

A  
PROJECT REPORT  
ON

# **ONLINE MUSIC APPLICATION**

**By**

VIDIT SHAH (CE-03) (19CEUON075)  
BHAVIN CHAVDA (CE-23) (19CEUBG024)

**B.Tech CE Semester-V**

**Subject: ADVANCED TECHNOLOGIES (CE-515)**

**Guided by:**

Prof. Prashant M. Jadav (Associate Professor)

Prof. Siddharth P. Shah (Assistant Professor)

Dept. of Computer Engg.





**Faculty of Technology Department of Computer  
Engineering**

**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical / term work carried out in  
the subject of **ADVANCED TECHNOLOGIES (CE-515)** and  
recorded in this journal is the bonafide work of

VIDIT SHAH (CE-03) (19CEUON075)

CHAVDA BHAVIN (CE-23) (19CEUBG024)

of B.Tech Semester **V** in the branch of **Computer Engineering**  
during the academic year **2021-2022**.

**Project Guide:**

Prof. Prashant M. Jadav

Associate Professor

Dept. of Computer Engg.,

Faculty of Technology

Dharmsinh Desai University, Nadiad

Prof. Siddharth P. Shah

Assistant Professor

Dept. of Computer Engg.,

Faculty of Technology

Dharmsinh Desai University, Nadiad

# INDEX

---

1. Abstract .....	1
2. Introduction.....	2
2.1 Brief Introduction.....	2
2.2 Tools/Technologies Used .....	2
3. Software Requirement Specifications .....	3
4. Design.....	6
4.1 Use Class Diagram:.....	6
4.2 Activity Diagram:.....	7
4.3 Class Diagram: .....	8
4.4 DFD Model: .....	9
4.5 State Diagram:.....	11
5. Implementation Details.....	12
5.1 Login Page: .....	12
5.2 Home Page:.....	12
5.3 Search Page: .....	12
5.4 Profile Page: .....	12
5.5 Function prototypes: .....	13
6. Screenshots.....	18
7. Conclusion.....	21
8. Limitation and Future Enhancements .....	22
9. Reference / Bibliography .....	22

# 1. Abstract

---

**TuneTide** is an online music player. We have added some songs which can be played through our platform. User can create its own playlist and add multiple songs to it. You can search different albums, songs and artist and play them. There are multiple features like "Most played song", "Liked song".

## 2. Introduction

---

### 2.1 Brief Introduction

Online music players are basically those platforms which enable you to play music on the internet without downloading any player or song. These are music streaming platforms from where we can stream and listen to any song, any time. Playlist makes it user friendly and helps user to quickly play the song of his favourite choice. Newly added songs can be viewed separately. User can give preference to language and enjoy music of his/her preferred language.

### 2.2 Tools/Technologies Used

#### **Technologies:**

- JavaScript
- CSS
- React
- Bootstrap

#### **Tools:**

- Git
- Visual Studio Code

#### **Platform:**

- Local development server

## 3. Software Requirement Specifications

### 3.1 Product Scope

This System is designed to provide user online music streaming.

### 3.2 System Functional Requirements

#### **R.1 Manage User**

##### R.1.1 Login

Description: User logs in to the system by entering valid user-id and password.

Input: User-id and Password.

Output: Home page is displayed if user successfully logs in.

Processing: User-id and Password validation.

#### **R.2 Manage Songs**

##### R.2.1 Search Songs

Description: Users can search for any song.

Input: Select "Search" option

Output: User prompted to enter key words

Processing: Search the songs based on the keywords entered.

##### R.2.2 Add Product

Description: Admin can add any song.

Input: "Add song" option

Output: Confirmation Message (i.e., Song successfully added)

### R.2.3 Song Description

Description: Users gets the description of a song.

Input: "Select" option

Output: Detailed description of the song (i.e. Artist name and album name)

### R.2.4 Play/Pause Song

Description: User can play/pause song anytime.

Input: Click on the Play/Pause button

Output: Song will start/stop playing.

### R.2.5 Repeat songs/play next

Description: User can use loop function-to play same songs/play next.

Input: Select loop icon on player

Output: Song repeat/plays next.

### R.2.6 Sort by Language

Description: User can listen to songs of his/her preferred language.

Input: Select language

Output: Songs of chosen language will be visible

## **R.3 Manage Playlist**

### R.3.1 Liked Songs

Description: User can add favourite song.

Input: Select Heart icon

Output: Add song to Liked playlist.

### R.3.2 Recently played

Description: User can directly get last played song.

Input: Recently played playlist

Output: Songs played recently.

### 3.3 Other Non-functional Requirements

#### 1. Performance

The system must be interactive and must not involve long delays. Though in case of opening the website components or loading the page the system shows the delays less than 2 seconds.

#### 2. Safety

The user's data is highly personal. The system has authorization to avoid any un- authorized access to user's private data.

