



Tilak Maharashtra Vidyapeeth, Pune
Department of Computer Science

A PROJECT REPORT

ON

Music streaming from a web-server to an Android app and
Website.

By

Imran Shaikh

Shriraj Pethe

Towards The Partial Fulfillment of the
Master of Computer Application

[LFP-0217]
[Mumbai]

Tilak Maharashtra Vidyapeeth, Pune
Department of Computer Science

[2019-2020]



Tilak Maharashtra Vidyapeeth, Pune
Department of Computer Science

CERTIFICATE

This is to certify that the project,
“*Music streaming from a web-server to an Android app and Website.*”
Has been satisfactorily completed by

Imran Shaikh

PRN: 04518003013

Shriraj Pethe

PRN: 04518003015

Towards The Partial Fulfillment of the ‘Master of Computer Application’,
For the Academic Year [2019-2020] at [LFP-0217] Center,
[Mumbai],
Tilak Maharashtra Vidyapeeth, Pune (Department of Computer Science),
And it is approved.

Project Guide

Examiner

Head of Department
[LFP – Mumbai]



Tilak Maharashtra Vidyapeeth, Pune

Department of Computer Science

ACKNOWLEDGEMENT

With immense pleasure we are presenting “Music streaming from a web-server to an Android app and Website” Project report as part of the curriculum of ‘Master of Computer Application’. We wish to thank all the people who gave us unending support.

We express our profound thanks to our Project Guide Mrs. “Rupali Shinde” and all those who have indirectly guided and helped us in preparation of this project.

Imran Shaikh

PRN: 04518003013

Shriraj Pethe

PRN: 04518003015

PROJECT SYNOPSIS

Title

Music streaming from a web-server to an Android app and Website.

Group Members

Imran Shaikh

Shriraj Pethe

Front End

XML, Java for Android

HTML, CSS, JS, AJAX for Web

Back End

XAMPP Stack

PHP

AJAX

Database – MySQL, MariaDB on UNIX socket

Software requirements

<u>Android (For App):</u>	<u>For Web version (For Website):</u>
Android API 25 (Android 7.1) or better. Will support ≈66.5% device. Preferably API 26+ (Android 8+)	Any HTML5 supported modern Web browser, Chromium based browser preferred Preferably Desktop browser.

Hardware requirements

<u>Android (For App):</u>	<u>For Web version (For Website):</u>
3 GB of ram or more. 1 GB of free space. Preferred SOC: MediaTek Helio P22 MT6762 6 or more cores preferred. Communication module(s): WLAN OR LTE / Cellular Data.	4 GB of ram or more. 6 GB of free space. Dual-Core CPU (Intel 3 rd generation or above). Communication module(s): WLAN OR LAN.

PROJECT SUMMARY

This project report is meant to serve as a guide for this project. In this project we have created a server to host our music files, and to connect our android app along with our website with the database. This book comprehensively covers all major aspects of our project.

ABSTRACT

In this project, we have implemented software to stream and play music files stored on our webserver to an online Android Device or to our website.

We attempt to use various methods to link server and the device. And implement various background and foreground services in Android to provide a seamless and user-friendly experience for music playback. We have also made all the features available on our website.

We have tried to provide the same features on both our platform. We have made user interface as easy as possible. And have provided users ability login and register their accounts.

Digital music streaming websites have taken over the musical landscape. While the digital music market is booming, both data and time have revealed that in the current system as it exists users prefer free service over the paid option(s). In our project we try to host for free and to provide the users with legally freely available music. We analysed and reviewed currently available free or freemium music streaming services which are popular. Including but not limited to Spotify, Apple Music, Tidal, YouTube music, Gaana.com. Currently, Spotify is the most popular streaming service. However, this may be due to the fact that it offers a freemium service alongside a paid service. Apple Music was launched on the world's most popular smartphone, the iPhone, as well as other Apple products. Gaana.com is from India and has more focus on Indian market.

Music streaming from a web-server to an Android app and Website.

Because of our goal to provide Software As A Service for music streaming, as affordable as possible, we were restricted to using legally free music. And we tried to obtain Free Web hosting.

This project is a website as well as an android app, through which after creating a user account user can login from n number of devices simultaneously and listen to our music offering. We differentiate and allow user to browse the music offering by Albums, Artists, and Genres. We also provide search functionality, and ability to create, delete playlists. On our android app we also provide a queue in which music files are displayed that are currently being played or have been played, this list is synchronised for each user, which means a user can use single account on multiple android devices and can still listen to same music queue. We also provide auto timer feature through which user may select time (in minutes), after the time runs out the music playback stops.

<u>INDEX</u>		
<u>CHAPTER</u>	<u>NAME</u>	<u>PAGE RANGE</u>
<u>1</u>	<u>INTRODUCTION</u>	<u>8 - 16</u>
	Music Streaming an Overview	<u>11</u>
	Time line of significant events	<u>16</u>
<u>2</u>	<u>SURVEY OF EXISTING SYSTEMS</u>	<u>17 - 29</u>
	Overview	<u>17</u>
	Important on-demand music streaming service providers	<u>17</u>
	Other Types	<u>26</u>
	Several market leaders	<u>26</u>
	Statistics	<u>28</u>
<u>3</u>	<u>PROPOSED SYSTEM</u>	<u>30 - 35</u>
	OUR PROJECT'S GOALS	<u>30</u>
	Market overview	<u>30</u>
	How are we going to build a Music Streaming Service	<u>31</u>
	Factors	<u>31</u>
	Scope of proposed system	<u>35</u>
<u>4</u>	<u>FEASIBILITY ANALYSIS</u>	<u>36 - 42</u>
	FEASIBILITY STUDY	<u>36</u>
	UNDERSTANDING FEASIBILITY STUDIES	<u>36</u>
	Goals of feasibility studies	<u>36</u>
	The Importance of Feasibility Studies	<u>37</u>
	Tools for Conducting a Feasibility Study	<u>37</u>
	Types of Feasibility	<u>37</u>

<u>INDEX</u>		
<u>CHAPTER</u>	<u>NAME</u>	<u>PAGE RANGE</u>
5	<u>SOFTWARE DEVELOPMENT LIFE CYCLE</u>	43 – 64
	SDLC, an overview	43
	SDLC Models	43
	Stages of SDLC	43
	Some SDLC models	45
	<u>PERT & GANTT chart</u>	60
	PERT Chart	60
	GANTT chart	61
	Our project's GANTT CHART	62
6	<u>REQUIREMENT ANALYSIS</u>	65 - 82
	Overview	65
	Features and Cost	65
	About the used TECHNOLOGY	72

<u>INDEX</u>		
<u>CHAPTER</u>	<u>NAME</u>	<u>PAGE RANGE</u>
7	<u>SYSTEM DESIGN AND ANALYSIS</u>	83 - 112
	Overview	83
	2 approaches	83
	Tools for System analysis and design	84
	UML, an Overview	84
	List of few UML Diagrams	85
	UML diagrams shown as categories	85
	A. Use case diagram	86
	B. Sequence diagram	87
	C. Database Tables diagram	100
	D. Class diagram	101
	E. State diagram	112
8	<u>IMPLEMENTATION</u>	113 - 122
	Overview	113
	Modules specification	113
	How we implemented each module	113
9	<u>TESTING</u>	123 - 130
	Overview	123
	Testing Objectives	123
	Software Testing Types	123
	Challenges and Solutions found as result of testing phase	126
	Actual Performance and Resource requirement by our app	126

<u>INDEX</u>		
<u>CHAPTER</u>	<u>NAME</u>	<u>PAGE RANGE</u>
<u>10</u>	<u>SCREENSHOTS</u>	<u>131 - 145</u>
<u>11</u>	<u>CONCLUSION</u>	<u>146 - 146</u>
	Limitations	<u>146</u>
	Application and Future Scope	<u>146</u>
	Conclusion	<u>146</u>
<u>12</u>	<u>REFERENCE AND BIBLIOGRAPHY</u>	<u>147 - 149</u>

CHAPTER 1

INTRODUCTION

Streaming media is multimedia that is constantly received by and presented to an end-user while being delivered by a provider. The verb "to stream" in this context means the process of delivering or obtaining media, however streaming is different than traditional file downloads, where the file is downloaded from provider completely before it could be read or accessed partially or completely. Streaming refers to the delivery method of the medium, rather than the medium itself. When streaming the end user may be receiving the pieces of the file and is also able to access or read the pieces that they are receiving. The user doesn't have to wait till the file is completely received.

Streaming and Download are two different methods of delivery of content. Some methods of telecommunication can be considered to be inherently of type streaming, for example: Radio, Television, and Streaming apps / services. While some are inherently of type non-streaming, for example: Books, Video CDs, and Audio cassettes. Streaming (over internet and otherwise) has types: Live and Non-Live also known as Video-On-Demand or Music-On-Demand. On demand streaming is technically streaming but is not of a live content.

Major General George Owen Squier, in 1922 was granted patents for a system for the transmission and distribution of signals over electrical lines. He created Wired Radio, a service which piped music to businesses and subscribers over wires. In 1934, he changed the service's name to 'Muzak'. Muzak was a technology streaming continuous music to commercial customers without the use of radio. Attempts to display media on computers date back to the earliest days of computing in the mid-20th century. However, little progress was made for several decades, primarily due to the high cost and limited capabilities of computer hardware. From the late 1980s through the 1990s, consumer-grade personal computers became powerful enough to broadcast and display various media. The primary technical issues related to streaming were having enough CPU power bus bandwidth to support the required data rates, creating low-latency interrupt paths in the operating system to prevent buffer underrun, and enabling skip-free streaming of the content. However, computer networks were also yet limited in the mid-1990s, and audio and video media were usually

Music streaming from a web-server to an Android app and Website.

delivered over non-streaming channels, such as by downloading a digital file from a remote server and then saving it to a local drive on the end user's computer or storing it as a digital file and playing it back from CD-ROMs.

Kalpana introduced the first commercial Ethernet switch in 1990, this enabled the more powerful computer networks that lead to the first streaming video solutions. In the mid-1990s the World Wide Web was established, but streaming audio would not be practical until years later.

