio latency, background adjustment tools. They have rich sound mixing options and the excellent sound engine section, which features an impressive 18-band equalizer and numerous sound effects like Reverb, Flanger, Enhancer, etc.

They have numerous features like this also: -

- Auto-tag your music using the music databases of your choice.
- Keep your files automatically organized in folder based on tags.
- Customize a number of different keyboard shortcuts.
- Subscribe, download, and listen to podcasts.

3. SOFTWARE REQUIREMENT SPECIFICATIONS

Below is the brief details of the requirements of a software i.e. a music player.

1. Firstly play button is used to play the first song or the currently paused song.
2. Secondly pause button is a button used to pause the current song.
3. Next Button is used to play the next song in the playlist.
4. Previous button is used to play the previous song in the playlist selected song.
5. Selected Song by this feature we can select any songs in the playlist and that song will be started playing.
6. Replay button is used to start the currently playing song from beginning.
7. Shuffle button is used to shuffle all the songs in the playlist in a random order.
8. Minimise button is used to minimise the software to task bar.
9. Close button this button is used to exit the music player.
10. File menu in this a drop down menu appears which has option to add songs by clicking on the songs and then a dialogue box appears from which we can select the songs by respective formats.
11. Playlist here in our software is shown that consists of the selected song from the dialogue box and it is shown in a list box in our software.

3.1 Software requirements

Below are the minimum requirements of our software: -

- Minimum 512 MB Ram
- Celeron D processor
- 10 MB of space on hard disk
- One audio output port
- Some songs to play

4. TOOLS, PLATFORM AND LANGUAGES USED

4.1 FRONT END

The frontend or the user interface is developed by using C# language and the images and the buttons used are developed using Photoshop cc pro 2018 all the list box and the menus in the UI (user interface) are taken from the library of Visual Studio 2012.

4.2 BACK END

All the code for each and every function used in our music player is written in C# language and also a dynamic link library is included in our software i.e. Windows Media Player library. We have imported some of the functions through the dll files to make the code a little bit easy.

4.3 PLATFORM USED

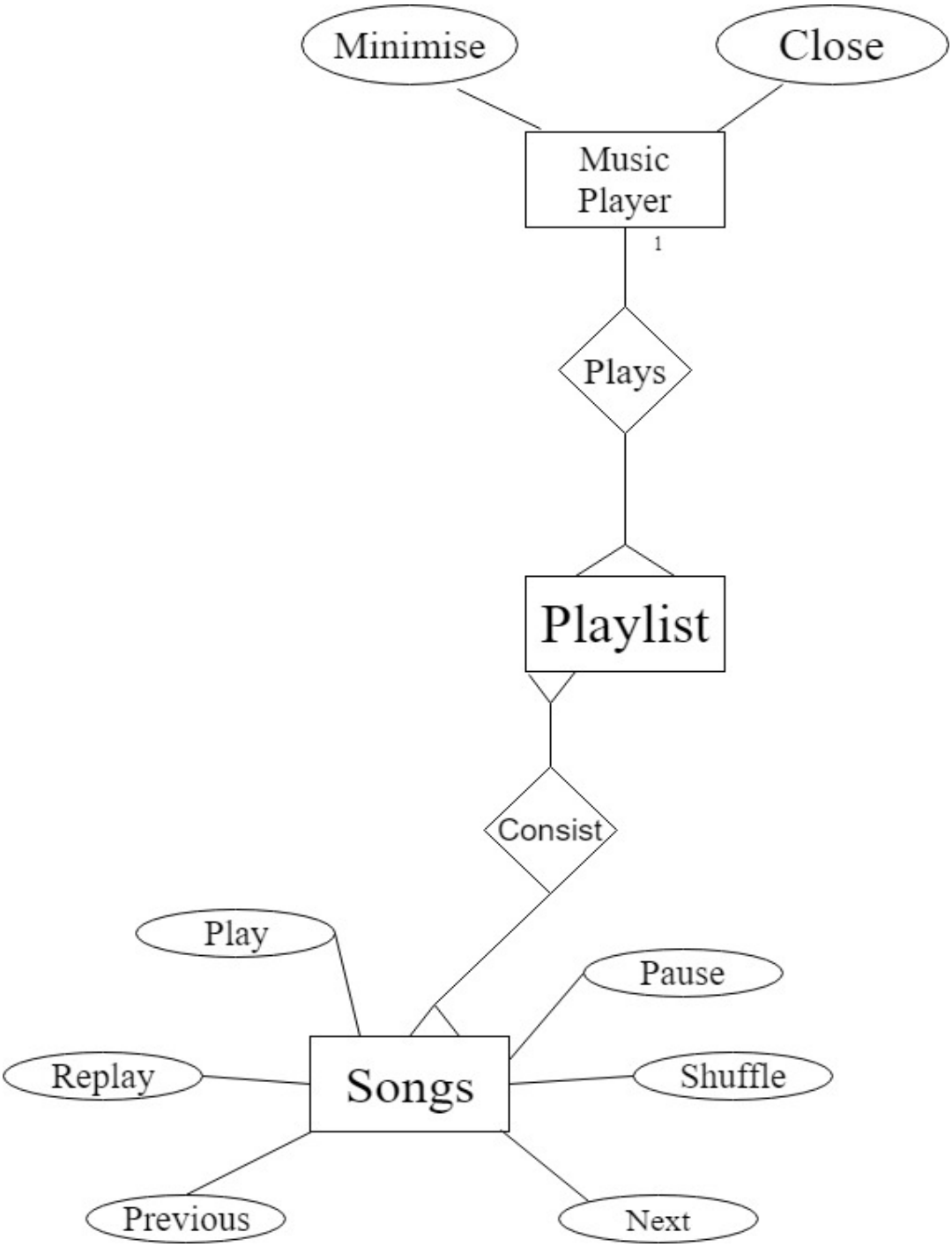
.NET platform is used for making the functions work in the music player and for the designing purpose we have used Adobe Photoshop CC 2018.