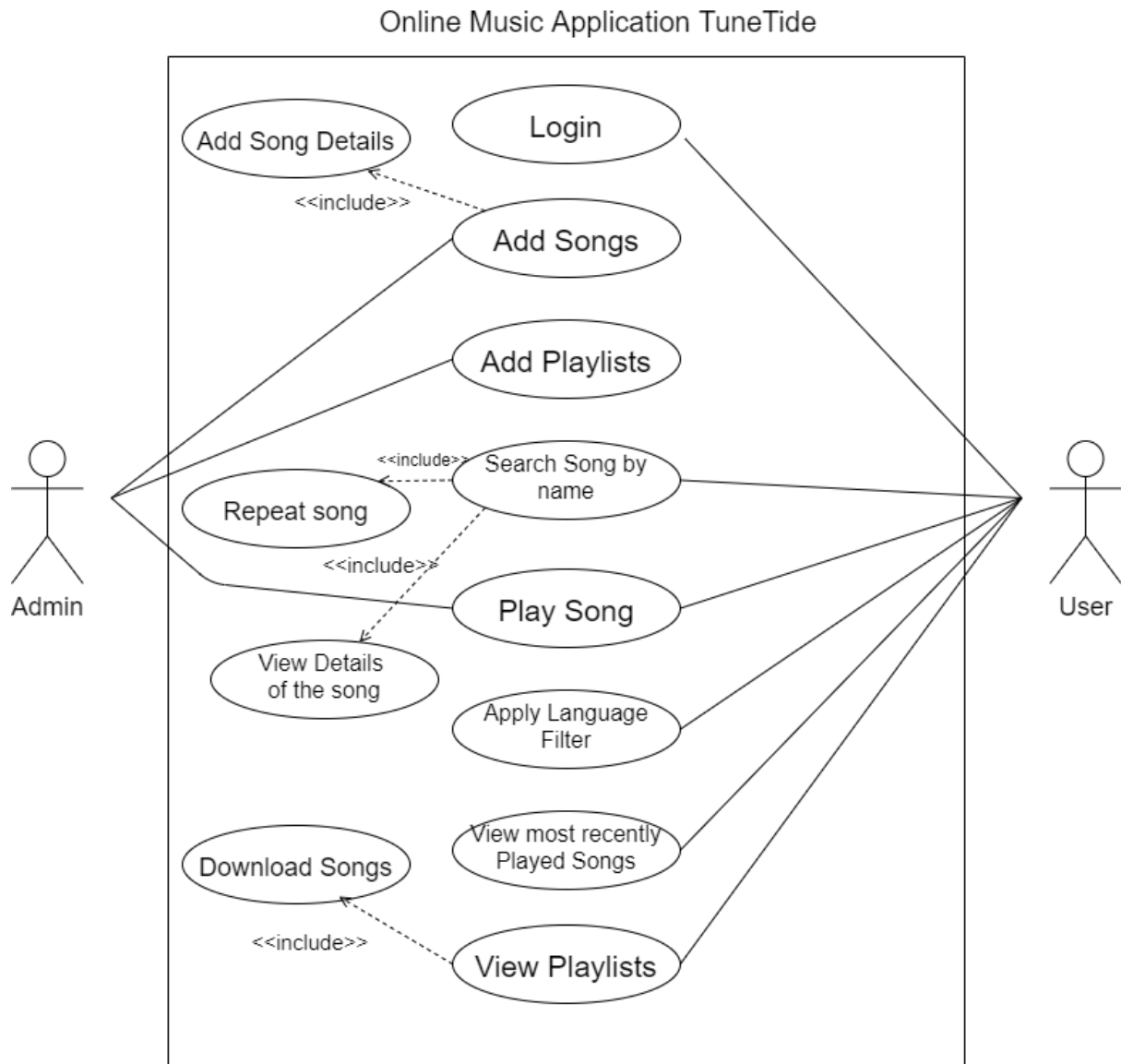
#### 3. Reliability

As the system has personal data, its reliability is the major factor for consideration.