During the late 1990s and early 2000s, users had increased access to computer networks, especially the Internet. During the early 2000s, users had access to increased network bandwidth, especially in the "last mile". These technological improvements facilitated the streaming of audio and video content to computer users in their homes and workplaces. There was also an increasing use of standard protocols and formats, such as TCP/IP, HTTP, and HTML as the Internet became increasingly commercialized, which led to an infusion of investment into the sector.

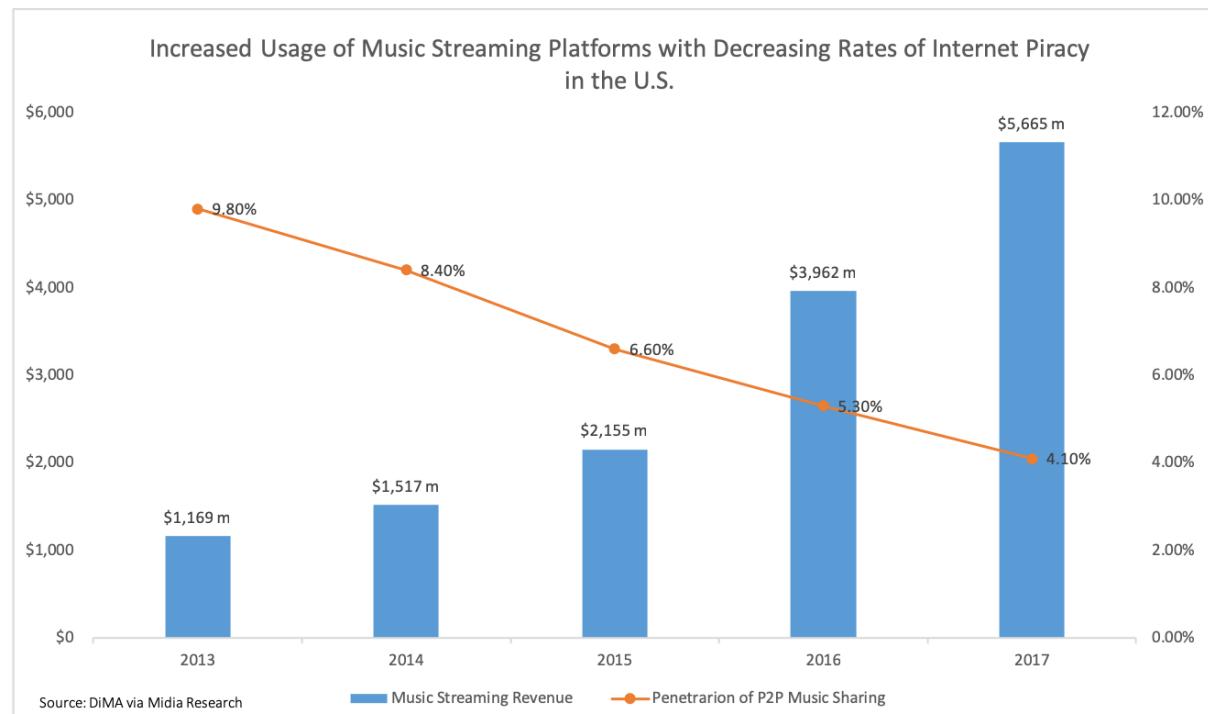
The band Severe Tire Damage was the first group to perform live on the Internet. On June 24, 1993, the band was playing at Xerox PARC while elsewhere in the building, scientists were discussing new technology (the Mbone) for broadcasting on the Internet using multicasting. As proof of PARC's technology, the band's performance was broadcast and could be seen live in Australia and elsewhere.

The evolution of computer and related technologies in between 1980s and 2000s made the streaming possible.

'Rise and fall' of Napster. In early 1999 Napster was launched out of Hull, Massachusetts. The platform was developed by Shawn and John Fanning as well as Sean Parker. This was peer to peer file sharing network where users could upload and download MP3 files freely, broke all music industry conventions. Napster was making possible for its users to share .mp3 files with each other. And users also started sharing mp3 files for which they had paid for, and thus breaking the law. The lawsuit A&M Records, Inc. v. Napster, Inc. fundamentally changed the way consumers interact with music streaming. After the service was shut down by court order. Similar Peer to Peer soft wares and services began to take form. Along with them, music industry also wanted to create something which will allow it to sell their music pieces for profit. This gave rise to various online Music stores. Napster also had a method for its users to Stream or Download mp3 files.

Music streaming from a web-server to an Android app and Website.

The music industry's response to music streaming (legal or pirated) was initially negative. Along with music piracy, streaming services disrupted the market and contributed to the fall in revenue from \$14.6 billion in revenue in 1999 to \$6.3 billion in 2009 for the U.S. CD's and single-track downloads were not selling because content was freely available on the Internet. The result of this was, that producers started preferring and prioritising popular artists, because only those on Music Chart would be able to bring enough revenue. Which meant the unique bands or artists were no longer a viable option. However, in 2018, music streaming revenue exceeded that of traditional revenue streams (e.g. record sales, album sales, downloads). 2017 alone saw a 41.1% increase in streaming revenue alone and an 8.1% increase in overall revenue. Streaming revenue is now one of the largest driving forces behind the growth in the music industry. But as we can clearly observe, it required 8 ~ 18 years since the fall of revenue till the period where Streaming services would bring the expected revenue (or better).



As music streaming platforms have become more prevalent in the U.S., music piracy rates have fallen. Piracy rates are calculated as a function of U.S. total population. This data was sourced from the Digital Media Association's (DiMA) annual report from March 2018.

Music streaming from a web-server to an Android app and Website.

For streaming multiple file formats, protocols, and compression methods are used. The audio stream is compressed to make the file size smaller using an audio coding format such as MP3, Vorbis, AAC or Opus. In the 2010s, technologies such as Apple's HLS, Microsoft's Smooth Streaming, Adobe's HDS and non-proprietary formats such as MPEG-DASH have emerged to enable adaptive bitrate streaming over HTTP as an alternative to using proprietary transport protocols. The quality of the interaction between servers and users is based on the workload of the streaming service; as more users attempt to access a service, the more quality is affected unless there is enough bandwidth or the host is using enough proxy networks. Deploying clusters of streaming servers is one such method where there are regional servers spread across the network, managed by a singular, central server containing copies of all the media files as well as the IP addresses of the regional servers. This is observed with YouTube. There are many methods and strategies to Stream music or videos, some are better optimised for speed, some for load sharing.

For this project the context of media transfer, download, and Streaming is related to that of music files and over the internet. And we are using a single, free, shared host service provider, which publicly hosts our music files, and provides a way to connect to our database with use of PHP.

How Does Music Streaming Work? An Overview

All of these music streaming services work in a very simple way: the streaming service delivers data to a client devices / browser / application in small amounts so that the user can get pre-buffered music that has been pre-buffered a few minutes or even seconds before playing a song.

In this scenario it is assumed or required that user and service providers have good internet connection, as we have written above, some service providers use multiple geo location based servers connected to central server, and they all keep copies of the data across all of the servers. While some service providers may choose to keep single server.

In our project, because we have used a free hosting server, the connection of the server itself is likely to be slower and restricted. And hence we would be unable to guarantee smaller delay time, or better or comparable performance.

Music streaming from a web-server to an Android app and Website.

Given that the user has a good internet connection, streaming technology provides an uninterrupted listening experience, without ever actually saving the files on the user's device.

This technology and the streaming services overall developed as a response to the music piracy platforms of the late 90's and early 2000's.

By 2008, by IFPI estimations, 95% of all digital music was downloaded illegally via piracy. The streaming services appeared as a solution to the fact that the market was in desperate need of a change, in need of a business model that could compete with the piracy services.

The open music model is an economic and technological framework for the recording industry based on research conducted at the Massachusetts Institute of Technology. It predicts that the playback of pre-recorded music will be regarded as a service rather than as individually sold products, and that the only system for the digital distribution of music that will be viable against piracy is a subscription-based system supporting file sharing and free of digital rights management. The research also indicated that US\$9 per month for unlimited use would be the market clearing price at that time, but recommended \$5 per month as the long-term optimal price.

Since its creation in 2002, a number of its principles have been adopted throughout the recording industry, and it has been cited as the basis for the business model of many music subscription services.

The model asserts that there are five necessary requirements for a viable commercial music digital distribution network:

#	Requirement	Description
1	Open file sharing	users must be free to share files with each other
2	Open file formats	content must be distributed in open formats with no DRM restrictions

3	Open membership	copyright holders must be able to freely register to receive payment
4	Open payment	payment should be accepted via multiple means, not a closed system
5	Open competition	multiple such systems must exist which can interoperate, not a designed monopoly

The model was proposed by Shuman Ghosemajumder in his 2002 research paper Advanced Peer-Based Technology Business Models at the MIT Sloan School of Management. The following year, it was publicly referred to as the Open Music Model.

The model suggests changing the way consumers interact with the digital property market: rather than being seen as a good to be purchased from online vendor, music would be treated as a service being provided by the industry, with firms based on the model serving as intermediaries between the music industry and its consumers. The model proposed giving consumers unlimited access to music for the price of US\$5 per month (as of 2002), based on research showing that this could be a long-term optimal price, expected to bring in a total revenue of over US\$3 billion per year.

The research demonstrated the demand for third-party file sharing programs. Insofar as the interest for a particular piece of digital property is high, and the risk of acquiring the good via illegitimate means is low, people will naturally flock towards third-party services such as Napster and Morpheus (more recently, BitTorrent and The Pirate Bay).

The research showed that consumers would use file sharing services not primarily due to cost but because of convenience, indicating that services which provided access to the most music would be the most successful.

The model predicted the failure of online music distribution systems based on digital rights management.

Criticisms of the model included that it would not eliminate the issue of piracy. Others countered that it was in fact the most viable solution to piracy, since piracy was "inevitable". Supporters argued that it offered a superior alternative to the current law-enforcement based methods used by the recording industry. One start-up in Germany,

Playment, announced plans to adapt the entire model to a commercial setting as the basis for its business model.

