

MUSIC PLAYER

A Project Report Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTERS OF COMPUTER APPLICATIONS

by

AJIT KUMAR

(University Roll no. 2100290140010)

CHIRAG SHARMA

(University Roll no. 2100290140055)

VIVEK SINGH

(University Roll no. 2100290140153)

HIMANSHU SAINI

(University Roll no. 2100290140071)

Under the Supervision of

Dr. Vipin Kumar

Associate Professor



to the

**Faculty of Computer Applications
KIET Group of Institutions, Ghaziabad
Uttar Pradesh – 201206**

DECLARATION

We hereby declare that the work presented in this report entitled “Music Player”, was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma from any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, and results, that are not our original contributions. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. We affirm that no portion of our work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

Signature

AJIT KUMAR (University Roll no. 2100290140010)

CHIRAG SHARMA (University Roll no. 2100290140055)

VIVEK SINGH (University Roll no. 2100290140153)

HIMANSHU SAINI (University Roll no. 2100290140071)

Department of Computer Applications

KIET Group of Institutions, Ghaziabad

Uttar Pradesh – 201206

CERTIFICATE

Certified that **AJIT KUMAR (University Roll no. 2100290140010)**, **CHIRAG SHARMA (University Roll no. 2100290140055)**, **VIVEK SINGH (University Roll no. 2100290140153)**, **HIMANSHU SAINI (University Roll no. 2100290140071)**, has carried out the research work entered in this project report entitled “Music Player” for the award of Master of Computer Applications from KIET Group of Institutions under my supervision. The project report embodies results of original work, and studies are carried out by the students themselves and the contents of the thesis do not form the basis for the award of any other degree to the candidates or to anybody else from this or any other University/Institution.

AJIT KUMAR (University Roll no.2100290140010)
CHIRAG SHARMA (University Roll no. 2100290140055)
VIVEK SINGH (University Roll no. 2100290140153)
HIMANSHU SAINI (University Roll no. 2100290140071)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Dr. Vipin Kumar
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner

Signature of External Examiner

Dr. Arun Tripathi
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

The purpose of Music Player is to automate the existing manual system with the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy access and manipulation of the same. The required software and hardware are easily available and easy to work with. Music Player, as described above, can lead to an error-free, secure, reliable, and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on record keeping. Thus, it will help the organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant while being able to reach the information. The aim is to automate its existing manual system with the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy access and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

In addition, depending on the user's usage scenario, the music player will also add some modes, such as driving mode and night mode, to allow users to use the application in any situation or environment. Moreover, the music player will have audio trim features, allowing users to trim the best part of their favourite song into phone ringtone or alarm. On the other hand, the existing music players pay less attention to the control of gestures. Therefore, the music player will solve the limitation by adding more gestures and shake the phone feature for media control to make it more user-friendly and humanity.

In a nutshell, the methodology for developing the mp3 music application used in this project is the agile development cycle. The agile development cycle consists of six phases, which is requirements analysis, planning, design, implementation or development, testing, and deployment. Due to the iterative and flexible nature of this approach, it is able to effectively adapt to users with changing requirements.

ACKNOWLEDGMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Dr. Vipin Kumar** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Tripathi, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

AJIT KUMAR (University Roll no. 2100290140010)

CHIRAG SHARMA (University Roll no. 2100290140055)

VIVEK SINGH (University Roll no. 2100290140153)

HIMANSHU SAINI (University Roll no. 2100290140071)

LIST OF ABBREVIATIONS

<i>Apps</i>	Applications
<i>CPU</i>	Central processing unit
<i>IOS</i>	iPhone Operating System
<i>MP3</i>	MPEG 1 Audio Layer 3
<i>UI</i>	User interface
<i>UTAR</i>	University Tunku Abdul Rahman
<i>VIP</i>	Very Important Person
<i>XML</i>	Full extensible markup language
<i>APK</i>	Android Application Package

Table of Contents

CHAPTER 1.....	1
INTRODUCTION	1
1.1 Project Description.....	2
1.2 Project Scope.....	2
1.3 Hardware/ Software used in the project	3
1.3.1 Hardware Requirement for development	3
1.3.2 Hardware Requirement for User.....	4
1.3.3 Software Requirement for development.....	4
1.3.4 Software Requirement for User	4
CHAPTER 2.....	5
FEASIBILITY STUDY	5
2.1 Technical Feasibility	5
2.2 Operational Feasibility	5
2.3 Economical Feasibility.....	6
CHAPTER 3.....	7
FLOWCHART/ USE CASE DIAGRAM	7
3.1 Flowchart.....	7
3.2 Use case Diagram.....	8
CHAPTER 4.....	9
WEBSITE DESIGN	9
4.1 Screenshots	9
CHAPTER 5.....	12
CODES.....	12
5.1 App.js.....	12
5.2 Index.css	15
5.3 Index.js	20
5.4 reportWebVitals.js	21
5.5 Control.js	21
5.6 Details.js	22
5.7 Player.js	23
5.8 Manifest.json	25

CHAPTER 6.....	27
MODULES OF MUSIC PLAYER.....	27
CHAPTER 7.....	30
INPUT DATA AND VALIDATION OF THE PROJECT.....	30
CHAPTER 8.....	31
SOFTWARE REQUIREMENT SPECIFICATION.....	31
8.1 Function Requirements.....	31
8.2 Non-Functional Requirements.....	32
CHAPTER 9.....	33
IDENTIFICATION OF NEED	33
CHAPTER 10.....	35
SYSTEM DESIGN	35
10.1 Primary Design Phase.....	35
10.2 Secondary Design Phase	36
CHAPTER 11.....	37
USER INTERFACE DESIGN	37
CHAPTER 12.....	38
PRELIMINARY PRODUCT DESCRIPTION.....	38
CHAPTER 13.....	39
LITERARY REVIEW.....	39
CHAPTER 13.....	41
SOFTWARE DEVELOPMENT LIFE CYCLE MODEL.....	41
CHAPTER 14.....	45
TESTING.....	45
CHAPTER 15.....	53
LIMITATION OF PROJECT.....	53
BIBLIOGRAPHY	54

CHAPTER 1

INTRODUCTION

The "Music Player" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides an error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Music Player, as described above, can lead to an error-free, secure, reliable, and fast management system. It can assist the user to concentrate on their other activities rather than concentrate on record keeping. Thus, it will help organizations in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and manage the information of Album, and Performer. Track, User, Music. Every Music Player has different Performer needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning and will help you ensure that your organization is equipped with the right level of information and details for your future goals.