4.4 LANGUAGE USED

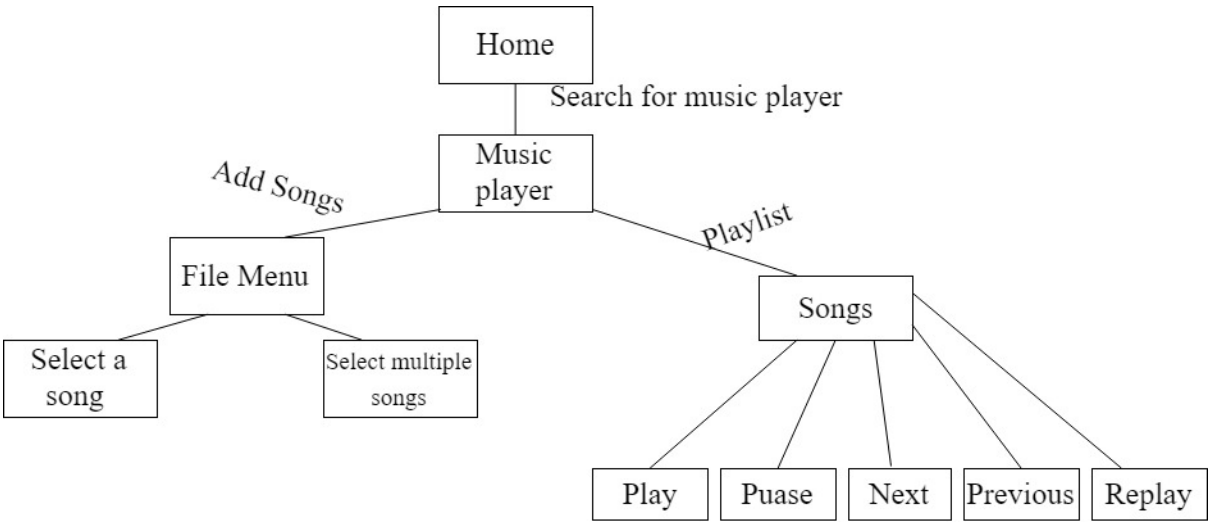
We have used C# that is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java.