Several aspects of the model have been adopted by the recording industry and its partners over time:

- The abolition of digital rights management represented a major shift for the industry. In 2007, Steve Jobs, CEO of Apple, published a letter calling for an end to DRM in music. A few months later, Amazon.com launched a store single individual DRM-free mp3's. One year later, iTunes Store abolished DRM on most of its individual tracks.
- Open payment was relatively straightforward to implement, and the iTunes Store offered gift cards, which could be purchased with cash, from its launch in 2003.
- In 2010, Rhapsody announced a download ability for their subscribers using iPhones.
- In 2011, Apple launched its iTunes Match service with a subscription model, supporting file-sharing between a user's own devices. However, the subscription price did not include the cost of acquiring content, which would still have to be purchased on a per track basis from the iTunes Store.
- Pricing close to the model's suggested \$5 per month price, or its \$9 per month market clearing price, has been adopted by many platforms:
 - In 2005, Yahoo! Music was launched at \$5 per month with digital rights management.
 - In 2011, Spotify introduced a \$5 per month premium subscription in the United States with digital rights management.
 - In 2011, Microsoft Zune offered a subscription service for music downloads with digital rights management known as a Zune Pass, at \$10 a month.
 - In 2012, Google Play Music launched unlimited music streaming for a subscription price of \$9.99 per month. Users can upload their own MP3s to the service and download them, but cannot download songs they have not uploaded themselves.
 - In 2014, Amazon added DRM music streaming to their Amazon Prime service.
 - In 2015, Apple announced Apple Music, which would offer unlimited streaming of songs encrypted with FairPlay DRM for a subscription price of \$9.99 per month, and compensate artists on the basis of song popularity.

Music streaming from a web-server to an Android app and Website.

Apple reportedly wanted to enter the market with a lower price but was pressured by record labels to adopt a higher subscription fee.

Time line of significant events for Music Streaming Services

- 
- Napster was founded
1991
 - Launch of IUMA
1993
 - Birth of MP3 Audio format
1993
 - IndiaFM.com / Hungama Play launched
2000
 - The launch of iPod
2001
 - Introduction of Last.fm
2002
 - iTunes Store launch
2003
 - Pandora Internet Radio launched
2005
 - SoundCloud launched
2007
 - Saavn / JioSaavn launched
2007
 - Bandcamp formed
2007
 - Spotify introduced
2008
 - Gaana.com launched
2010
 - Wynk Music Launched
2014
 - Entry of Apple Music
2015

CHAPTER 2

SURVEY OF EXISTING SYSTEMS

Overview

There were, and are many On Demand Music Streaming services present over the internet. Many of these have presence in North American Markets, some are partially global or fully global. Some try to cater to specific audiences, categories, genres. Largely all are freemium, however only a few allow users access without login, regardless free or freemium or paid.

The Regional availability is a noticeable issue, because many of the studios which produce music have long standing contracts with partner concern, sister concern, and only these companies have rights to sell, stream these audio tracks or albums in their respective geo-location. For example: gaana.com is owned by Times internet, which is owned by Time Of India group, which also owns Times Music, and hence we can assume they'll have strategic advantages over rights and assets. Navigating all these contracts is tough. Hence many a times the services prefer not to expand globally, or expand in slow pace, or if they do they have to limit the availability of their music offering. These issues matter less, for and when the platform has independent artist with full ownership of their music uploading music by themselves. Some of these services may be limited by design on certain platforms. For example: Ovi music on Nokia.

Now we'll see a few important on-demand music streaming service providers (Some of these are or were Digital Music Stores)

Music streaming from a web-server to an Android app and Website.

A. Napster

Here we are looking into Napster as a music streaming provider, after the court order it was closed and sold off after bankruptcy, the brand is now owned by Rhapsody. And now is a music Streaming service provider

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
Napster	No	Yes	No	Yes	US, Canada, and most of Europe	Web, Android, iOS, Windows, Sonos, WebOS

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
Napster	~2002 (As Streaming Platform, Rhapsody)	XX	32	2.5	??	Yes

B. Last.FM

Last.fm is a music website founded in the United Kingdom in 2002. Using a music recommender system called "Audioscrobbler", Last.fm used to build a detailed profile of each user's musical taste by recording details of the tracks the user listens to, either from Internet radio stations, or the user's computer or many portable music devices. They merged with Audioscrobbler in 2005. As a streaming service provider it stopped its operations and links to music were replaced by links to YouTube and Spotify, where available.

Music streaming from a web-server to an Android app and Website.

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
Last.fm	Discontinued					

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
Last.fm	20 March 2002	28 April 2014	Discontinued			

C. iTunes Store / iTunes Radio / Apple Music

iTunes Store is a Digital store for apple products, specifically made for Music. It is a music player, media library, mobile device connection software (for apple products). Its Radio Broadcast services started as a music streaming service. It was available in US and Australian market. It was replaced completely by Apple Music on January 28, 2016.

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
Apple Music	No	Yes	No	Yes	Partial	Web, Android, iOS, Windows and Mac with iTunes, Sonos, AirPlay, Apple TV, Apple Watch, Apple HomePod, Amazon Echo

Music streaming from a web-server to an Android app and Website.

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
Apple Music	June 30, 2015	XX	60	60	60	Yes

D. Pandora

Pandora Radio is a popular streaming service. It gives recommendations to users based on their taste in music. It offers free with adverts options and paid options with better quality and offline access.

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
Pandora	Yes (radio mode)	Yes	No	Yes	US	Web, Android, iOS, Windows, macOS, BlackBerry, Bose, Chromecast, Android TV, PlayStation, Roku, Sonos, Amazon Echo, Google Home, TiVo, Xbox One Fitbit

Music streaming from a web-server to an Android app and Website.

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
Pandora	~2005	XX	30~40	78	5.48	No

E. SoundCloud

SoundCloud has grown to be one of the largest music streaming services reaching over 175 million monthly users worldwide. SoundCloud offers both free and paid membership on the platform, available for desktop and mobile devices. It allows anyone to share any kind of audio. SoundCloud has influenced the music industry through the success of many artists who have used the service to launch or advance their careers. SoundCloud has received support from many investors and other media platforms such as Twitter, although the streaming platform itself has dealt with funding issues and has laid off many employees in order to remain profitable.

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
SoundCloud	Yes	Yes	Partial	Limited	Global	Web, Android, iOS, Windows, macOS, Sonos

Music streaming from a web-server to an Android app and Website.

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
SoundCloud	August 2007	XX	200+	175	??	No

F. JioSaavn / Saavn

JioSaavn is the Indian online music streaming service and a digital distributor of Bollywood, English and other regional Indian music across the world. Since it was founded in 2007 as Saavn, the company has acquired rights to over 5 crore (50 million) music tracks in 15 languages. It is a freemium service; basic features are free with advertisements or limitations, while additional features, such as improved streaming quality and music downloads, are offered via paid subscriptions.

On 23 March 2018, Saavn announced merger with JioMusic in a deal worth more than \$1 billion. Saavn was founded in 2006 as BODVOD Network and initially distributed Bollywood content in North America in a B2B offering.

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
Saavn	Yes	Yes	No	Yes	Global	Web, Android, iOS, Windows, macOS, BlackBerry, Bose, Chromecast, Android TV, PlayStation ,

Music streaming from a web-server to an Android app and Website.

						Roku, Sonos, Amazon Echo, Google Home, TiVo, Xbox One Fitbit
--	--	--	--	--	--	--

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
Saavn	~2006 23 March 2018 as JioSaavn	XX	30	20	??	??

G. Spotify

Spotify is a Swedish Music Streaming service provider. Currently headquartered in Stockholm, Sweden. Was founded in 2006. The company mainly provides an audio streaming platform. Mainly provides music and podcast streaming service. It is a Freemium Service. For additional features like, advertisement and interruption free, offline listening are offered via paid subscription. Users can browse by parameters such as artist, album, or genre, and can create, edit, and share playlists. Spotify is available in most of Europe and the Americas, Australia, New Zealand, and parts of Africa and Asia, and on most modern devices, including Windows, macOS, and Linux computers, and iOS, and Android smartphones and tablets.

As of April 2020, the company had 286 million monthly active users, including 130 million paying subscribers.

Unlike physical or download sales, which pay artists a fixed price per song or album sold, Spotify pays royalties based on the number of artist streams as a proportion of total

Music streaming from a web-server to an Android app and Website.

songs streamed. It distributes approximately 70% of its total revenue to rights holders, who then pay artists based on their individual agreements.

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
Spotify	Yes	Yes	No	Yes	Partial	Web, Android, iOS, Windows, macOS, Linux, Bose, Palm, Sonos, Symbian, Android TV, Android Auto, CarPlay, Boxee, Telia Digital-TV, Virgin TiVo, Linux, PlayStation, Xbox One, BlackBerry, Chromecast, Amazon Echo, Google Home, Tesla, WebOS, Roku

Music streaming from a web-server to an Android app and Website.

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
Spotify	October 2008	XX	50	289	130	Yes

H. Gaana.com

Gaana.com was launched in April 2010. They formed a partnership in February 2013 with South Indian Music Companies Association to acquire rights to music from 79 different labels. Consequently, Gaana.com had access to more than 45+ million songs. Gaana generates an annual revenue of US\$5 million for Times Internet.

Name	Free access available	Selective streaming	Lossless (hi-fi)	Track downloads	Global availability	Other platforms
Gaana.com	Yes	Yes	Yes	Yes	Partial	Web, Android, iOS

Name	First launched	Discontinued	# of tracks (millions)	# of active users (millions)	# of paying users (millions)	Login required
Gaana.com	April 2010	XX	45+	150	??	No

Music streaming from a web-server to an Android app and Website.

Other Types

Online music storage services also exist. Also known as Cloud Music Services. These are internet services that allow uploads of personally owned or licensed music to the cloud for listening on multiple devices.

There were three large services: Amazon Music, Apple's iTunes Match, and Google Play Music. Each incorporating an online music store, with purchased songs from the associated music store not counting toward storage limits. Other than additional storage space, the main additional feature provided with an annual fee by Apple (and formerly Amazon.com) is "scan-and-match", which examines music files on a computer and adds a copy of matched tracks to the user's music locker without having to upload the files. Google provides both a large amount of storage space and the scan-and-match feature at no cost.

These largely are services which provide a Digital music marketplace and have access to cloud hosting services, and so they also provide user the capability to keep their personal audio files on the servers of services provider.

Amazon started phasing out cloud storage from December 2017.

Current major player:

- A. Google Play Music
- B. YouTube Music
- C. iBroadcast
- D. iTunes Match
- E. VOX Music Cloud

Let's discuss several market leaders further to understand how and why some popular service providers are this popular

Spotify

It's impossible to discuss the music streaming market without Spotify. This app is in line with the two streaming giants — Apple Music and YouTube Music. It's no secret that Spotify has been growing fast, offering an enormous selection of music at no cost. Free

Music streaming from a web-server to an Android app and Website.

accounts have some listening limitations and ads. The service offers premium accounts to get unlimited access to music.