Also, for those busy executives who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

1.1 Project Description

The main objective of the Project on Music Player is to manage the details of the Performer, Album, Album Type, Track, and Music. It manages all the information about Performer, User, Music, and Performer. The project is totally built at the administrative end and thus only the administrator is guaranteed access. The project aims to build an application program to reduce the manual work for managing the Performer, Album, User, and Album Type. It tracks all the details about the Album Type, Track, and Music. Functionalities provided by Music Player are as follows:

- Provides search facilities based on various factors. Such as Performer, Album Type, Track, Music.
- Music Player also manages the User details online for Track details, Music details, and Performers.
- It tracks all the information of Album, User, Track, etc.
- Manage the information on the Album.
- Shows the information and description of the Performer, Album Type.
- To increase the efficiency of managing the Performer, Album
- It deals with monitoring the information and transactions of Track
- Manage the information of Performer Editing, adding, and updating of Records is improved which results in proper resource management of Performer data.
- Manage the information on Track
- Integration of all records of Music.

1.2 Project Scope

It may help collect perfect management in detail. The collection will be obvious, simple, and sensible in a very short time. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Music Player. It will be also reduced the cost of collecting the management & collection

procedure will go on smoothly. Our project aims at Business process automation, i.e., we have tried to computerize various processes of Music Player.

- In the computer system the person must fill the various forms & the number of copies of the forms can be easily generated at a time.
- In a computer system, it is not necessary to create the manifest but we can directly print it, which saves our time.
- To assist the staff in capturing the effort spent on their respective working areas
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfies the user's requirement
- Be easy to understand by the user and operator
- Be easy to operate
- Have a good user interface
- Be expandable
- Delivered on schedule within the budget

1.3 Hardware/ Software used in the project

1.3.1 Hardware Requirement for development

- **Output Devices** - All basic output devices like printer, monitor, etc.
- **Secondary storage devices** -
 - HDD - 60GB or above.
 - Back-up - Flash Drive, CD/DVD, cloud storage etc.
- **Internal components** –
 - ▪ RAM - 512 MB (Minimum)
 - ▪ Processor - Intel Pentium 4 Processor or above

1.3.2 Hardware Requirement for User

- **Input Devices** - All basic input devices like keyboard, mouse, etc.
- **Output Devices** - All basic output devices like printer, monitor, etc.

1.3.3 Software Requirement for development

- **System Software**
 - Operating system (Windows XP or above)
- **Web Technologies**
 - HTML
 - CSS
 - Bootstrap
 - JavaScript
- **Frameworks**
 - React

1.3.4 Software Requirement for User

- **System Software**
 - Operating system (Windows XP or above)
- **Web Browser**
 - Microsoft Edge, Google Chrome

CHAPTER 2

FEASIBILITY STUDY

After doing the project Music Player, study and analyse all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time.

The feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

2.1 Technical Feasibility

This included the study of function, performance, and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied the complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using a different type of frontend and backend platforms.

2.2 Operational Feasibility

No doubt the proposed system is fully GUI based very user-friendly and all inputs to be taken all self-explanatory even to a layman. Besides, proper training has been conducted to let them know the essence of the system to the users so that they feel comfortable with the new system. As far as our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

2.3 Economical Feasibility

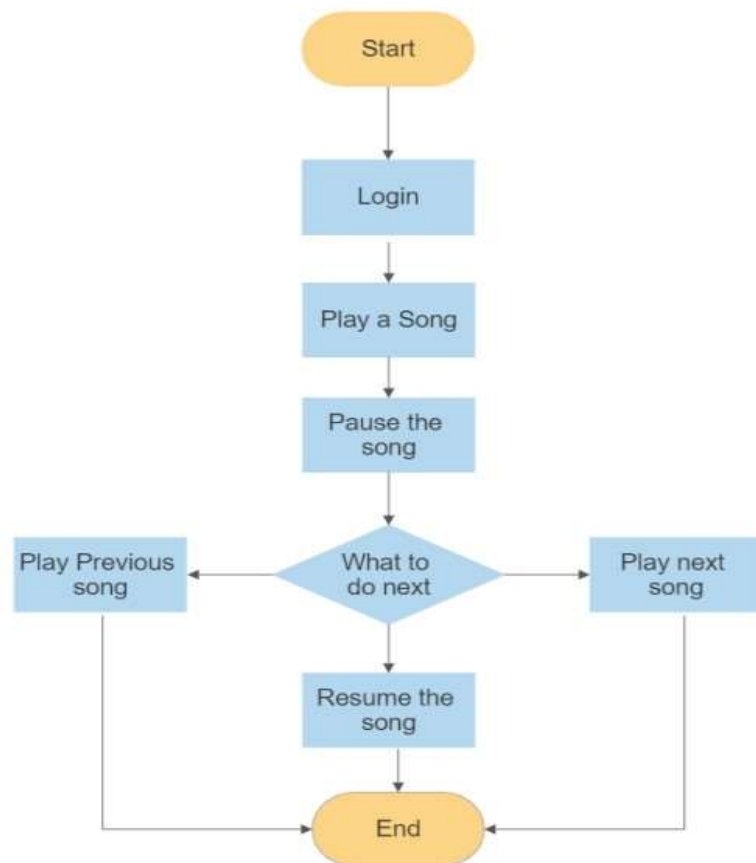
This is a very important aspect to be considered while developing a project. We decided on the technology based on the minimum possible cost factor.

- All hardware and software costs must be borne by the organization.
- Overall, we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and the later-on running cost for the system.