5. SYSTEM DESIGN

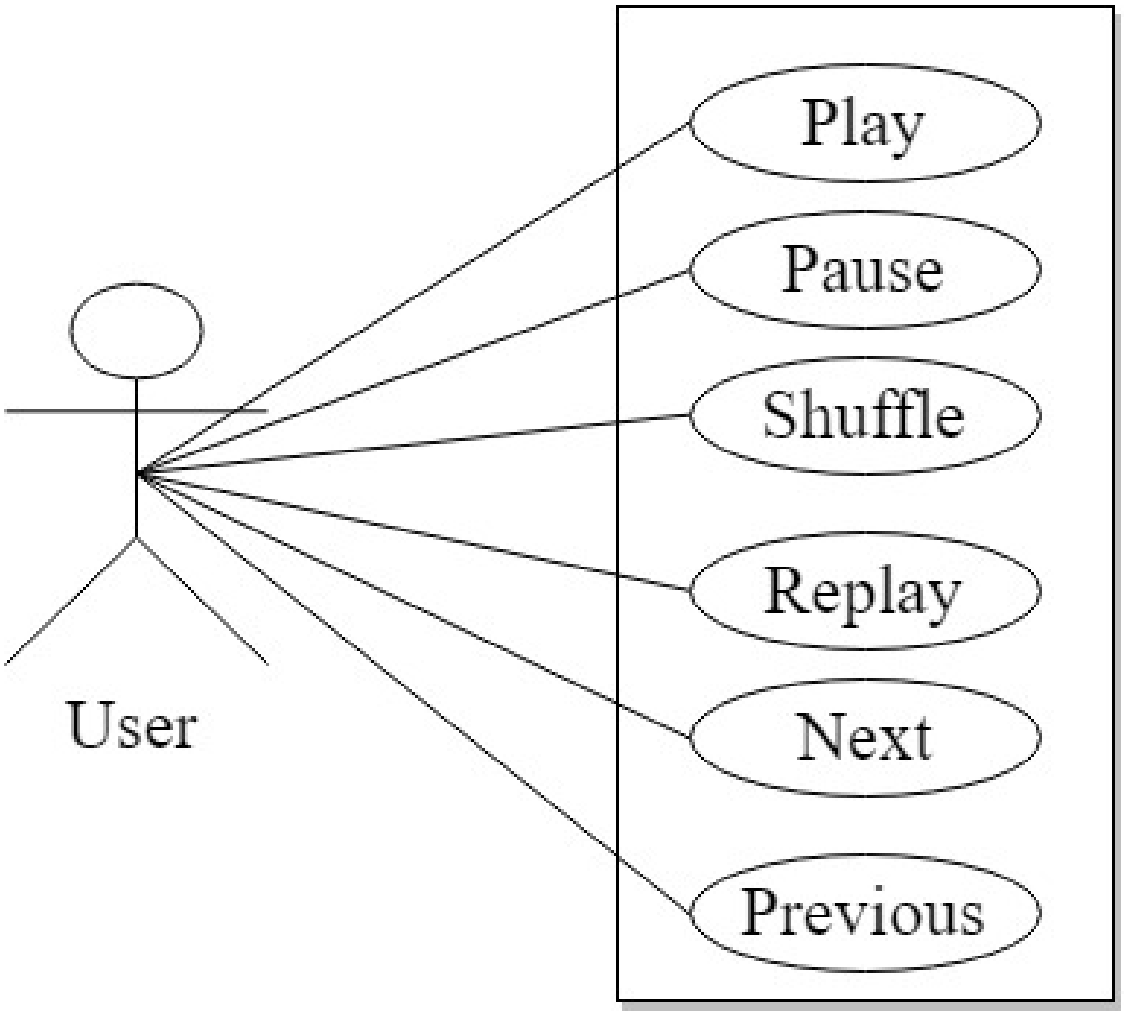
5.1 ER DIAGRAM OF MUSIC PLAYER



5.2 DECISION TREE DIAGRAM OF MUSIC PLAYER



5.3 USE CASE DIAGRAM OF MUSIC PLAYER



6. OUTPUTS

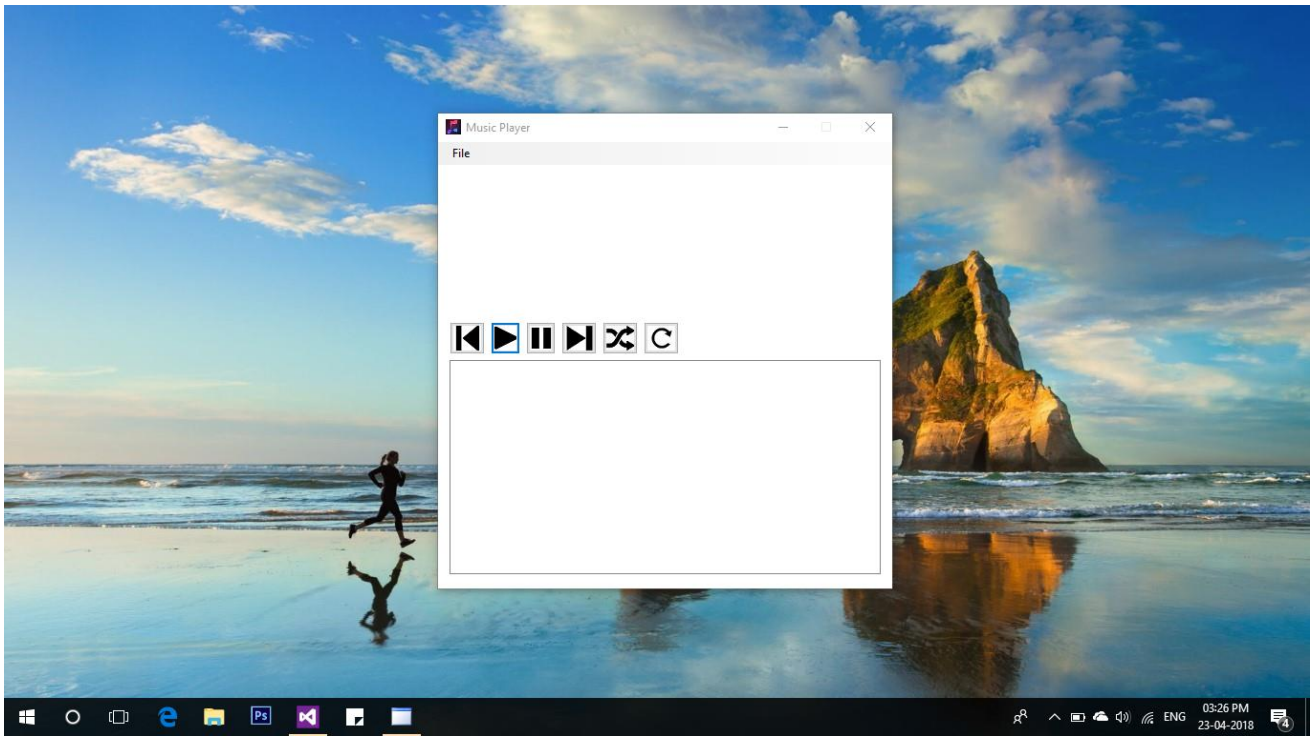


Figure 1: - Launch of Music Player

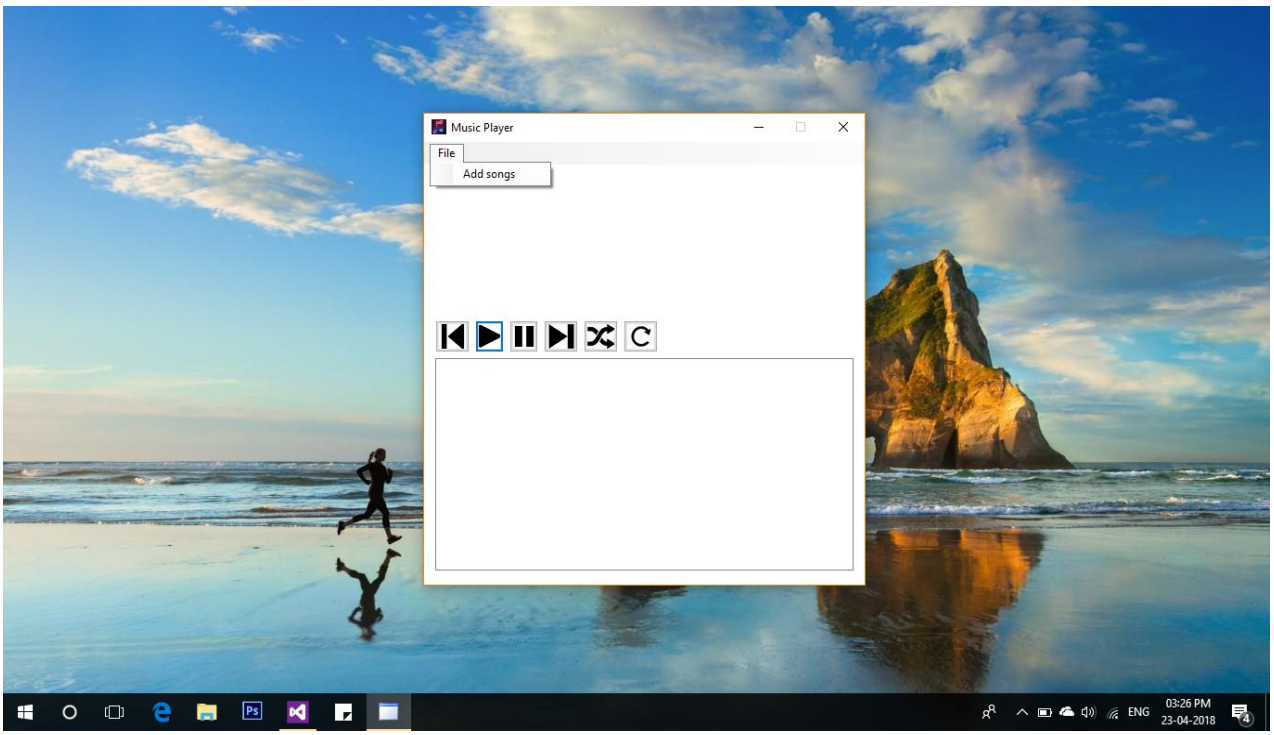


Figure 2: - Dropdown menu showing add songs

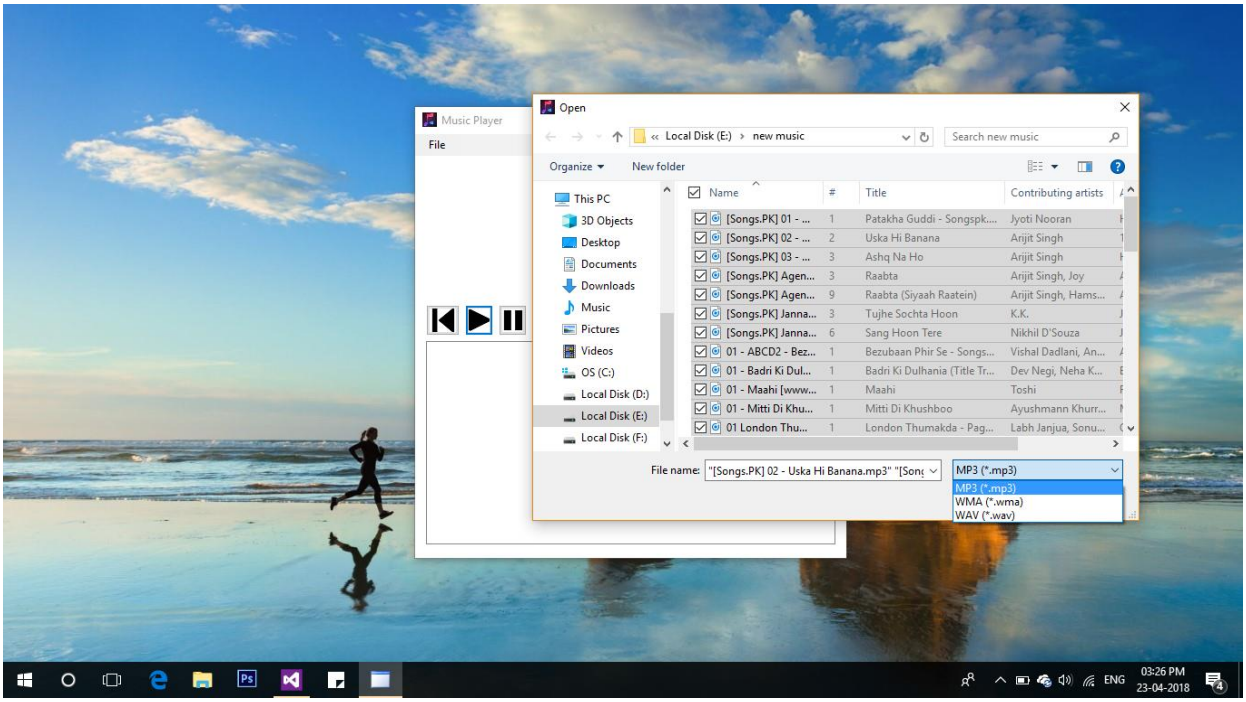


Figure 3: - Dialog Box Showing Multiple Selection of Songs

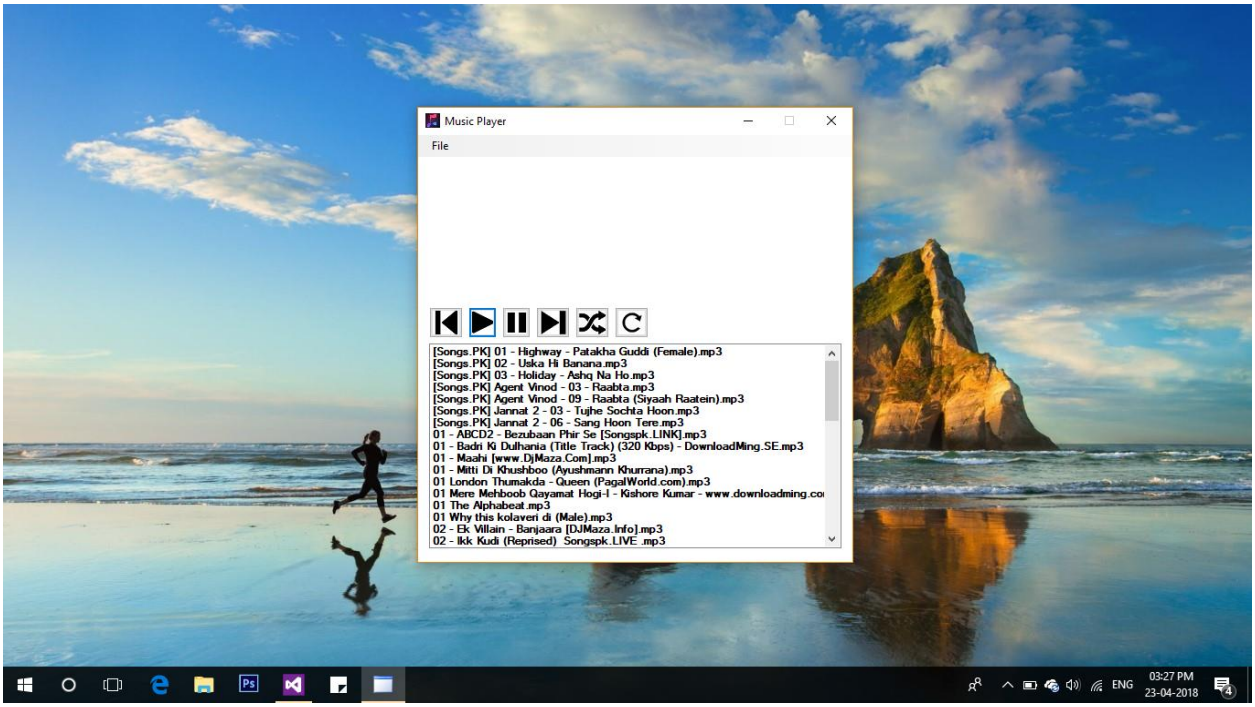


Figure 4: - Showing the playlist