Pandora

Pandora is also highly competitive in the world of streaming services due to the decent audio quality and great customization. Additionally, it has acquired the on-demand music service called Rdio and reclaimed its title of being among the best in the music streaming game.

Apple Music

Apple created apps for iOS and Android devices, offering various songs for users from all over the world. Additionally, users can listen to local radio stations using the application.

Just as we have analysed from above presented example. Some of these are either:

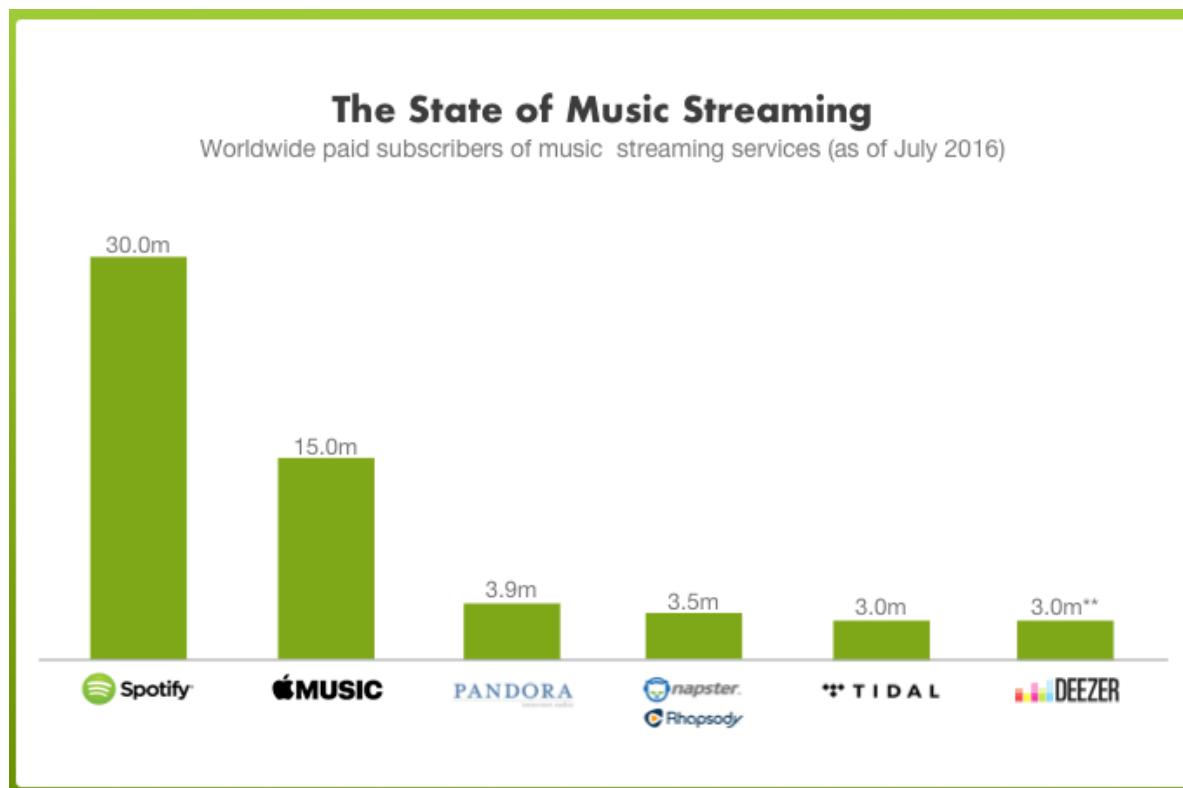
- Online music stores.
Or
- Online music storage services
Or
- Online On-demand music streaming service providers

Some of the services try to provide all these services in one.

Apart from these 3 categories, several services providers exist in a category called: ‘Online music databases’. They are largely free of charge. Many of the sites provide a specialized service or focus on a particular music genre. Some of these operate as an online music store or purchase referral service in some capacity. Among the sites that have information on the largest number of entities are those sites that focus on discographies of composing and performing artists. These may act as a music Community sites, or music focused social media.

Music streaming from a web-server to an Android app and Website.

Statistics

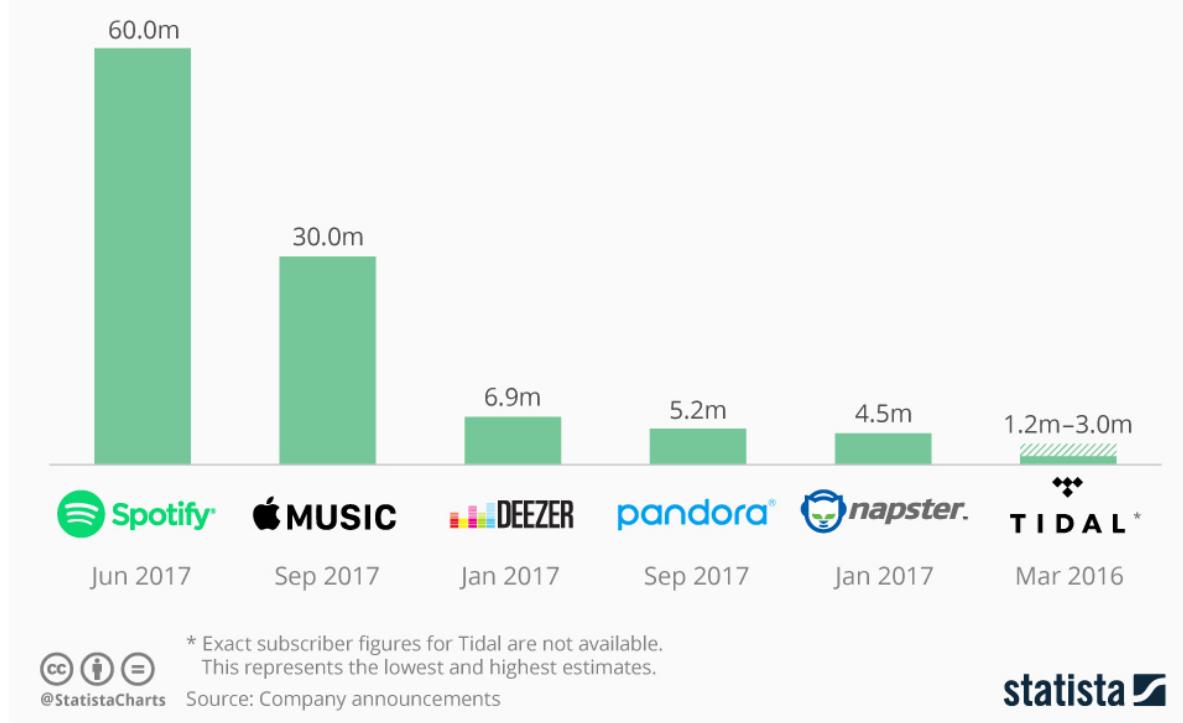


We can see the stark difference between 2016 (Above) and 2017 (Below).

DEEZER, Apple Music, Spotify has doubled their paid user base.

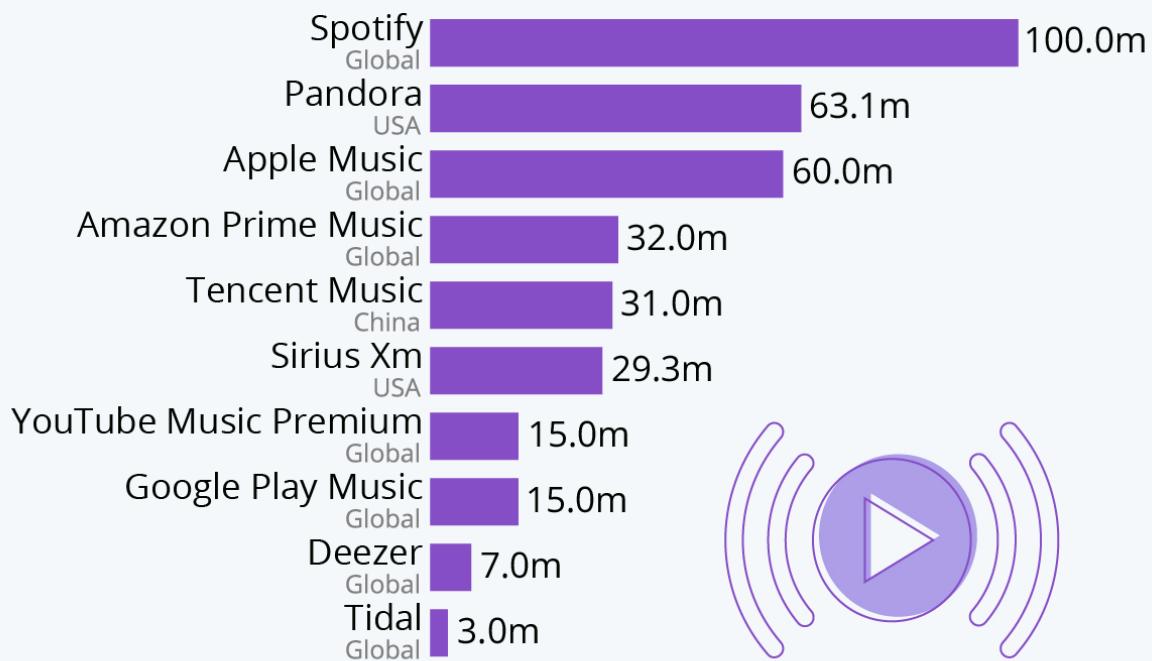
The Music Streaming Landscape

Worldwide paid subscriptions of music streaming services (lastest available figures)



The world's most popular music streaming services

Music streaming services with the most subscribers as of November 2019



Source: FIPP Global Digital Subscription Snapshot



statista

Popular music streaming apps



Spotify



Apple Music



Pandora

CHAPTER 3

PROPOSED SYSTEM

OUR PROJECT'S GOALS

In this project we have and will explore popular and / or important streaming services in necessary details. And now we will be attempting to replicate the technology in order to provide similar service to our website and android app.