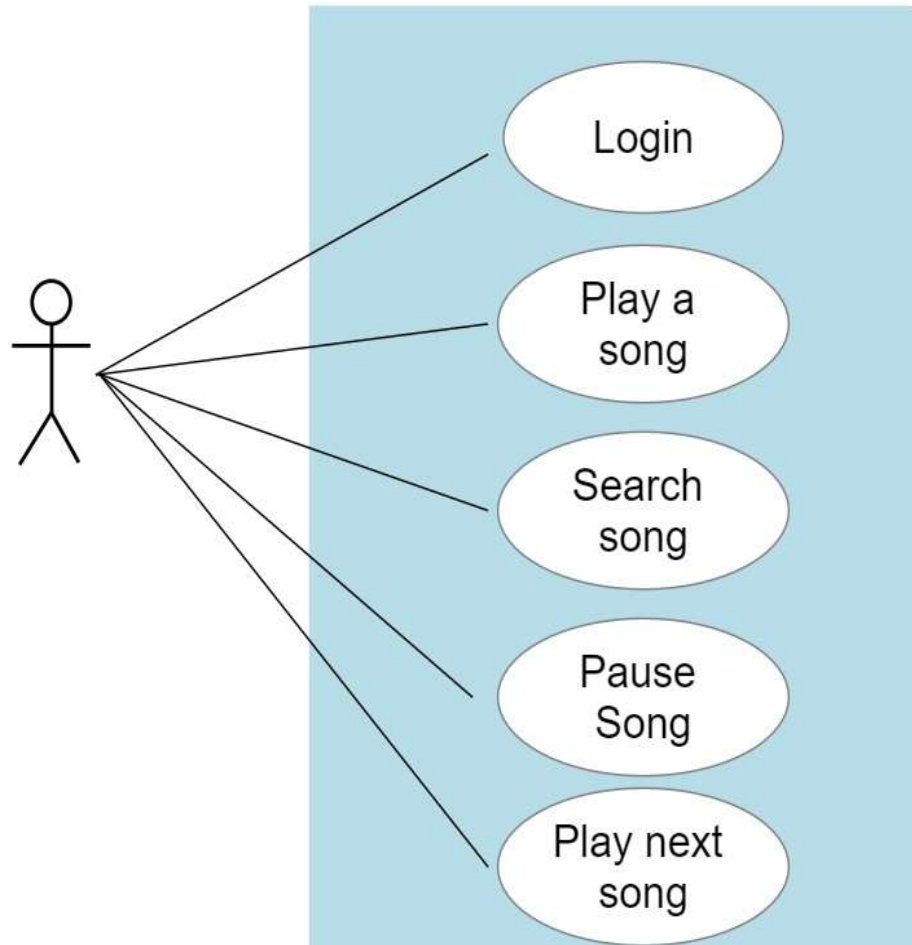
CHAPTER 3

FLOWCHART/ USE CASE DIAGRAM

3.1 Flowchart



3.2 Use case Diagram



CHAPTER 4

WEBSITE DESIGN

4.1 Screenshots

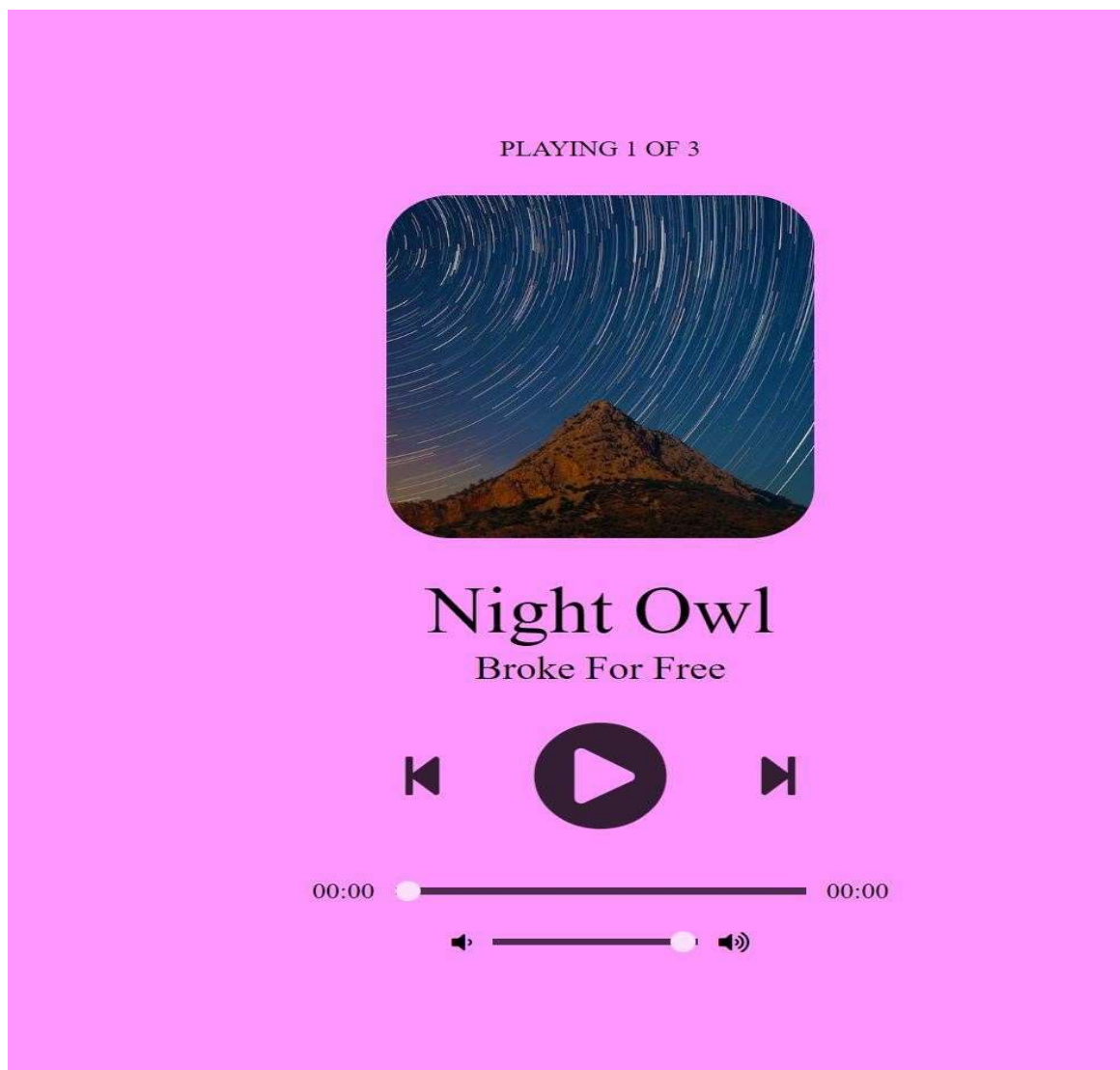



Fig 4.1

Login


Register

Your email

vivek@gmail.com

right

Your password

.....

right

LOG IN ➔

Forgot Password?