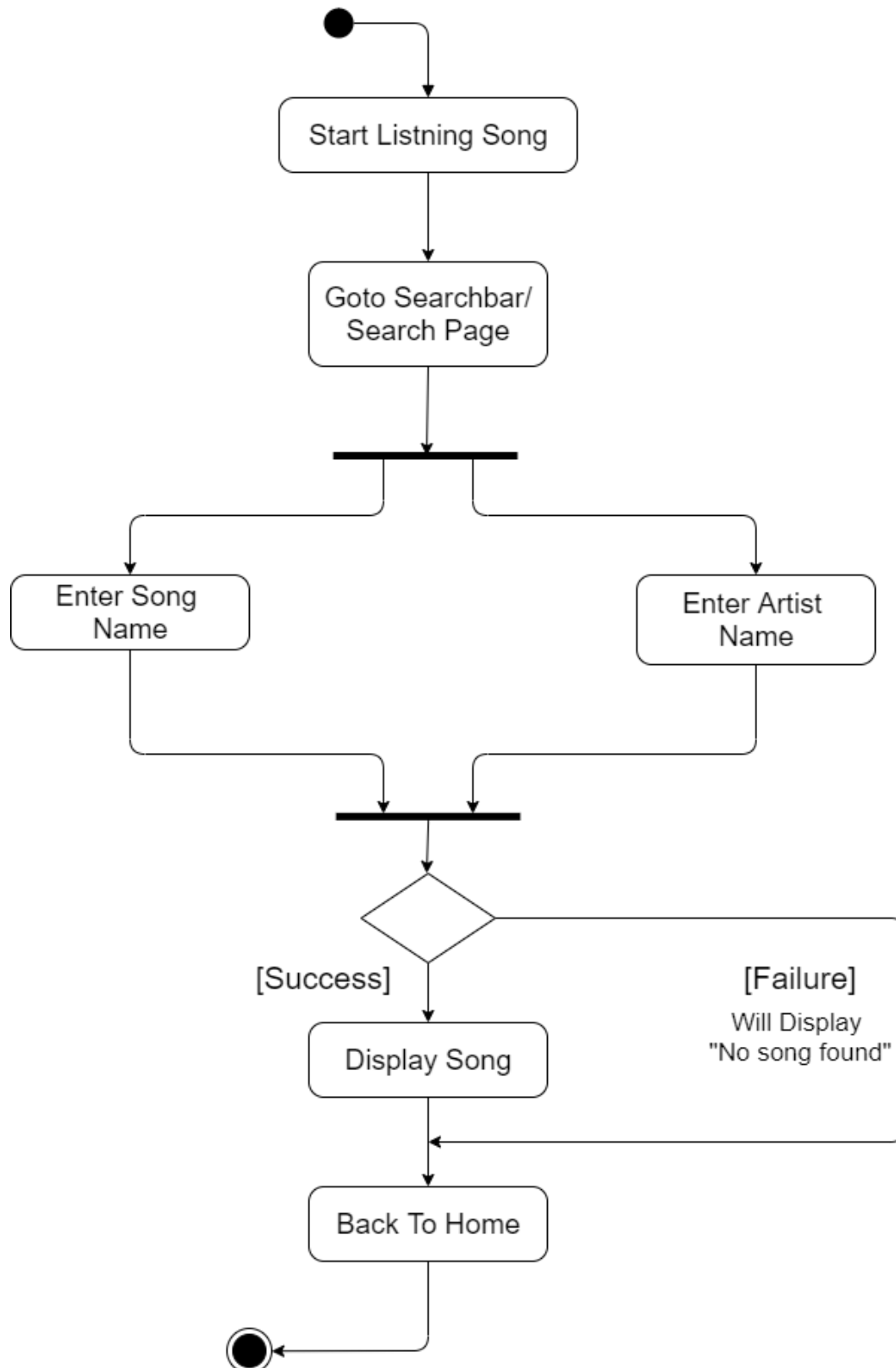


## 4. Design

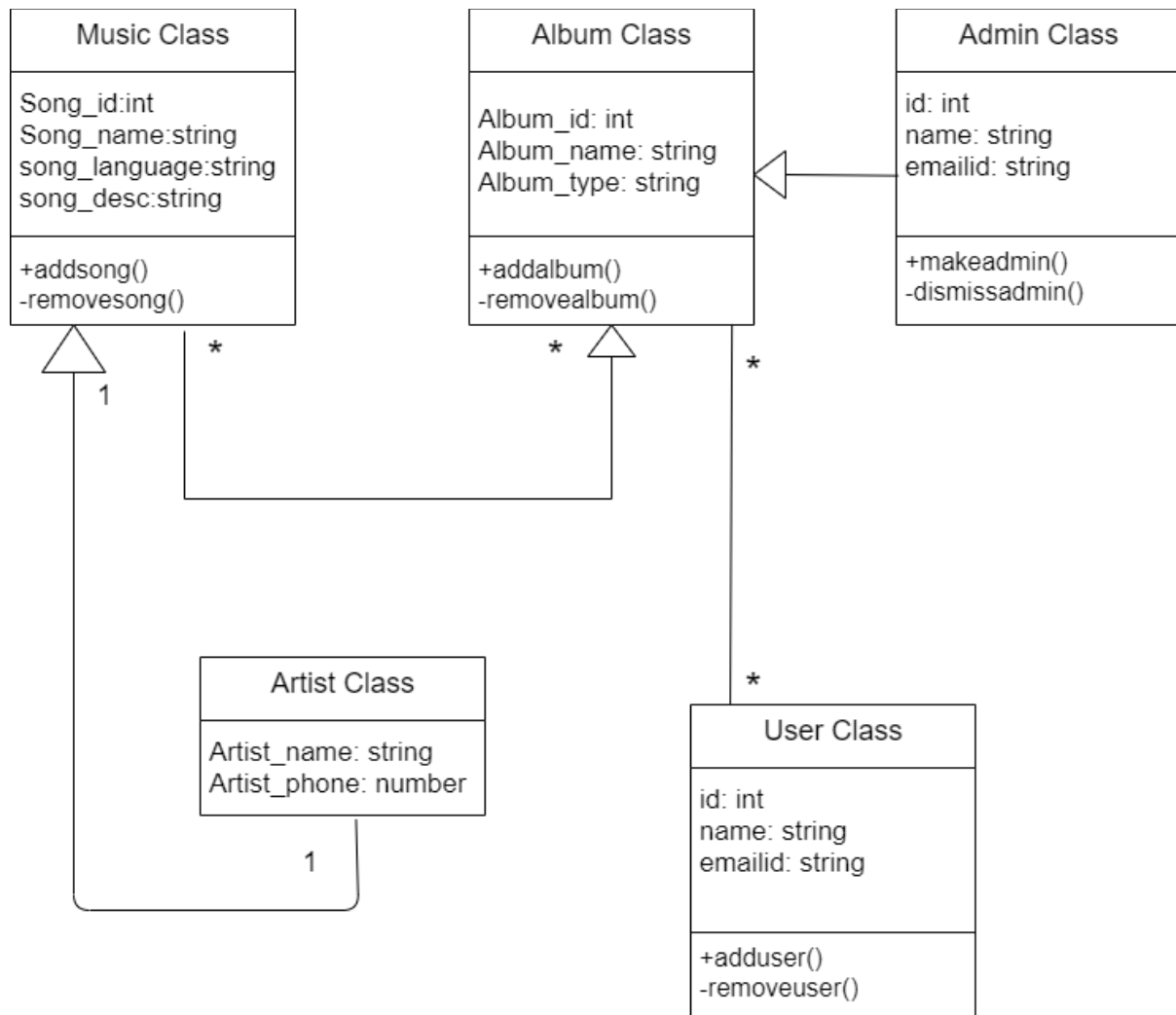
### 4.1 Use Class Diagram:



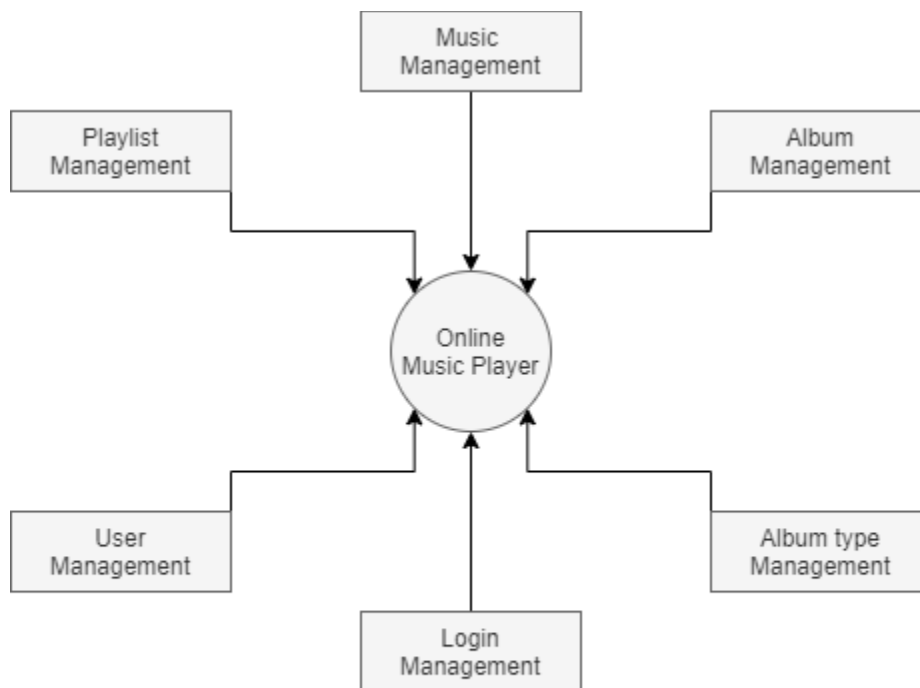
## 4.2 Activity Diagram:



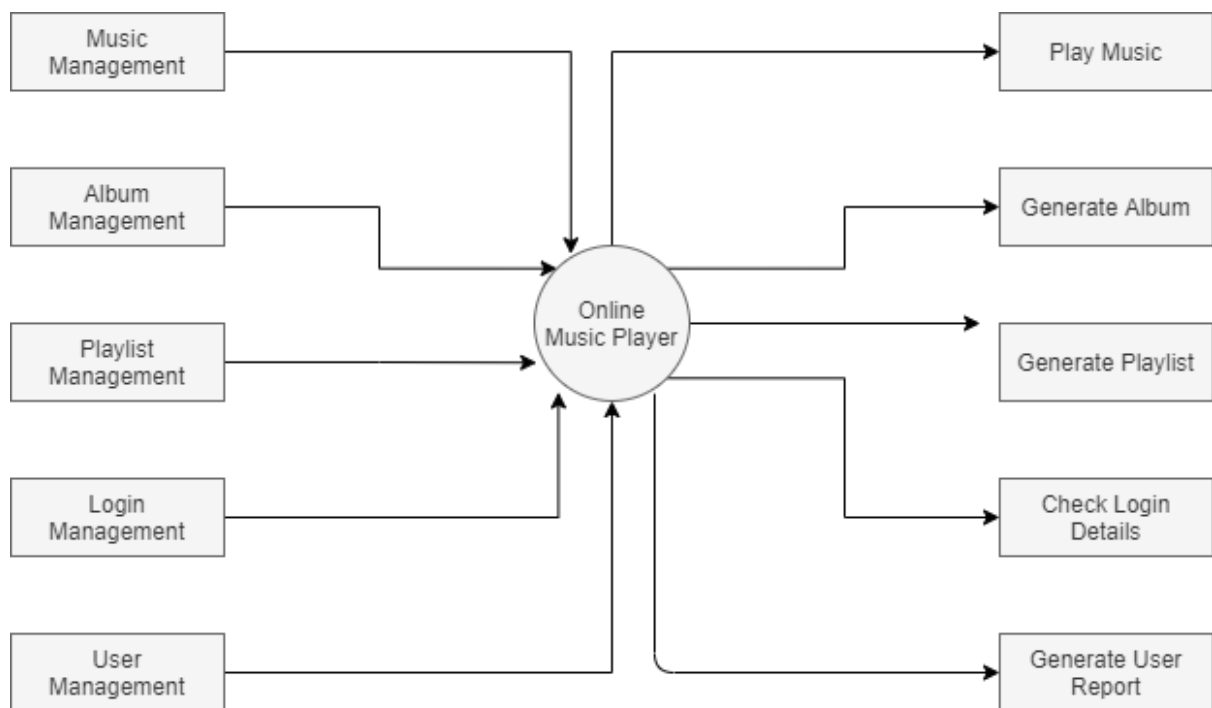
### 4.3 Class Diagram:



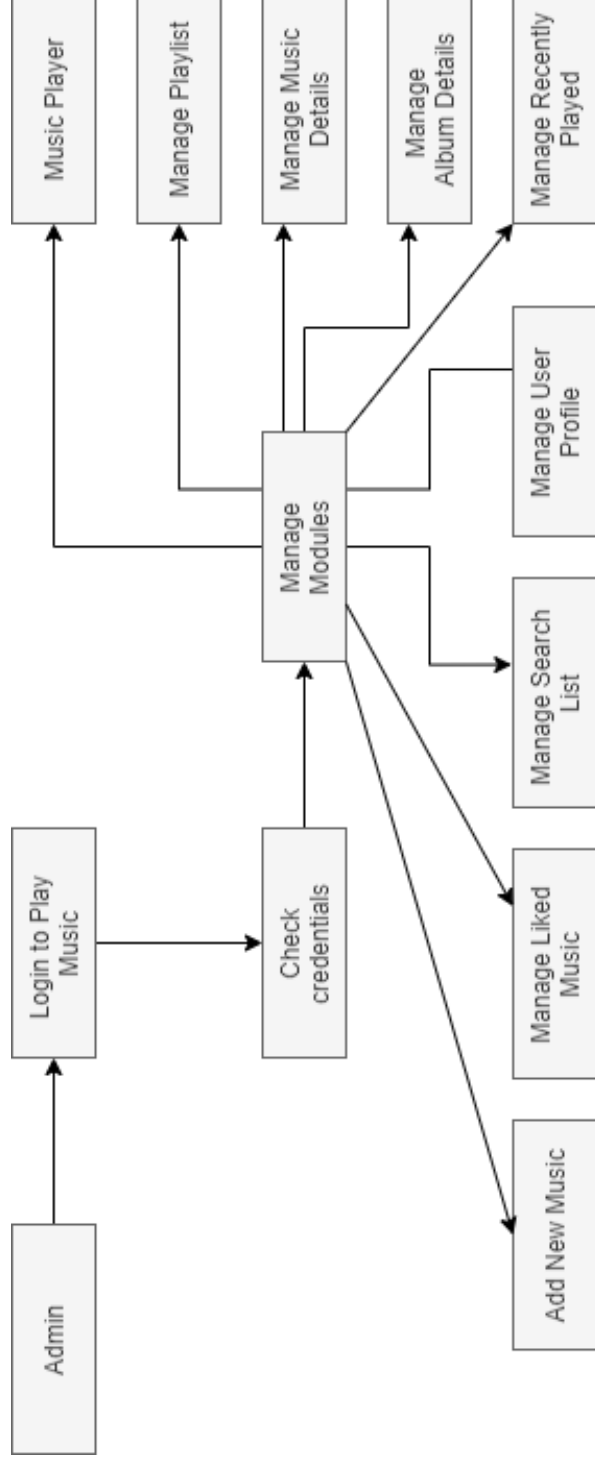
#### 4.4 DFD Model:



#### Zero-level

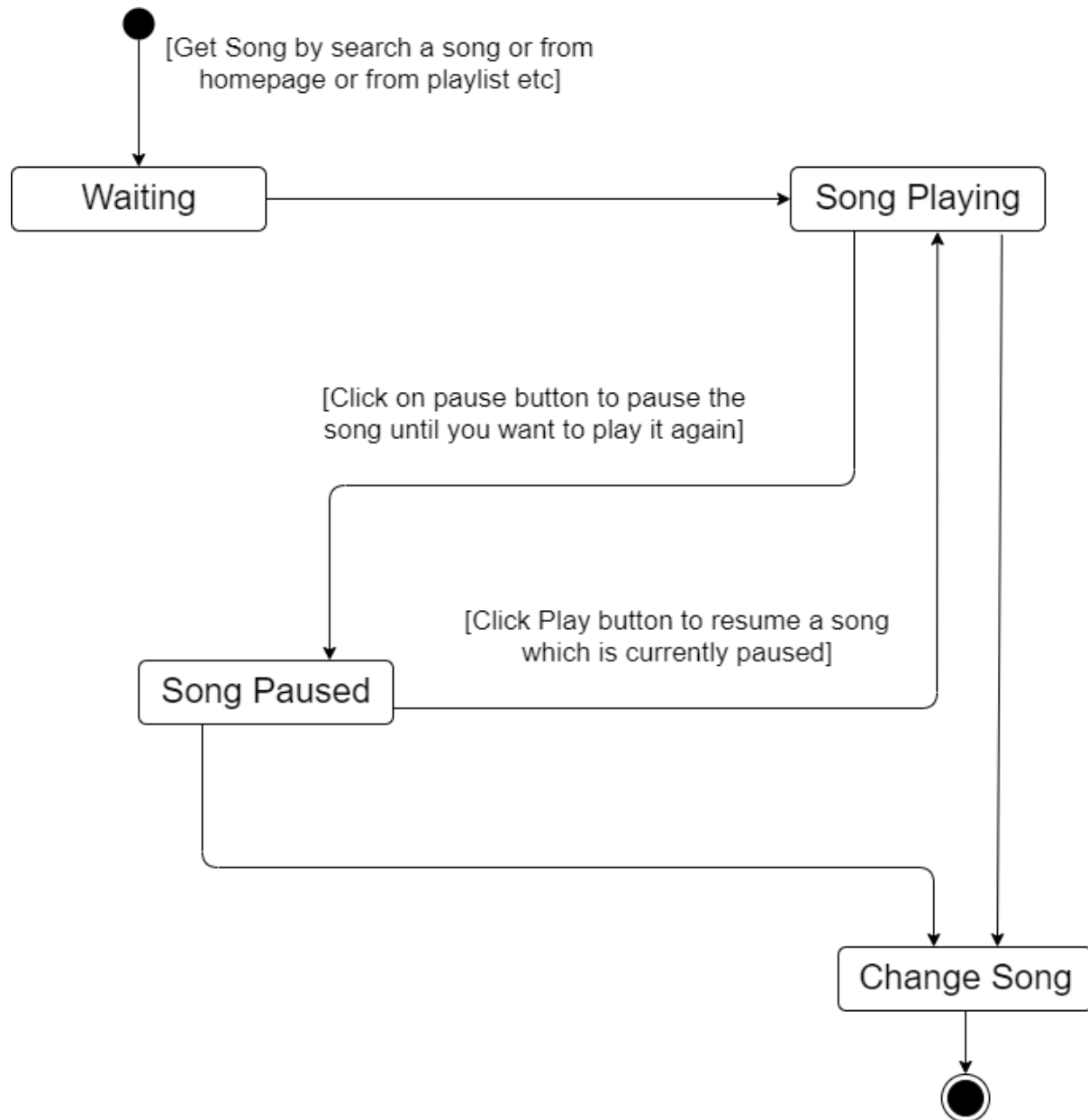


#### First-level



## Second Level

## 4.5 State Diagram:



## 5. Implementation Details

---

The system consists of 4 basic pages namely

1. Login Page
2. Home Page
3. Search Page
4. Profile Page

### 5.1 Login Page:

User will login using his/her credentials and will be redirected to home page to play songs.

### 5.2 Home Page:

User is just one click away from playing a song. Create your own playlist, add your favourites songs to it. Sort songs language preferred by you.

### 5.3 Search Page:

Search album, artist and song so, that you can enjoy it without wasting anytime.

### 5.4 Profile Page:

It contains details of the user such as name, phone number, email id.

## 5.5 Function prototypes:

### Login Page-

```
1  import React from "react";
2  import HeadPhone from '../assets/img/headphones.svg';
3  import './css/Login.scss';
4  import {Link} from "react-router-dom";
5
6  class Login extends React.Component{
7      render() {
8          return(
9              <section id="main">
10                 <div className="nav-item">
11                     <a className="navbar-brand" href="/">TuneTide</a>
12                 </div>
13                 <div className="main-row">
14                     <div className="main-row-img">
15                         <img className="head-phone-img" src={HeadPhone} alt=""/>
16                     </div>
17                     <div className="main-row-text">
18                         <h1>Music for everyone</h1>
19                         <p>Without music, life would be a mistake</p>
20                         <Link to={"/home"} className="btn">
21                             Start Listening
22                         </Link>
23                     </div>
24                 </div>
25             </section>
26          );
27      }
28  }
29
30  export default Login;
```



## Home Page-

```
40 function Home() {
41
42
43     const [screenSize, setScreenSize] = useState(undefined);
44     const [currMusic, setCurrMusic] = useState(null);
45     const [Page, setCurrPage] = useState(<MusicCardContainer/>);
46
47     let pathname = window.location.pathname;
48     useEffect(() => {
49         |   setCurrPage(getCurrPage(pathname))
50     }, [pathname]);
51
52     window.addEventListener("resize", handleResize);
53
54     function handleResize() {
55         |   setScreenSize(window.innerWidth);
56     }
57
58     useEffect(() => {
59         |   handleResize();
60         |   return () => window.removeEventListener("resize", handleResize);
61     });
62
63     const useStyle = useContext(ThemeContext);
64     const {playing, bannerOpen} = useSelector(state => state.musicReducer);
65
66
67     useEffect(() => {
68         |   setCurrMusic(playing)
69     }, [playing])
70
71     const [loaded, setLoaded] = useState(false);
72     useEffect(() => {
73         |   setLoaded(true)
74     }, []);
```