The leading music streaming services like Spotify, Apple Music, gaana.com, JioSaavn, and Pandora have become that popular because they're simply convenient and offer good value for money. However, in order to create a music streaming app, we need to consider various factors like license, platform, technology, implementation, and others if any about these services.

Market overview of Music Streaming Services / App / Websites

We can observe, that music streaming services gain their popularity due to convenience they bring to users. Now let's see several statistics.

- The global music streaming market is expected to exceed \$10 billion by 2023.
- The average revenue per user is approximately \$26.
- Spotify has 36% market share of music streaming subscribers.
- Apple claims that 60 million users pay for Apple Music subscription.

As we can observe, the music streaming market is quite prospective. However, this means, it can be challenging to compete with services like Spotify or Pandora. Such platforms have spent over six years to acquire a user base and become who they're now.

What Is the Network Effect?

The network effect is a phenomenon whereby increased numbers of people or participants improve the value of a good(s) or service. The Internet is one of the best example of the network effect. Initially, there were only a few users on the Internet since it was of little value to anyone outside of the military and some research scientists. However, as more users gained access to the Internet, they produced more content, information, and services.

The development and improvement of websites attracted more users to connect and do business with each other. As the Internet experienced increases in traffic, it offered more value, leading to a network effect. In the network effect it is also seen that unless a competitive or alternative service or goods producer is offering significantly higher benefits users tend to not change their service / goods provider. Users or consumers prefer familiarity, and they seldom change their habits or service provider.

For example: Dvorak keyboard layout is considered better for speed and efficiency, but because the QWERTY keyboard layout was early comer, it gained more users, people started using and teaching it. As we can see just because a service, technology or product is better doesn't necessarily mean that it'll be able to replace something which is widely used.

The pre-existing and popular music streaming services have invested a lot of money in machine learning and artificial intelligence to make music streaming apps capable of suggesting relevant tracks, artists, albums and even genres. And they have really large user and fan base, of both paying and not paying users. They also have benefit of being early, as they have license, access, rights, and Digital rights management contracts with music producers.

Now we will take an overview of how are we going to build a Music Streaming Service, and what will be part of it

There are various things to consider before starting music streaming app development. Now we'll see four significant factors that we need to consider to build our project.

Factors

- A. Type of Music Streaming Services
 - a. Music library
 - b. Cloud storage
 - c. Radio stations
- B. Getting Licences
- C. Data Storage
- D. Platform

Music streaming from a web-server to an Android app and Website.

A. Type of Music Streaming Services

Let's discover common three types of music streaming services:

a. Music library

This type has a server-based music library and offers customers access to the music under their conditions. Either they have paid subscriptions or free access or freemium model.

Example: Apple Music, Spotify

We are making our project of this type.

b. Cloud storage

Allows users to store their own songs and stream them anywhere and anytime they wish to.

Example: AudioBox, Groove Music, Google Play Music

c. Radio stations

This kind of service provider has various radio stations and arranges them by specific themes, which can be streamed to PC, home audio, or any mobile device.

Example: Shoutcast, Apple Music

B. Getting the License

Music streaming services allow users to listen to music without purchasing. However, as the owner, we need to get a license for using music content.

Hence we need the Public Performance Rights of the music, audio, song pieces that we are using.

1. In the United States, it can be handled by one of two agencies:
 - a. The American Society of Composers, Authors and Publishers (ASCAP).
 - b. Broadcast Music Incorporated (BMI).
2. For Europe, apply to:
 - a. The Society of European Stage Authors and Composers (SESAC).
3. For India:
 - a. Phonographic Performance Limited India, (PPL India).
 - b. The Indian Performing Right Society Ltd (IPRS).
 - c. We can search on (<http://copyright.gov.in/SearchRoc.aspx>) website, to know the owner of any song or music composition.
 - i. <https://dipp.gov.in/>
 - Department for Promotion of Industry and Internal Trade.
 - ii. <https://www.wipo.int/portal/en/index.html>
 - WIPO is the global forum for intellectual property (IP) services, policy, information and cooperation. They are a self-funding agency of the United Nations, with 193 member states.
 - iii. <http://www.ipindia.nic.in/>
 - Website of Intellectual Property India.

Spotify or similar services get two types of licensing for streaming music

1. Sound Recording License agreements to cover the rights to a particular record.
2. Musical Composition License Agreements to cover the people who own the rights to the song.

Streaming services have to sign contracts with large music labels like Universal Music Group, Sony Music Entertainment Group, and others to stream music with copyrights belonging to these companies.

Music streaming from a web-server to an Android app and Website.

Skipping licensing and streaming music without permission is largely illegal, and can cost a fine from a minimum of \$750 up to \$150,000 per one song played.

For our project we will rely on music pieces which are available for free, that we can stream from our project legally.

C. Data Storage

Before we build a music streaming service, we should take care of data storage. We need a back-end server that supports streaming, if not at least hosting of our Sound tracks. For example, we can consider using hosting such services as AWS.

Another option is to stream from Spotify or SoundCloud catalogues itself using SDK and API that they provide. However, there are restrictions listed in Terms of Use for developer tools.

We create a data storage based app on a self-hosted server. In this case, we'll upload songs into a readable directory on our own server, which is managed by us.

For this project we have selected: 000webhost.com

- Launched in 2007.
- 000WebHost can be considered as a free learning platform where students can start their online journey, by testing their skills, for free.
- 000WebHost uses a very powerful infrastructure of Hostinger.com which is the sister web host of 000WebHost.
- Hostinger is one of the very popular names in the hosting industry.
- Services, soft wares and offerings same or similar to paid hosting services.

D. Platform

Currently, there are two popular platforms in mobile devices: iOS and Android. There is also desktop category that includes Windows, macOS, Linux. And then there is Web which is universally accessible for any device with capacity to connect to internet.

Which platform to choose to create our own music streaming service?

- Google Android and Apple iOS have 98% of the global market share for operating systems.
- Android's market share will reach 87% in 2022, forecasts suggest.
- Research into Android vs iOS market share in the US shows that the market leader remains Google Android, with a 51.1% market share in June 2019.
- In 2017, there were 2 billion monthly active Android devices around the world.
- In the USA, people prefer iOS devices. While the European market is full of Android devices. Same with many continents.

Considering this, for our project we have chosen android for its wider user base. Even, Apple Inc. created an Apple Music app for Android devices to gain more users.

Scope of proposed system

- Our project's minimal requirement is that it should be able to stream the music tracks from its server to web music player, and through our app.
- It should have all the major essential as well as some advanced features.
- Its design should be as clean and intuitive as we can make.
- The features on the Android device and Web front end should be as similar as we can make them.
- The user registration option should be available, in order to target and customise the user experience.

CHAPTER 4

FEASIBILITY ANALYSIS

FEASIBILITY STUDY

A feasibility study is an analysis which takes all of the project's relevant factors and parameters into account. For example: economic, technical, legal, operational, and scheduling considerations to ascertain the likelihood of completing the project successfully. Sometimes this also involves Socio-Political consideration if relevant. We need to use feasibility studies to discern the advantages and disadvantages of undertaking a project before they invest a lot of time and money into it.

Without the feasibility study we simply wouldn't know the risks involved, or will it possible under given circumstances? This report prevents us from making mistakes, or entering into risky venture blindly. The process gives us a report where the likely hood of success in all the mentioned factors is given, how will it be? And the possible challenges that us would face.

UNDERSTANDING FEASIBILITY STUDIES

In the feasibility study we assess the practicality of the proposed plan, or venture. We simply investigate if the project is feasible or not? And if yes or no, how and why? And is it feasible under given criteria? For example budget, team, tools, resources and so on. For any business or for any one pursuing any project, it is very rational to get higher Returns on Investment (ROI). So this is calculated and investigated through the Feasibility study.

Goals of feasibility studies

- To understand all the involved aspects of the project.
- To know early on about potential problems, challenges and difficulties that we may face during the project till its implementation
- To conclude after the study of all the factors, if whether the project is viable or not?
And / or does it fulfils the goals the way we want?
 - To know is our investment even worth the return or not.

The Importance of Feasibility Studies

Feasibility studies are important to business development. They allow a business to deal with where, how and the way it'll operate. They identify potential obstacles which will impede operations of any project or business and recognize the quantity of funding and other resources that will get the business or project up and running. Feasibility studies give us result by suggesting best marketing strategies that can be implemented, so it can help convince investors or banks, or to the upper management of the same company or the people doing the project that investing resources in this particular project or business may be a wise choice.

Tools for Conducting a Feasibility Study

Best Practices-

Feasibility studies reflect a project's unique goals and wishes, so each is different. However there are several suggested best practices which we may use:

- Prepare a projected earnings report
- Prepare a gap day record
- Write an organizational, operational, or a business plan
- Analyse and ask questions on our data to make sure that it's solid and reliable (We don't wish to find out later when in the middle of the progress that our base data itself wasn't accurate, because that'll disturb and affect everything else.)
- Get feedback about the new concept, project from the acceptable stakeholders
- Make an initial "go" or "no-go" decision about moving ahead with the plan
- Conduct a market survey or marketing research to reinforce data collection

Types of Feasibility

- A. Technical Feasibility
- B. Market Feasibility
- C. Schedule Feasibility
- D. Economic / Financial Feasibility
- E. Operational Feasibility
- F. Legal Feasibility
- G. Socio-Political Feasibility

A. Technical Feasibility

This part of the feasibility study is in the aspect of technical requirements of our project. Upon reviewing the tech stack of similar services and upon further research, we find that currently this project is technically feasible, with the skillset and expertise that we (the team developing this) happen to have.

What we studied:

- The availability of technology and their compatibility with each other (for example: Ajax, JavaScript and PHP)
- What is technology is currently used in the industry? And what can be used? And How?
- Various ways to implement the service
- Connecting back end to front end of Website and Android app
- Android Libraries which allow us, either ease in development or certain advantages, or both (for example: Picasso library, android-networking by Amit Shekhar for JSON parsing and Sliding Up Panel library for certain UI usage)

B. Market Feasibility

This part of the feasibility study is in the aspect of what, and how the situation of market is currently? And how other similar service providers operate in this market? We targeted the market of music streaming services, and other services which are closely related to study them. And we realised that currently if our project targets the same audience, the probability of survival (long term and short term) would be less than what would be beneficial. And hence if we rather only use legally free music tracks, then our costs get low, and we gain uniqueness. Further this is a project aimed at demonstrating the technology, and not necessarily made to run for profit.