CLOSE

Fig 4.2

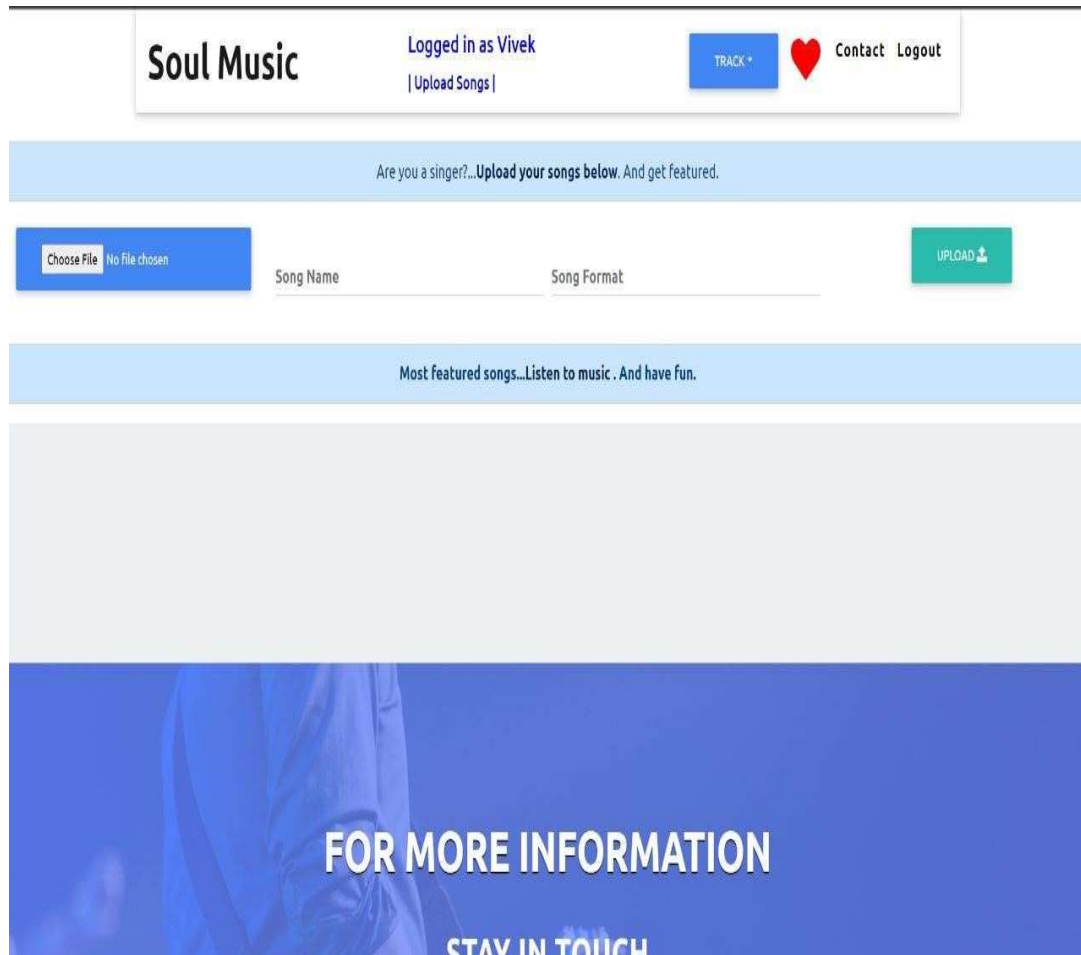


Fig 4.3

CHAPTER 5

CODES

5.1 App.js

This is the file for App Component. App Component is the main component in React which acts as a container for all other components.

```
import { useState, useEffect } from "react";
import Player from "../components/Player";

const [songs] = useState([
  {
    title: "Title:Ram siya ram",
    artist: "Artist:Ravindra jain",
    img_src: "../images/ramsiyaram.jpg",
    src: "../music/ram siya ram.mp3",
  },
  {
    title: "Title:Ram Ram",
    artist: "Artist:MC Square",
    img_src: "../images/maxresdefault.jpg",
    src: "../music/Shahar Ki Chhori Meri Le Le Ram Ram.mp3",
  },
  {
    title: "Title:Titaliya",
    artist: "Artist:Hardy Sandhu",
    img_src: "../images/titaliya.jpg",
    src: "../music/Titliaan Warga.mp3",
  },
]);
```

```

},
{
  title: "Title:Nach Meri Rani",
  artist: "Artist:Yo Yo Honey Singh",
  img_src: "./images/nach-meri-rani.jpg",
  src: "./music/Naach Meri Rani.mp3",

  {

    title: "Title:Care Ni Karda",
    artist: "Artist:Yo Yo Honey Singh",
    img_src: "./images/care-ni-karda.jpg",
    src: "./music/Care Ni Karda.mp3",
  },
  {
    title: "Title:Burj Khalifa",
    artist: "Artist:Shashi",
11
    img_src: "./images/burjkalifa.jpg",
    src: "./music/BurjKhalifa.mp3",
  },
  {
    title: "Title:Tango Del Fuego",
    artist: "Artist:Parov Stelar",
    img_src: "./images/ParovStelarGeorgiaGibbs-
TangoDelFuego.jpg",
    src: "./music/ParovStelarGeorgiaGibbs-
TangoDelFuego.mp3",
  },
  {
    title: "Title:Take You Dancing",

```

```

    artist: "Artist:Jason Derulo",
    img_src: "./images/JasonDerulo-TakeYouDancing.jpg",
    src: "./music/JasonDerulo-TakeYouDancing.mp3",
  },
  {
    title: "Title:Daisy",
    artist: "Artist:Ashnikko",
    img_src: "./images/Ashnikko-Daisy.jpg",
    src: "./music/Ashnikko-Daisy.mp3",
  },

  {
    title: "Title:Dolly Song",
    artist: "Artist:Partz Grimbard",
    img_src: "./images/PatzGrimbard-DollySong.jpg",
    src: "./music/PatzGrimbard-DollySong.mp3",
  },
]);

const [currentSongIndex, setCurrentSongIndex] =
useState(0);

const [nextSongIndex, setNextSongIndex] =
useState(0);

useEffect(() => {
  setNextSongIndex(() => {
    if (currentSongIndex + 1 > songs.length - 1) {
      return 0;
    } else {
      return currentSongIndex + 1;
    }
  });
}, [currentSongIndex, songs.length]);

```

```

return (
<div className="App">
  <Player
    currentSongIndex={currentSongIndex}
    setCurrentSongIndex={setCurrentSongIndex}
    nextSongIndex={nextSongIndex}
    songs={songs}
  />
</div>
);
}
export default App;

```

5.2 Index.css

Index.css file contains all styling, formatting and layout of pages.

```

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Fira Sans", sans-serif;
}
body {
  background: linear-gradient(
    0deg,
    rgba(195, 126, 34, 1) 0%,
    rgba(234, 253, 45, 0.5) 100%
  );
}
.App {