## Search Page-

```
11 const Search = () => {
12   const {playlists, search} = useSelector(state => state.musicReducer);
13   const [searchResult, setSearchResult] = useState([]);
14   useEffect(() => {
15     setSearchResult(playlists.filter((i) => (
16       (i.name.toLowerCase().startsWith(search))
17       ||
18       (i.author_name.toLowerCase().startsWith(search))
19       ||
20       (i.musicName.toLowerCase().startsWith(search))
21       ||
22       (i.lang && i.lang.toLowerCase().startsWith(search))
23     )));
24   }, [search, playlists]);
25   return [
26     <Container>
27       {
28         (search === "" || search === null)
29         ?
30         <div className={"Search"}>
31           <div className="Search-img">
32             <img className={"Rotate-img"} src={SearchMusicDisc} alt="search-music-icon"/>
33             <img src={SearchMusicMp3} alt="search-music-icon"/>
34             <img src={SearchMusic} alt="search-music-icon"/>
35             <img className={"Arrow"} src={ArrowUp} alt=""/>
36           </div>
37         </div>
38         :
39         <div className={"Search-result"}>
40           {
41             searchResult.length === 0
42             ?
43             <div className={"Search-fallback"}>
44               No result found.
45             </div>
46             :
```

## Profile Page-

```
function Profile() {

  const {playlists} = useSelector(state => state.musicReducer);
  const [mostPlayed, setMostPlayed] = useState([]);

  function sortByProperty(property) {
    return function (a, b) {
      if (a[property] > b[property])
        return 1;
      else if (a[property] < b[property])
        return -1;
      return 0;
    }
  }

  useEffect(() => {
    setMostPlayed(playlists.sort(sortByProperty("timesPlayed")));
  }, [playlists]);

  useEffect(() => {
    Grade(document.querySelectorAll('.gradient-wrap'))
  });

  return (
    <Container>
      <div className={"Profile"}>
        <div className="top-profile">
          <Avatar variant={"rounded"} src={require("../assets/img/avatar2.jpg").default}
            style={{width: "150px", height: "150px"}}>
            VS
          </Avatar>
        </div>
      </div>
    </Container>
  )
}
```

## Music Player:

```
function FooterMusicPlayer({music}) {

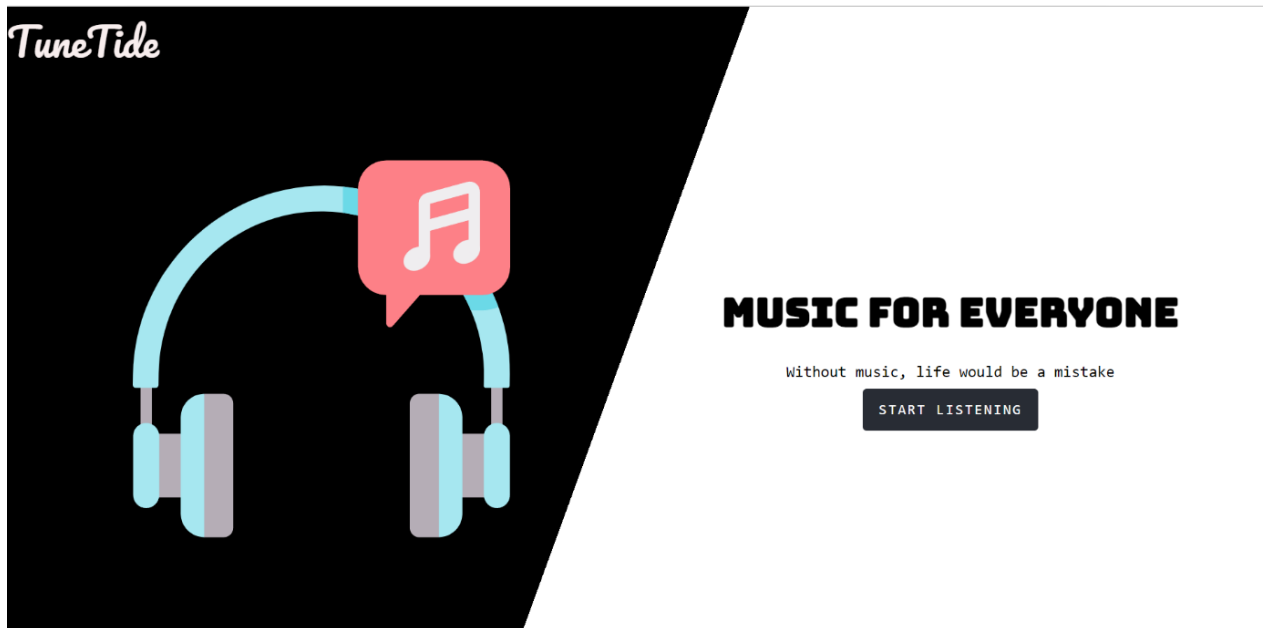
  const [{id, name, author_name, img, musicName}, setCurrTrack] = useState(music);
  const [isRepeatClicked, setRepeatClick] = useState(false);
  const [isPrevClicked, setPrevClicked] = useState(false);
  const [isNextClicked, setNextClicked] = useState(false);
  const [isPlaying, setPlayPauseClicked] = useState(false);
  const [isVolumeClicked, setVolumeClicked] = useState(false);
  const [volume, setVolume] = useState(50);
  const [seekTime, setSeekTime] = useState(0);
  const [duration, setDuration] = useState(0);
  const [currTime, setCurrTime] = useState(0);
  const [bannerToggle, setBannerToggle] = useState(false);

  const audioElement = useRef();
  const dispatch = useDispatch();
  const {playlists} = useSelector(state => state.musicReducer);
  const useStyle = useContext(ThemeContext);
  const pointer = { cursor: "pointer", color: useStyle.theme };

  const handleToggle = (type, val) => {
    switch (type) {
      case "repeat":
        setRepeatClick(val);
        break;
      case "prev":
        setPrevClicked(val);
        break;
      case "play-pause":
        setPlayPauseClicked(val);
        break;
      case "next":
        setNextClicked(val);
        break;
      case "volume":
        setVolumeClicked(val);
        break;
    }
  }
}
```