What we studied:

- Competition or presence of competing services
- Description of the industry
- Current Market Analysis
- Anticipated future market potential
- Potential buyers and sources of revenues
- User demand (Demand From users)

C. Schedule Feasibility

This is feasibility study of how much time is required for components of project, from start to finish? Including Design, Development, Testing, Implementation and extra time has to be allotted for any unknown development or implementation related challenges that may arise.

We also studied how much time is allotted? What is the time limit?

Anyone, may it be a business or an employer or client will always prefer everything to be quick. But it'll be highly unprofessional if we predict one time frame and exceed it. It is always best idea to study the project well and predict the time frame required, and adjust it early on as per the requirements.

What we studied:

- We studied time required for various aspect of software development life cycle
- And How much time the individual components may require

Estimation or Prediction of time for development of various components and features:

Feature	Back-end + (web) front-end	Development	Android
Registration, Login	~50 hours		~65 hours

Music streaming from a web-server to an Android app and Website.

Search	~60 hours	~40 hours
Music player	~30 hours	~95 hours
My music / Playlist	~24 hours	~20 hours
Settings	~45 hours	~35 hours
Total time	200 ~ 220 hours	250 ~ 270 hours

D. Economic / Financial Feasibility

Any business or project that starts with goal of making profit in its mind, needs to have economic feasibility. This study gives us the probability of economic survival. Any business needs to plan at the start of the year, how and where they will spending? And how much? And how much profit, revenue and cash flow they are expecting? With all these parameters in the mind one can know if the project, venture is economically feasible or not? The company, business can decide whether to embark on that project or not? Or if to post pone it to next year?

Because this is a demonstrative project where we'll be investing least or no money, and we don't intend to make any profit, that makes this project economically feasible, Given we use Legal and free music And don't break any other licences or agreements.

The music tracks are legally available to us in no cost to us. The Tech stack that we intend to use is also available to us for free. All the IDEs for development purposes are free.

Some of these components are free with condition that we don't earn profit from it.

E. Operational Feasibility

Operational feasibility is the measure of how well a proposed system is able to solve the issues? And whether if able to take advantage of present opportunities or not, those which were identified in the project scope? And if so and how well is it able to satisfy the needs that were

Music streaming from a web-server to an Android app and Website.

identified in the requirement analysis? This study focuses on the how much the proposed project fits in with the existing business environment.

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. This gives us the probability or support of management and / or employees for the project. It is rather difficult to gauge.

For our project this wasn't a concern as all members were willing to do allotted work.

F. Legal Feasibility

This is a measure of whether the project is in conflict with legal requirements or not. For example: EU-GDPR is far different and strict compared to laws / regulations in North America, India, and Africa. So when operating under the jurisdiction of any authority we must comply fully with that country's laws and regulations.

Currently in our project we aren't keeping any sensitive data, and we are securely storing the password with MD5 hash.

G. Socio-Political Feasibility

Social feasibility is where we study the acceptance of the people in consideration with the product to be launched. This evaluates if users will be willing or not to accept our service, and how, and why? This gives us the idea if our service is going to be accepted with relative ease or not. For example: a certain software may have better feature than old ones, but if it requires hours' worth of training for the employees then the employer may not subscribe or buy or software or service.

Our project is not found to be difficult for users to adapt, considering the currently existing services have similar UI and features. Our project will by no means cause any alarms or questions in the society nor will it challenge any existing social conventions. Further, our project doesn't have any social or political agenda, neither negative nor positive, neither willingly or unwillingly, it's a simple music streaming service, which is streaming free and

Music streaming from a web-server to an Android app and Website.

legal music. For example: certain sites may be encouraging piracy of someone else's work, we don't do that in our project.

CHAPTER 5

SOFTWARE DEVELOPMENT LIFE CYCLE

SDLC = Software Development Life Cycle

About SDLC, an overview

SDLC also known as Software Development Process. SDLC is a process that includes a few series of pre-planned activities to develop or alter the Software Products. With use of SDLC one can get better potential from software development process. Software development industry has a few SDLC models, each with their benefits and disadvantages. Each have certain use cases. Using SDLC one can aim to produce highest quality of software. These models are designed so we can reach our goals or beyond in given or available time and costs.

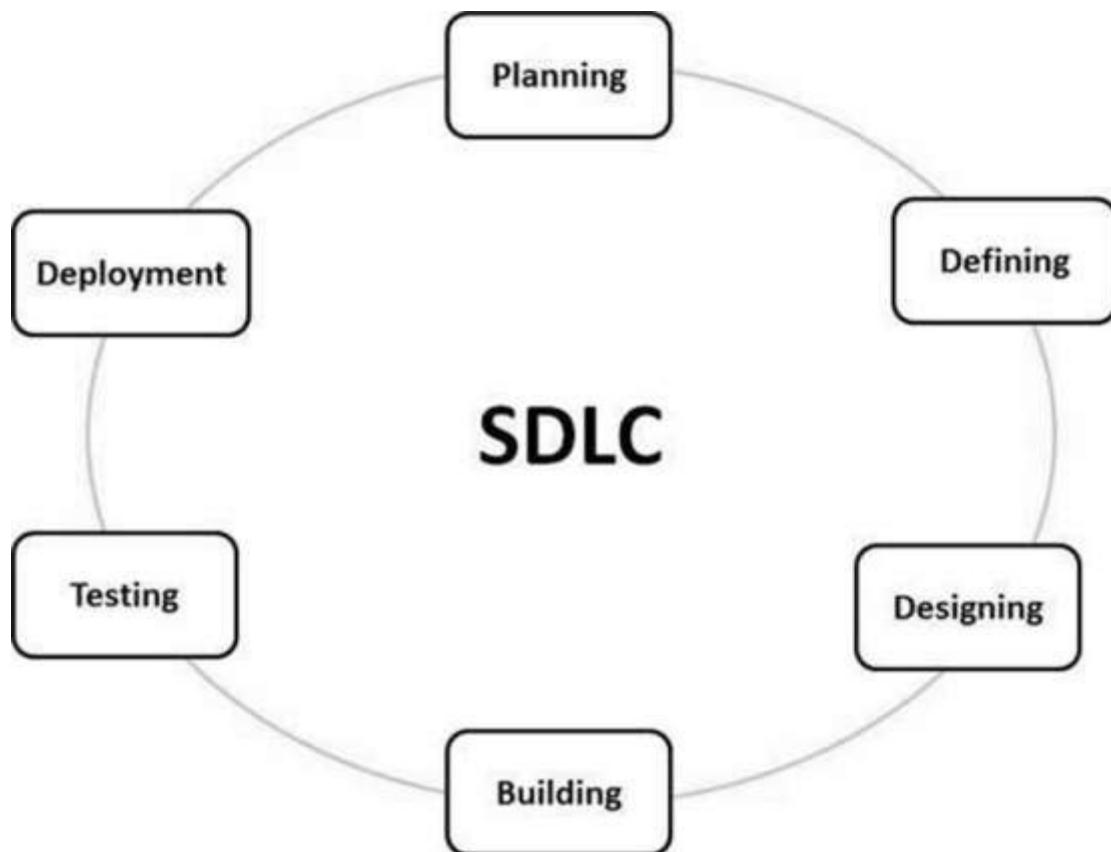
ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

SDLC Models

There are various SDLC models defined and designed which are followed during the process of development of a software project. Also known as SDP models. Each process model includes a Series of steps, and in different sequences as per the goal and purpose of that model. Depending on our need we must select the model best suited for us.

Stages of SDLC

- A. Planning and Requirement Analysis
- B. Defining Requirements
- C. Designing the Product Architecture
- D. Building or Developing the Product
- E. Testing the Product
- F. Deployment in the Market and Maintenance



A. Planning and Requirement Analysis

Most important and fundamental stage. We get to know what the goals and objectives of our software project is. We must perform feasibility study. We performed market survey to get relevant data. We planned for quality assurance and possible risks and challenges with this project.

B. Defining Requirements

After the requirement analysis we defined the requirements as clearly as possible. It is always helpful for the developer to understand the goals and needs of the project clearly. We defined everything that our software services has to do along with its features and operations. We used SRS – Software Requirement Specification document.

C. Designing the Product Architecture

Music streaming from a web-server to an Android app and Website.

From SRS, we made a DDS – Design Document Specification. We reviewed the DDS for major parameters, for example: budget, essential features, and time constraints. We defined architectural modules, communication and dataflow representation for our music streaming service.

The internal design of all the modules of the project architecture should always be clearly, and precisely defined with the minutest of the details in DDS.

D. Building or Developing the Product

As per the DDS, at this stage we shall start building our project. This is the most important stage where we basically make the whole project. Proper development, coding guidelines were to be followed, as they generally mean better performance.

E. Testing the Product

After the initial software product is ready it must be thoroughly tested, only then should we be able to know if there are any unknown bugs, problems, or if any requirement isn't met. If we find a problem, we shall fix it. We also requested several friends and colleagues to visit and try our site, and try to gain insight from their feedback.