```

```

display: flex;
align-items: center;
justify-content: center;
min-height: 100vh;
width: 100%;
}
.c-player {
display: block;
background-image: linear-gradient(to bottom, violet,
indigo);
margin: 0px auto;
padding: 30px;
border-radius: 16px;
box-shadow: inset -6px -6px 12px rgba(0, 0, 0, 0.8),
inset 6px 6px 12px rgba(255, 255, 255, 0.4);
}

.c-player > h4 {
color: #fff;
font-size: 14px;
text-transform: uppercase;
font-weight: 500;
text-align: center;
}

.c-player > p {
color: #aaa;
font-size: 14px;
text-align: center;
font-weight: 600;
}

.c-player > p span {

```



```

    font-weight: 400;
    width: fit-content;
}

.c-player--details .details-img {
    position: relative;
    width: fit-content;
    margin: 0 auto;
}

.c-player--details .details-img img {
    display: block;
    margin: 50px auto;
    object-fit: contain;
    width: 100%;
    max-width: 250px;
    border-radius: 50%;
    box-shadow: 6px 6px 12px rgba(0, 0, 0, 0.8),
        -6px -6px 12px rgba(255, 255, 255, 0.4);
}

.c-player--details .details-img:after {
    content: "";
    display: block;
    position: absolute;
    top: -25px;
    left: -25px;
    right: -25px;
    bottom: -25px;
    border-radius: 50%;
14
    border: 3px dashed rgba(255, 206, 0, 1);
}

```

```
.c-player--details .details-title {  
  color: #eee;  
  font-size: 28px;  
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.8),  
    -2px -2px 4px rgba(255, 255, 255, 0.4);  
  text-align: center;  
  margin-bottom: 10px;  
}
```

```
.c-player--details .details-artist {  
  color: #aaa;  
  font-size: 20px;  
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.8),  
    -2px -2px 4px rgba(255, 255, 255, 0.4);  
  text-align: center;  
  margin-bottom: 20px;  
}
```

```
.c-player--controls {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  margin-bottom: 30px;  
}
```

```
.c-player--controls .play-btn {  
  display: flex;  
  margin: 0 30px;  
  padding: 20px;  
  border-radius: 50%;  
  box-shadow: 4px 4px 10px rgba(0, 0, 0, 0.8),
```

```

-4px -4px 10px rgba(255, 255, 255, 0.4),
inset -4px -4px 10px rgba(0, 0, 0, 0.4),
inset 4px 4px 10px rgba(255, 255, 255, 0.4);
border: none;
outline: none;
background-color: #ffce00;
color: #fff;
font-size: 24px;
cursor: pointer;
}

```

```

.c-player--controls .play-btn:hover {
  color: #ffce00;
15
  background-color: inherit;
}

```

```

.c-player--audio {
  margin-top: -20px;
  margin-bottom: 5px;
  outline-width: 0;
  opacity: 0.4;
}

```

```

.c-player--controls .skip-btn {
  background: none;
  border: none;
  outline: none;
  cursor: pointer;
  color: #888;
}

```

```

    font-size: 18px;
  }

.c-player--controls .skip-btn:hover {
  transition: 0.5s ease-in-out;
  transform: scale(1.1);
  color: white;
  background-color: inherit;
}

audio::-webkit-media-controls-play-button {
  display: none;
}

```

5.3 Index.js

Index.js file is just like index.html file, if no specification is provided a directory points to it's index file. Index files are equivalent doesn't matter if it's java script or typescript. for convenience we'll use index.

```

import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')

```

```

);

// If you want to start measuring performance in your app,
pass a function

// to log results (for example:
reportWebVitals(console.log))

// or send to an analytics endpoint. Learn more:
https://bit.ly/CRA-vitals

reportWebVitals();

```

5.4 reportWebVitals.js

```

const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP,
    getLCP, getTTFB }) => {
      getCLS(onPerfEntry);
      getFID(onPerfEntry);
      getFCP(onPerfEntry);
      getLCP(onPerfEntry);
      getTTFB(onPerfEntry);
    });
  }
};

export default reportWebVitals;

```

5.5 Control.js

```

import React from "react";

import { FontAwesomeIcon } from "@fortawesome/react-
fontawesome";

import {
  faPlay,

```

```

    faPause,
    faForward,
    faBackward,
  } from "@fortawesome/free-solid-svg-icons";
function Controls(props) {
  return (
    <div className="c-player--controls">
      <button className="skip-btn" onClick={() =>
        props.SkipSong(false)}>
        <FontAwesomeIcon icon={faBackward} />
      </button>
      <button
        className="play-btn"
        onClick={() => props.setIsPlaying(!props.isPlaying)}
      >
        <FontAwesomeIcon icon={props.isPlaying ? faPause :
        faPlay} />
      </button>
      <button className="skip-btn" onClick={() =>
        props.SkipSong()}>
        <FontAwesomeIcon icon={faForward} />
      </button>
    </div>
  );
}
export default Controls;

```

5.6 Details.js

```

import React from "react";
function Details(props) {
  return (

```

```

<div className="c-player--details">
  <div className="details-img">
    <img src={props.song.img_src} alt="" />
  </div>
  <h3 className="details-title">{props.song.title}</h3>
  <h4 className="details-artist">{props.song.artist}</h4>
</div>
);
}
export default Details;

```

5.7 Player.js

```

import React, { useState, useRef, useEffect } from
"react";

import Controls from "../Controls";
import Details from "../Details";

function Player(props) {
  const audioEl = useRef(null);
  const [isPlaying, setIsPlaying] = useState(false);
  useEffect(() => {
    if (isPlaying) {
      audioEl.current.play();
    } else {
      audioEl.current.pause();
    }
  });

  const SkipSong = (forwards = true) => {
    if (forwards) {
      props.setCurrentSongIndex(() => {

```

```

    let temp = props.currentSongIndex;
    temp++;
    if (temp > props.songs.length - 1) {
      temp = 0;
    }
    return temp;
  });
}
else {
  props.setCurrentSongIndex(() => {
    let temp = props.currentSongIndex;
    temp--;
    if (temp < 0) {
      temp = props.songs.length - 1;
    }
    return temp;
  });
}
};

return (
  <div className="c-player">
    <h4>Soul Music : Playing now</h4>
    <Details song={props.songs[props.currentSongIndex]}
  />

```

```

<Controls
  isPlaying={isPlaying}
  setIsPlaying={setIsPlaying}

```



```

SkipSong={SkipSong}
/>

<audio
  className="c-player--audio"
  src={props.songs[props.currentSongIndex].src}
  ref={audioEl}
  controls
></audio>
<p>

Next up:{" "}
<span>
  {props.songs[props.nextSongIndex].title} by{" "}
  {props.songs[props.nextSongIndex].artist}
</span>
</p>
</div>

);
}
export default Player;

```

5.8 Manifest.json

```

{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",

```

```
"type": "image/x-icon"
},
{
  "src": "logo192.png",
  "type": "image/png",
  "sizes": "192x192"
},
{
  "src": "logo512.png",
  "type": "image/png",
  "sizes": "512x512"
}
],
"start_url": ".",
"display": "standalone",
"theme_color": "#000000",
"background_color": "#ffffff"
}
```

CHAPTER 6

MODULES OF MUSIC PLAYER

A music player typically has several modules or components that work together to play music. Some of these modules include:

1. User interface:

This module allows users to navigate and interact with the music player, such as selecting songs, adjusting the volume, and creating playlists.