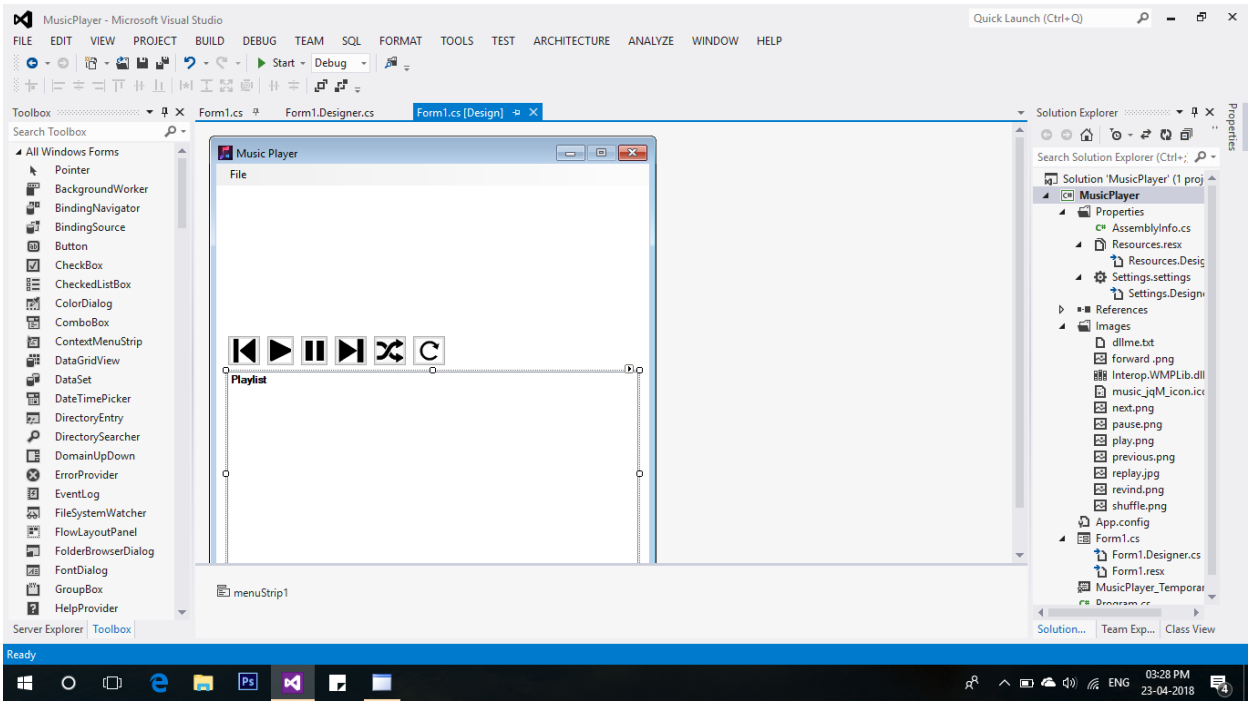


Figure 5: - UI Designed in Visual Studio 2012

7. RESULT

Development music player has been successfully accomplished all the features included are working properly the user can accomplish all the task with the few button clicks only

8. CONCLUSION

We have finally implemented the music player having the following features which are described as we have the provision for adding the songs with multiple song format support which are as follows:

- .Mp3
- .Wav
- .Wmv

Here, we can add multiple songs at a time to playlist. The objective of the music player is to allow the user to perform various functions such as:

- Play
- Pause
- Next
- Previous
- Shuffle
- Replay

This music player project mainly perform various actions related to a music file. Like when we click the play button the first song in the playlist will start playing. If the songs had been added in the playlist by the option of add songs present in the File menu.

Now when the song has been selected to play and it is been played then at the time of the playing of that song we can perform some operations on the songs in the playlist including the currently playing song.

The operations that can be performed are firstly pause i.e. it will temporarily stops the playing song. Secondly, we can press the next button to go to the next song in the playlist and vice-versa we can press the previous button for going to the previous song in the playlist. Also we have included an option for starting the song from beginning i.e. when the user will press the replay button. Now if the user wish to play any song in the playlist he/she can to go the song and simply select the song and the selected song will start playing. The last feature that had been added is the shuffle which is used to shuffle all the songs in the playlist.

9. REFERENCES

List of sources we have used on a Scavenger Hunt to build this software: -

- https://www.google.co.in/
- https://www.stackoverflow.com/
- https://www.tutorialspoint.com/csharp/
- https://msdn.microsoft.com/