## 6. Screenshots

---



TuneTide

Search

Music Languages ^ (AT) ^

LIBRARY

Home

About

Search

MY PLAYLIST

Latest Songs

Favourite

All Time

Shakedown

Clarx

Games Worldb...

Bernardo R.

OverMyHead

ROY KNOX

Want You (fe...

Subtact, Sar...

Apocalypse

Midranger

Alive

Neoni, N3WPO...

Impact Moder...

Kevin MacLeo...

Impact Moder...

Kevin MacLeo...

Ignite

Tom wilson, ...

Shakedown

Clarx

⏮

⏪

⏸

⏩

⏭


00:04/02:45

🔊

Search

Music Languages ^ (AT) ^

ABOUT US



VIDIT SHAH - CE003

BHAVIN CHAVDA -CE023

COMPUTER ENGINEERING STUDENTS AT DHARMSINH DESAI UNIVERSITY, NADIAD.

ADVANCE TECHNOLOGY SEMESTER PROJECT

GUIDED BY: PROF.PRASHANT M.JADAV

VIDIT SHAH

📘

🌐

📷

BHAVIN CHAVDA

📘

🌐

📷

MUSIC

MUSIC IS A MEANS OF GIVING FORM TO OUR INNER FEELINGS WITHOUT ATTACHING THEM TO EVENTS OR OBJECTS IN THE WORLD.

THERE IS MUSIC IN THE AIR, MUSIC ALL AROUND US; THE WORLD IS FULL OF IT,AND YOU CAN SIMPLY TAKEAS MUCH AS YOU REQUIRE.

Song: Clarx - Shakedown [NCS Release]

Music by: NoCopyrightSounds

Download:<http://ncs.io/Shakedown>

Search

Music Languages ^

AT



# Advance Technology

## MOST PLAYED



**Games Worldb...**  
Bernardo R.



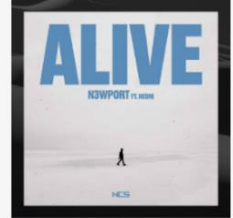
**OverMyHead**  
ROY KNOX



**Want You (fe...**  
Subtact, Sar...

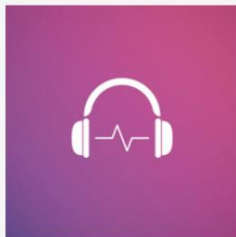


**Apocalypse**  
Midranger



**Alive**  
Neoni, N3WPO...

## YOUR LATEST-SONGS PLAYLIST



**Games Worldb...**  
Bernardo R.



**Want You (fe...**  
Subtact, Sar...



**Shakedown**  
Clarx

## 7. Conclusion

---

The functionalities are implemented in system after understanding all the system modules according to the requirements. Functionalities that are successfully implemented in the system are:

- Login
- User validation
- Logout
- Add Songs (by Admin)
- Remove Songs (by Admin)
- Play/Pause Songs (by User)
- Rewind/Next Song (by User)
- Add/Remove Songs to Playlist (by User)

After the implementation and coding of system, comprehensive testing was performed on the system to determine the errors and possible flaws in the system.



## 8. Limitation and Future Enhancements

---

We are able to implement some of the functionality of all modules. We aim to complete all the functionality of all modules and make this product ready to be used practically in all scenarios. Currently, the project runs completely fine if all the inputs / selections are given within proper criteria but it doesn't cover all the corner cases.

The project can be extended to run robust. The project supports online in-browser view.

In future, we will try to cover up features such as multiple users can login, users can hear to podcast and can browse recently played songs.

## 9. Reference / Bibliography

---

Following links and websites were referred during the development of this project:

[youtube.com](https://www.youtube.com)

[getbootstrap.com](https://getbootstrap.com)

[reactjs.org](https://reactjs.org)