2. Login module:

This module allows users to login to his/her account where he enjoy music.

3. Signup module:

This module allows users to create new account.

4. Audio decoding:

This module is responsible for decoding the audio file format, such as mp3 or AAC, and converting it into a format that can be played by the player.

5. Audio output:

This module is responsible for sending the audio signal to the speaker or headphones.

6. Playback control:

This module controls the playback of the music, such as play, pause, skip, and repeat.

7. Playlist management:

This module allows users to create and manage playlists of songs.

8. Library management:

This module is responsible for organizing and cataloging the songs in the player's library.

9. Equalization:

This module allows users to adjust the balance of different frequencies in the audio signal, such as bass and treble.

10. Audio effects:

This module allows users to add effects to the audio signal, such as reverb and distortion.

11. Upload Module:

This module allows users to add music and upload music on site which is available for other users.

12. File Browser:

This module allows the user to browse and select audio files stored on the device or on external storage devices.

13. Playlist Management:

This module allows the user to create, edit, and manage playlists of audio files.

14. Metadata Management:

This module manages the metadata associated with the audio files, such as song title, artist, and album information.

15. Album Art:

This module displays album art associated with the currently playing audio file.

16. Lyrics Display:

This module displays lyrics associated with the currently playing audio file.

17. Podcast Support:

This module allows the user to subscribe to and play podcasts.

18. Radio Support:

This module allows the user to listen to and manage radio stations.

19. Albums, Artists, Genres, and Albums Views:

This module allows the user to browse music by different categories such as album, artist, genre and so on.

The specific modules included in a music player may vary depending on the device or software. Some music players may also include additional features such as support for different audio formats, built-in effects, and advanced equalization options.

These are some of the basic modules that a music player has, depending on the type of player, it may have more or less features.

CHAPTER 7

INPUT DATA AND VALIDATION OF THE PROJECT

- All the fields such as Performer, Album Type, and Music are validated and do not take invalid values
- Each form for Performer, Album, and User cannot accept blank value fields
- Avoiding errors in data Controlling the amount of input
- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing is done manually.
- Recording of all the reproduced errors.
- Modifications are done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.
- Functionality of the entire module/forms.
- Validations for user input.
- Checking of the Coding standards to be maintained during coding.

CHAPTER 8

SOFTWARE REQUIREMENT SPECIFICATION

The Software Requirements Specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed functional and behavioural description, an indication of performance requirements and design constraints, appropriate validation criteria, and other data pertinent to requirements. The proposed system has the following requirements:

8.1 Function Requirements

1. User Interface:

The music player should have a user-friendly interface that allows the user to easily browse and select songs, create playlists, and adjust settings.

2. Media Library:

The music player should support a wide variety of audio file formats and should be able to read and display metadata such as song title, artist, and album information.

3. Playback Control:

The music player should support basic playback controls such as play, pause, rewind, and fast forward.

4. Audio Output:

The music player should be able to output audio through a variety of devices including speakers, headphones and so on.

5. Equalizer:

The music player should allow the user to adjust the balance of different frequency bands in the audio signal.

6. Networking:

The music player should be able to connect to other devices and networks, such as streaming music from the internet or wirelessly connecting to other speakers.

8.2 Non-Functional Requirements

1. Performance:

The music player should have minimal latency and be able to play audio files smoothly and without interruption.

2. Compatibility:

The music player should be compatible with a wide variety of devices and operating systems.

3. Security:

The music player should protect the user's privacy and prevent unauthorized access to the user's media library and playback history.

4. Usability:

The music player should be easy to use and understand, even for non-technical users.

5. Reliability:

The music player should be reliable and have minimal downtime.

6. Power management:

The music player should have power management features such as battery optimization, sleep mode and so on.

This is just a general list and the specific requirements for a music player will depend on the intended users, devices, and features. The SRS should be a detailed document that includes all the requirement and constraints for the software development team.

CHAPTER 9

IDENTIFICATION OF NEED

The old manual system was suffering from a series of drawbacks. Since the whole of the system was to be maintained with hands the process of keeping, maintaining, and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order. there used to be lots of difficulties in associating any particular transaction with a particular context. If any information was to be found it was required to go through the different registers, and documents there would never exist anything like report generation. There would always be unnecessary consumption of time while entering records and retrieving records. One more problem was that it was very difficult to find errors while entering the records. Once the records were entered it was very difficult to update these records.

The reason behind it is that there is a lot of information to be maintained and has to be kept in mind while running the business For this reason we have provided features Present system is partially automated (computerized), actually, the existing system is quite laborious as one has to enter same information at three different places.

The following points should be well considered:

- Documents and reports that must be provided by the new system: there can also be a few reports, which can help management in decision-making and cost control, but since these reports do not get the required attention, such kinds of reports and information were also identified and give the required attention.
- Details of the information needed for each document and report.
- The required frequency and distribution for each document.

- Probable sources of information for each document and report.
- With the implementation of a computerized system, the task of keeping records in an organized manner will be solved. The greatest of all is the retrieval of information, which will be at the click of the mouse. So the proposed system helps in saving time in different operations and making information flow easy giving valuable reports.

CHAPTER 10

SYSTEM DESIGN

In this phase, a logical system is built which fulfils the given requirements. The design phase of software development deals with transforming the clients' requirements into a logically working system.

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system. System Analysis is the process that decomposes a system into its component pieces for the purpose of defining how well those components interact to accomplish the set requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

Normally, design is performed in the following two steps:

10.1 Primary Design Phase

In this phase, the system is designed at the block level. The blocks are created based on analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between blocks. Thus, all activities which require more interaction are kept in one block.

10.2 Secondary Design Phase

In the second phase, the detailed design of every block is performed.

The general tasks involved in the design process are the following:

- Design various blocks for overall system processes.
- Design smaller, compact and workable modules in each block
- Design various database structures
- Specify details of programs to achieve the desired functionality
- Design the form of inputs, and outputs of the system.
- Perform documentation of the design.
- System reviews

CHAPTER 11

USER INTERFACE DESIGN

User Interface Design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventual presentation of desired inputs and outputs. The overall flow of screens and messages is called dialogue.