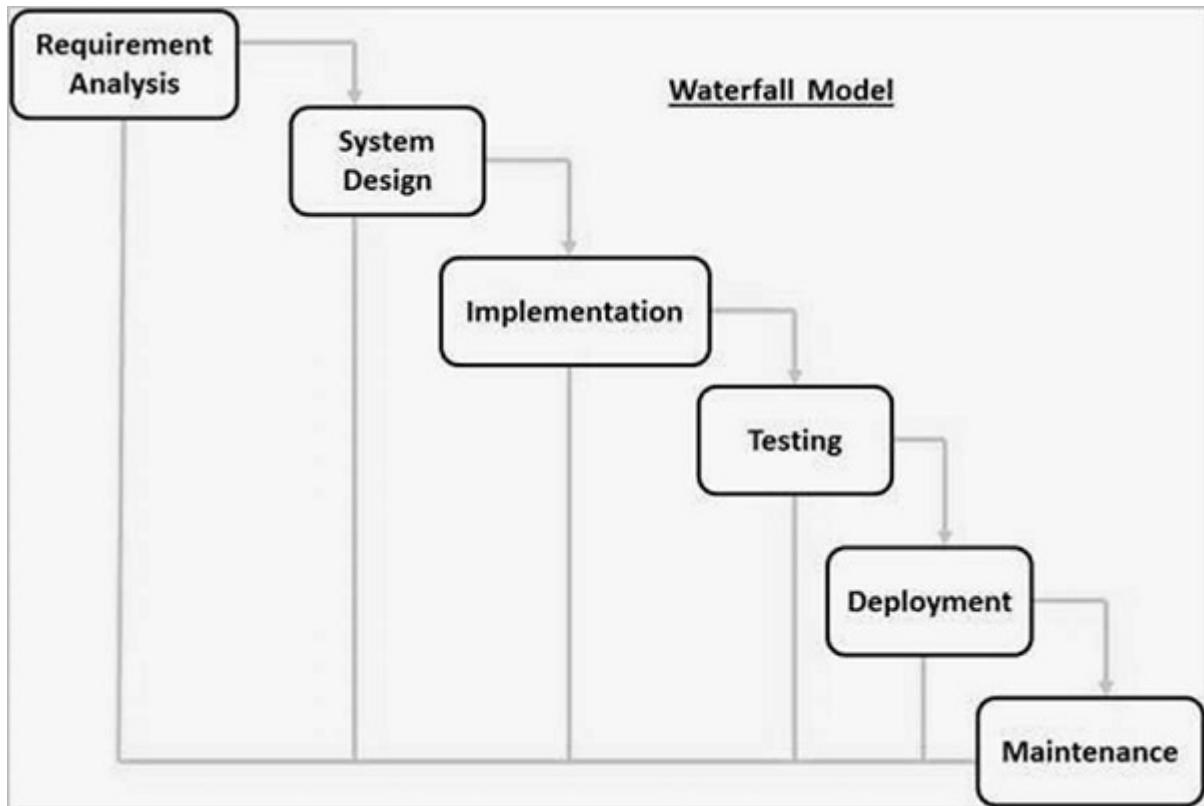
F. Deployment in the Market and Maintenance

Due to our software being web based, and on android platform, our front end was already deployed publically, although we didn't receive much real user traffic. After the initial feedback we made changes to our service. And after the Android app was completed we deployed the final version.

Some SDLC models

- A. Waterfall Model
- B. Iterative Model
- C. Spiral Model
- D. V-Model
- E. Big Bang Model
- F. Agile Model
- G. RAD Model
- H. Software Prototype Model

A. Waterfall Model



Also referred to as Linear Sequential model. In this model, as seen in the diagram, each step starts only after the end of previous. There has to be No overlap. This was the earliest model used in and by the industry. Outcome of previous stage will be the input for next stage.

In all of the SDLC models, most of them have 6 stages in common, as shown in the diagram above.

Applications:

Every software developed is different and requires an appropriate SDLC approach to be followed supported the interior and external factors.

- The project is small.
- Requirements are well documented, clear and fixed.
- Product definition is stable.

Music streaming from a web-server to an Android app and Website.

- Technology is known and isn't dynamic.
- There are not any ambiguous requirements.
- Ample resources with required expertise are available to support the product.

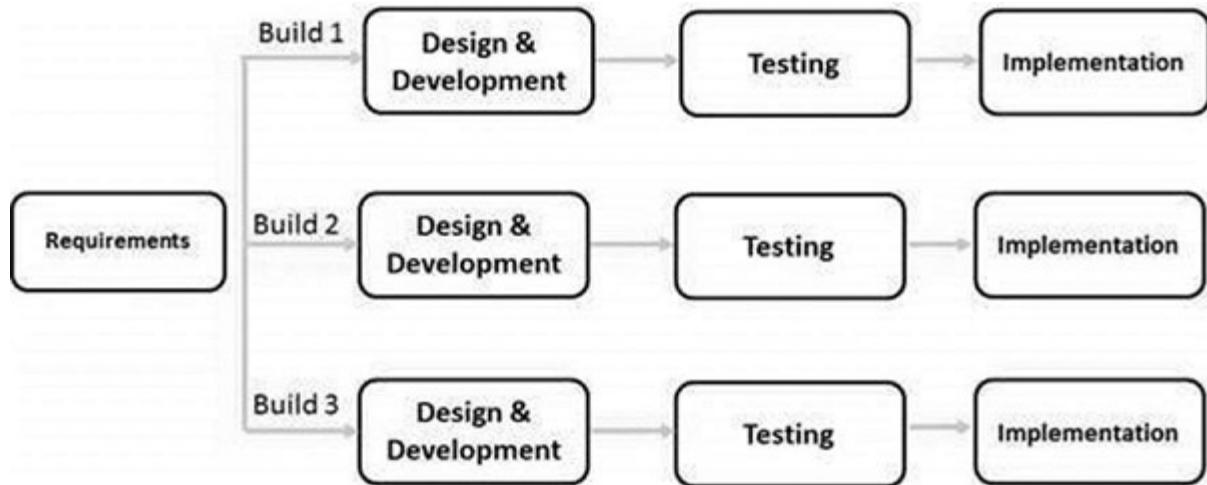
Advantage:

- Phases are processed and completed one at a time.
- Simple and straightforward to know and use
- Easy to manage thanks to the rigidity of the model.
- Easy to rearrange tasks.
- Works well for smaller projects where requirements are alright understood.
- Well understood milestones.
- Clearly defined stages.
- Process and results are well documented.

Disadvantages:

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Making changes and revisions is not possible, unless we want to restart everything.
- Not suitable model for long and ongoing projects.
- It is difficult to live progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.

B. Iterative Model



In this model we start with only a few requirements, we go through the whole process of Software development for each component that we are making. We don't need to have full requirement or specifications or goals. We start with what requirements we have.

In each iteration design modifications are made, and functional components are added.

Applications:

- Requirements of the entire system are clearly defined and understood. To be clear, some requirements may be not known, but as a whole the basic requirement of the whole project shall be known.
- There is a time to the market constraint.
- A new technology is getting used and is being learnt by the development team while working on the project.
- Resources with needed skill sets aren't available and are planned to be used on contract basis for specific iterations.
- There are some high-risk features and goals which can change in the future.

Advantages:

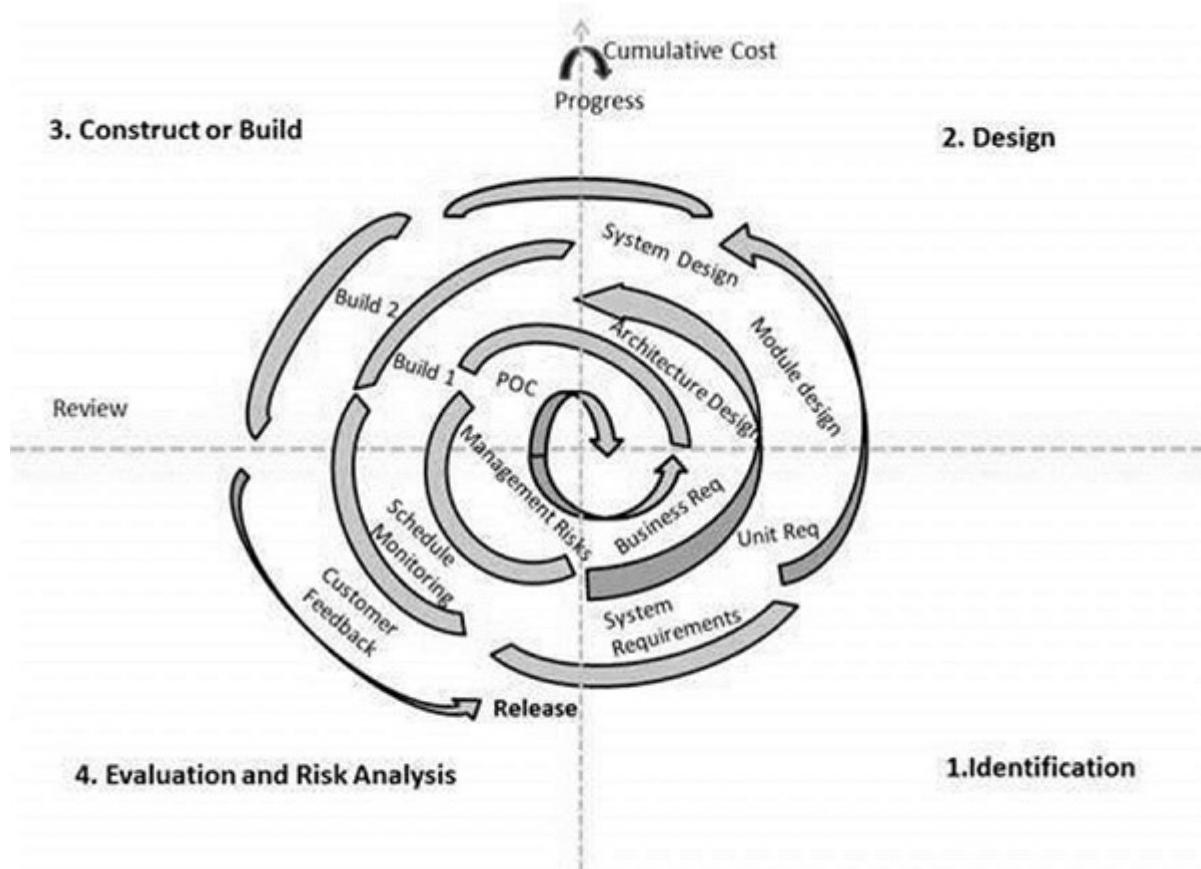
Music streaming from a web-server to an Android app and Website.

- Some working functionality can be developed quickly and early in the life cycle.
- Results are obtained early and periodically.
- Parallel development can be planned.
- Progress can be measured.
- Less costly to change the scope/requirements.
- Testing and debugging during smaller iteration is easy.
- With every increment, operational product is delivered.
- Risk analysis is better.
- It supports changing requirements.
- Initial Operating time is less.

Disadvantages:

- High amount of resources may be required.
- Defining increments may require definition of the complete system.
- End of project may not be known which a risk is.
- More management attention is required.
- Not suitable for smaller projects.
- Highly skilled resources are required for risk analysis.
- Management complexity is more.

C. Spiral Model



The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. This model may be a combination of iterative development process model and the waterfall model with a really high emphasis on risk analysis. It allows incremental releases of the product or incremental revisions through each iteration around the spiral. Each full circle around the centre is single iteration of software.