The following steps are various guidelines for User Interface Design:

- The system user should always be aware of what to do next.
- The screen should be formatted so that various types of information, and instructions messages always appear in the same general display area.
- Messages, instructions, or information should be displayed long enough to allow the system user to read them.
- Use display attributes sparingly.
- Default values for fields and answers to be entered by the user should be specified.
- A user should not be allowed to proceed without correcting an error.
- The system user should never get an operating system message or fatal error.

CHAPTER 12

PRELIMINARY PRODUCT DESCRIPTION

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of the project request and make an informed judgment about the feasibility of the proposed project.

Analysts working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request.
- Determine the size of the project.
- Assess the costs and benefits of alternative approaches.
- Determine the technical and operational feasibility of alternative approaches.
- Report the findings to management, with recommendations outlining the acceptance or rejection of the proposal

CHAPTER 13

LITERARY REVIEW

The following research journals from AKTU Nalanda E-Consortium were used as references for completing this project:

1. Advances in Engineering Software:

<https://www.sciencedirect.com/journal/advances-inengineering-software>

2. International Journal on Software Tools for Technology Transfer:

<https://www.springer.com/journal/10009>

3. Optimization Methods and Software:

<https://www.tandfonline.com/journals/goms20>

4. Software Quality Journal:

<https://www.springer.com/journal/11219/>

5. Journal of Web Semantics:

<https://www.sciencedirect.com/journal/journal-of-web-semantics>

6. Start Programming Using HTML, CSS, and JavaScript:

<https://www.taylorfrancis.com/books/mono/10.1201/b19402/start-programming-usinghtml-css-javascript-iztok-fajfar>

7. HTML5 Mobile Websites:

<https://www.taylorfrancis.com/books/mono/10.4324/9780240818146/html5-mobilewebsites-matthew-david>

8. Managing Web Projects:

<https://www.taylorfrancis.com/books/mono/10.1201/9781439804964/managing-webprojects-edward-farkas>

9. Web Programming for Business:

<https://www.taylorfrancis.com/books/mono/10.4324/9780203582084/web-programmingbusiness-david-paper>

10. Web Technology:

<https://www.taylorfrancis.com/books/mono/10.1201/9781351029902/web-technologyakshi-kumar>

11. Web Security: <https://www.taylorfrancis.com/books/mono/10.1201/b18327/web-securityhanqing-wu-liz-zhao>
12. E-Commerce Usability:
<https://www.taylorfrancis.com/books/mono/10.1201/9780203245910/commerce-usabilitydavid-travis>
13. Innovations in E-Systems for Business and Commerce:
<https://www.taylorfrancis.com/books/edit/10.1201/9781315207353/innovations-systemsbusiness-commerce-abdelkhalak-el-hami-seifedine-kadry>
14. Cloud Database Development and Management:
<https://www.taylorfrancis.com/books/mono/10.1201/b15264/cloud-database-developmentmanagement-lee-chao>
15. NoSQL Database for Storage and Retrieval of Data in Cloud:
<https://www.taylorfrancis.com/books/edit/10.1201/9781315155579/nosql-ganesh-chandradek>

CHAPTER 13

SOFTWARE DEVELOPMENT LIFE CYCLE MODEL

Waterfall Model

The waterfall model is a well-known structured methodology for software development. The whole process of system development is divided into distinct phases. The model has been introduced in 1970s. Every phase has a unique output. It was the first SDLC model to be used widely. So that, sometime it is referred to Waterfall by SDLC. The waterfall model is used when the system requirements are well known, technology is understood and the system is a new version of an existing product (Dennis, Wixom and Roth, 2012).

Mainly there are six phases in Waterfall model. If there is a problem faced in any phase of the cycle, the system goes to the previous phase. The phases of Waterfall method is:

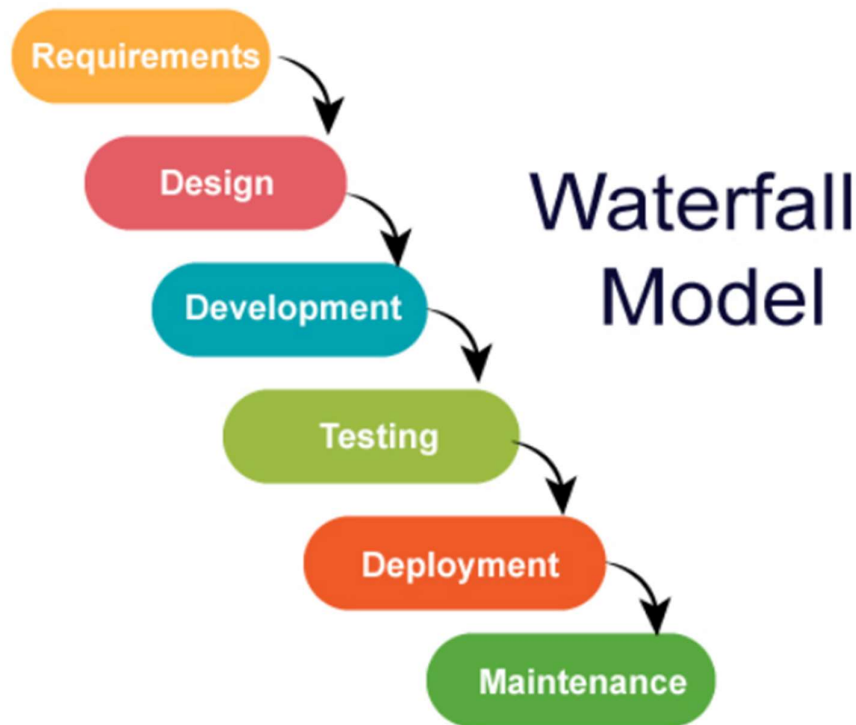


Figure: 13.1

Requirements Gathering & Analysis:

In this phase, the project title had been selected. The project title for the system is Music player. This project starting with brainstorming ideas with supervisor and proposed the title of the project. An abstract and description of the project module has also been done and attached. Besides, the Gantt chart also needed as a guideline and references for the project. This phase is to analyze the existing system and the article of the techniques or method that will be used for this project. In this phase also get all the requirements that are needed to design and develop the new system. Based on the collection of information through article, method and technique that is suitable been decided.

System Design:

The requirements documented in previous phase are studied in this phase and the system design is prepared. All the data or requirement obtained during planning and analysis phase transformed into the design.

Implementation:

With inputs from system design, the system is developed in several units. Then the units are tested. This phase is where the design will implement into the coding. The system will develop regarding the user and system requirement. In this project, to develop the system will be use Visual Studio Code to code, Bootstrap as framework . This phase is a critical phase because user part needed to fulfil and to make sure the objectives accomplish.

Integration & Testing:

The units of the program developed in previous phase are integrated into a system. Then the whole system is tested. This testing phase will test the system to check the error and ensure the function run well as a whole system. Any error or bugs will be fixed and repeated testing the system until all the function can be use.

Deployment of the system:

This phase is when the system has successfully done and fulfil all the objective. The system can be deployed and finally the system will publish to the user for use as their need. Deployment is the mechanism through which applications, modules, updates, and patches are delivered from developers to users. The methods used by developers to build, test and deploy new code will impact how fast a product can respond to changes in customer preferences or requirements and the quality of each change.

Maintenance:

There are some issues which are found in the client environment. Patches are

released to fix those issues. System maintenance is an ongoing activity, which covers a wide variety of activities, including removing program and design errors, updating documentation and test data and updating user support.

CHAPTER 14

TESTING

Testing is vital for the success of any software. No system design is ever perfect. Testing is also carried in two phases, first is during the software engineering that is during the module creation, second phase is after the completion of software, this is system testing which verifies that the whole set of programs hanged together.

White Box Testing:

In this technique, the close examination of the logical parts through the software are tested by cases that exercise species sets of conditions or loops. All logical parts of the software checked once. Errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops. White box testing tests all the independent parts within a module and logical decisions on their true and the false side are exercised. All loops and bound within their operational bounds were exercised and internal data structure to ensure their validity were exercised once.

Black Box Testing:

This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. Black Box tests the input, the output and the external data. It checks whether the input data is correct and whether we are getting the desired output.

Alpha Testing:

Acceptance testing is also sometimes called alpha testing. Be spoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is an acceptable implementation of the system requirements.

Beta Testing:

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

Unit Testing:

Each module is considered independently. It focuses on each until of software as implemented in the source code. It is white box testing.

Integration Testing:

Integration Testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules. Modules are integrated by using the top-down approach.

Validation Testing:

Validation testing was performed to ensure that all the functional and performance requirements are met. The validation testing ensures that all functional requirements as stated in the Software Requirement Specification Document are properly implemented in the application.

System Testing:

It is executing programs to check logical changes made in it with intention of finding errors. A system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

Implementation and Software Specification Testing

Detailed Design of implementation

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, trains users and implements extensive testing procedures. To evaluate design and operating specifications and/or provide the basis for further modification.

Technical Design

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

Test Specifications and Planning

This activity prepares detailed test specifications for individual modules and programs, job systems, subsystems, and for the system as a whole.

Programming and Testing

This activity encompasses actual development, writing, and testing of program units or modules.

User Training

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

Acceptance Test

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

Installation Phase

In this phase the new Computerized system is installed the conversion to new procedures is fully implemented and the potential of the new system is explored.

System Installation

The process of starting the actual use of a system and training user personnel in its operation.

Review Phase

This phase evaluates the successes and failures during a systems development project, and to measure the results of a new Computerized system in terms of benefits and savings projected at the start of the project.

Development Recap

A review of a project immediately after completion to find successes and potential problems in future work.

Post-Implementation Review

A review, conducted after a new system has been in operation for some time, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

THE STEPS IN THE SOFTWARE TESTING

The steps involved during Unit Testing are as follows:

- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.

- Complete code review of the module.
- Actual testing done manually.
- Modifications done for the errors found during testing.
- Prepared the test result scripts.

The unit testing done included the testing of the following items:

- Functionality of the entire module/forms.
- Validations for user input.
- Checking of the Coding standards to be maintained during coding.
- Testing the module with all the possible test data.
- Testing of the functionality involving all type of calculations etc.
- Commenting standard in the source files.

After completing the Unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While System Integration, we integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

The steps involved during System testing are as follows:

- Integration of all the modules/forms in the system.
- Preparation of the test cases.
- Preparation of the possible test data with all the validation checks.
- Actual testing done manually.
- Recording of all the reproduced errors.
- Modifications done for the errors found during testing.
- Prepared the test result scripts after rectification of the errors.

The System Testing done included the testing of the following items:

- Functionality of the entire system as a whole.
- User Interface of the system.
- Testing the dependent modules together with all the possible test data scripts.
- Verification and Validation testing.
- Testing the reports with all its functionality.

After the completion of system testing, the next following phase was the Acceptance Testing Clients at their end this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

There are other six tests, which fall under special category. They are described below:

• **Peak Load Test:**

It determines whether the system will handle the volume of activities that occur when the system is at the peak of its processing demand. For example, test the system by activating all terminals at the same time.

• **Storage Testing:**

It determines the capacity of the system to store transaction data on a disk or in other files.

- **Performance Time Testing:**

It determines the length of time system used by the system to process transaction data. This test is conducted prior to implementation to determine how long it takes to get a response to an inquiry, make a backup copy of a file, or send a transmission and get a response.

- **Recovery Testing:**

This testing determines the ability of user to recover data or re-start system after failure. For example, load backup copy of data and resume processing without data or integrity loss.

- **Procedure Testing:**

It determines the clarity of documentation on operation and users of system by having users do exactly what manuals request. For example, powering down system at the end of week or responding to paper-out light on printer.

- **Human Factors Testing:**

It determines how users will use the system when processing data or preparing reports

CHAPTER 15

LIMITATION OF PROJECT

Although I have put my best efforts to make the software flexible, easy to operate but limitations cannot be ruled out even by me. Though the software presents a broad range of options to its users some intricate options could not be covered into it, partly because of lack of sophistication. Paucity of time was also major constraint; hence it was not possible to make the software fool proof and dynamic. Lack of time also compelled me to ignore some part such as storing old result of the candidate etc.

Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it is acknowledged that a layman may find it a bit problematic at the first instance. The user is provided help at each step for his convenience in working with the software.

List of limitations which is available in the Music Player:

- Excel export has not been developed for Music playlists due to some criticality.

BIBLIOGRAPHY

- Google:
https://www.irjmets.com/uploadedfiles/paper//issue_6_june_2022/26682/final/final_irjmets1656008098.pdf
<https://pub.tik.ee.ethz.ch/students/2009-FS/SA-2009-10.pdf>
- Coursehero: <https://www.coursehero.com/file/74580183/reportpdf/>
- https://courses.cs.washington.edu/courses/cse403/08wi/projects/sds/web_music_player.ppt
- Stack overflow: <https://stackoverflow.com/questions/54377045/music-player-application>
<https://stackoverflow.com/questions/68154003/music-player-in-background>