Applications:

- When there is a budget constraint and risk evaluation is important.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- For medium to high-risk projects.
- We are not sure of our project requirements.

Music streaming from a web-server to an Android app and Website.

- Requirements are complex and need evaluation to get clarity.
- Significant changes are expected in the product during the development cycle.

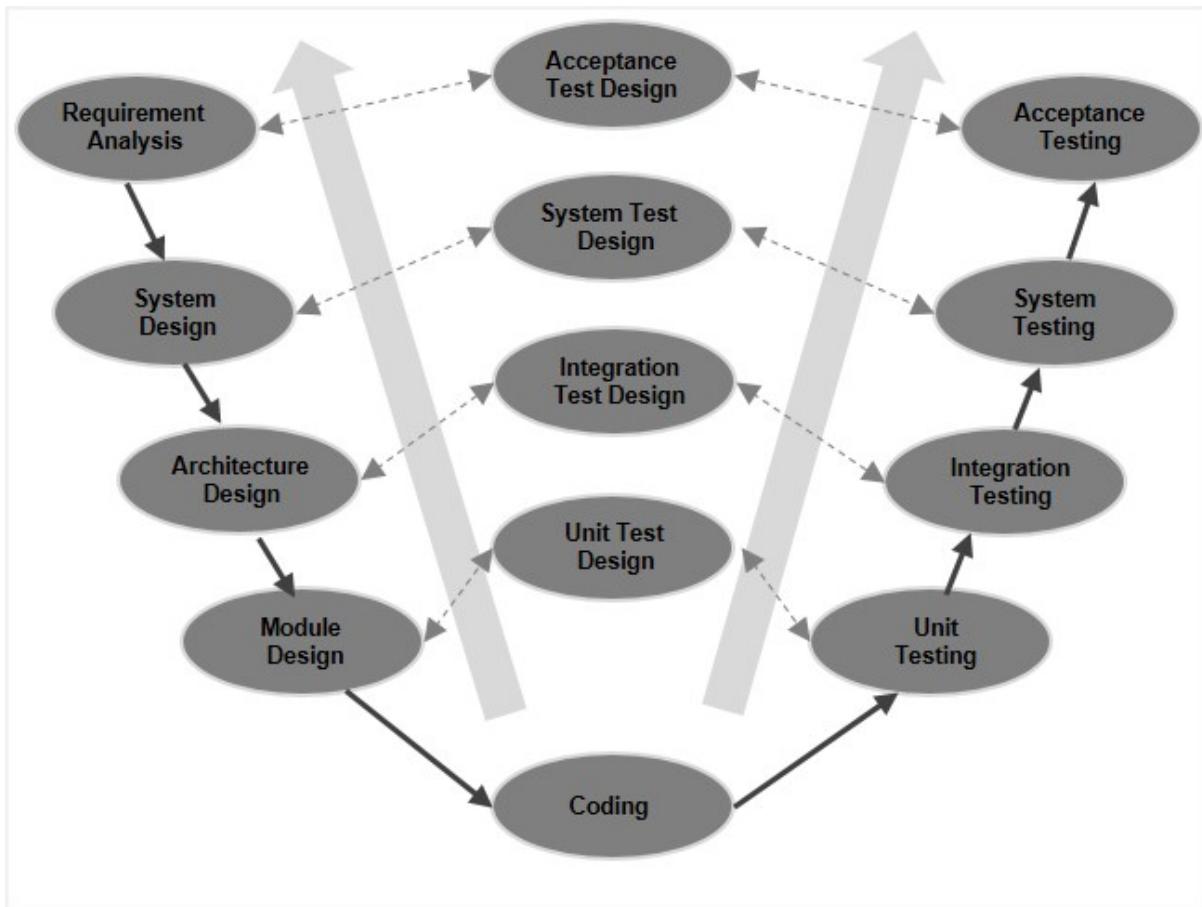
Advantages:

- Users see the system early.
- Allows extensive use of prototypes.
- Requirements can be captured more accurately.
- Changing requirements can be accommodated.
- Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

Disadvantages:

- Not suitable for small or low risk projects and could be expensive for small projects.
- Management is more complex.
- Process is complex
- Spiral may go on indefinitely.
- End of the project may not be known early.
- Large number of intermediate stages requires excessive documentation.

D. V-Model



Also known as Verification and Validation model. Extension of Waterfall model.

Highly disciplined model, next step starts only after the end of the previous step. This model is based on the association of a testing phase for each corresponding development stage.

Which means for each stage in development phase, there's a corresponding testing stage.

In the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the 'V' and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model.

Applications:

- Requirements are well defined, clearly documented and fixed.
- The project is short.

Music streaming from a web-server to an Android app and Website.

- Technology is not dynamic and is well understood by the project team.
- Product definition is stable.
- There are no ambiguous or undefined requirements.

Advantages:

- This is a highly-disciplined model and Phases are completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Simple and easy to understand and use.

Disadvantages:

- High risk and uncertainty.
- No working software is produced until late during the life cycle.
- Not a good model for complex and object-oriented projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.
- Poor model for long and ongoing projects.
- Once an application is in the testing stage, it is difficult to go back and change a functionality.

E. Big Bang Model

Big Bang Development model usually used for small projects and for smaller teams. In this model we don't follow any specific process. Development starts with required money and inputs, and output is software developed which may or may not be as per customer requirement. This Big Bang Model does not follow a process/procedure and there is a very little planning required. Even the customer is not sure about what exactly he wants and the requirements are implemented on the fly without much analysis.

Applications:

- Ideal for small projects.
- Useful for academic and practise projects.
- For a project where requirements are not well understood.
- For a project where the final release date is not given.

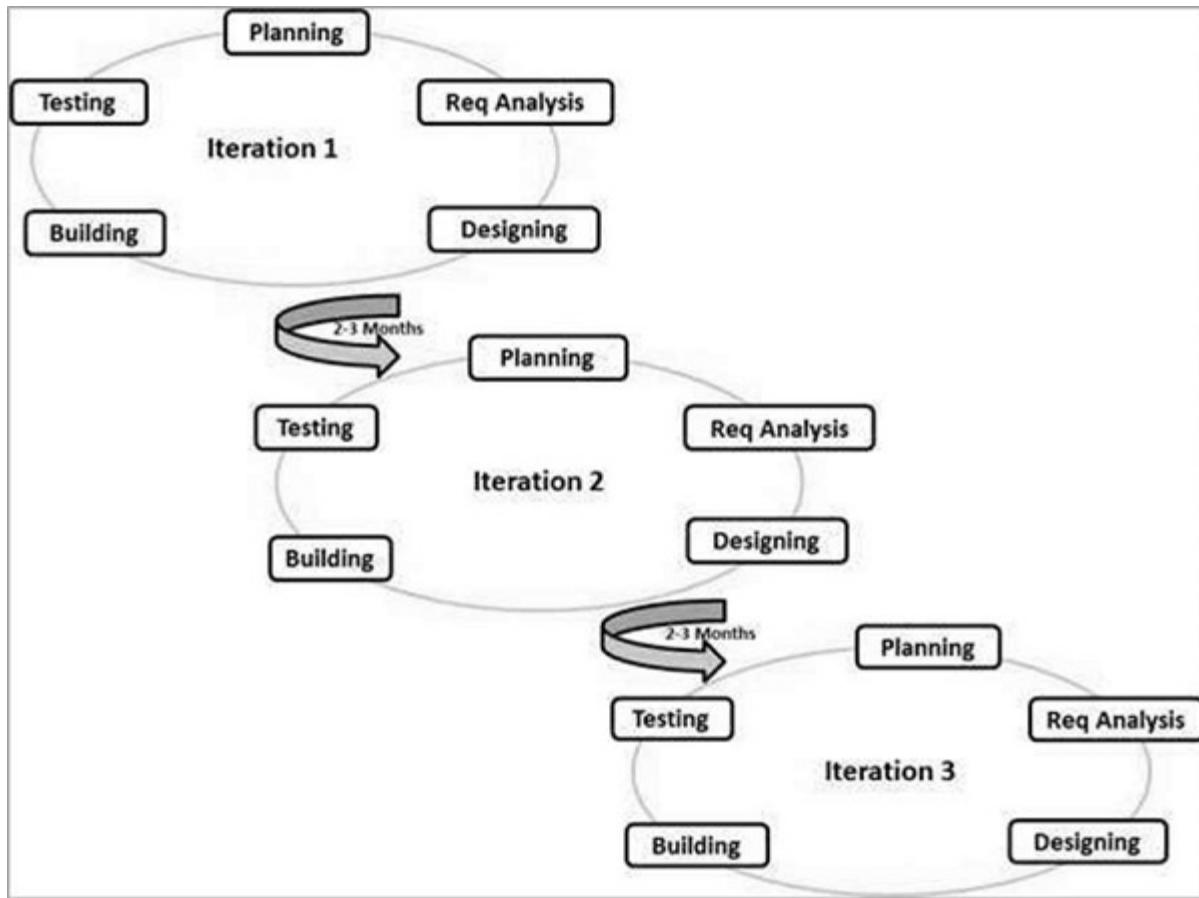
Advantages:

- Very few resources required
- Easy to manage
- Gives flexibility to developers
- Little or no planning required
- It is a good learning aid for new comers or students.
- This is a very simple model

Disadvantages:

- Very High risk and uncertainty.
- Not suitable for long and ongoing projects.
- Not a good model for complex and object-oriented projects.
- Can turn out to be very expensive if requirements are misunderstood.

F. Agile Model



In this model we believe that each project needs to be handled differently. And that existing methods need to be tailored for the project's requirement. Tasks are divided in smaller time frames so we can deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

Agile is based on the adaptive software development methods

Some of the popular Agile methods:

- A. Rational Unified Process (1994)
- B. Scrum (1995)
- C. Crystal Clear

Music streaming from a web-server to an Android app and Website.

- D. Extreme Programming (1996)
- E. Adaptive Software Development
- F. Feature Driven Development
- G. Dynamic Systems Development Method (DSDM) (1995)

These are now collectively referred to as Agile Methodologies, after the Agile Manifesto was published in 2001.

Agile Manifesto principles:

- A. Individuals and interactions
- B. Working software
- C. Customer collaboration
- D. Responding to change

Applications:

- When high customer interaction is possible.
- Non-complex dependencies.
- When less resources are required.

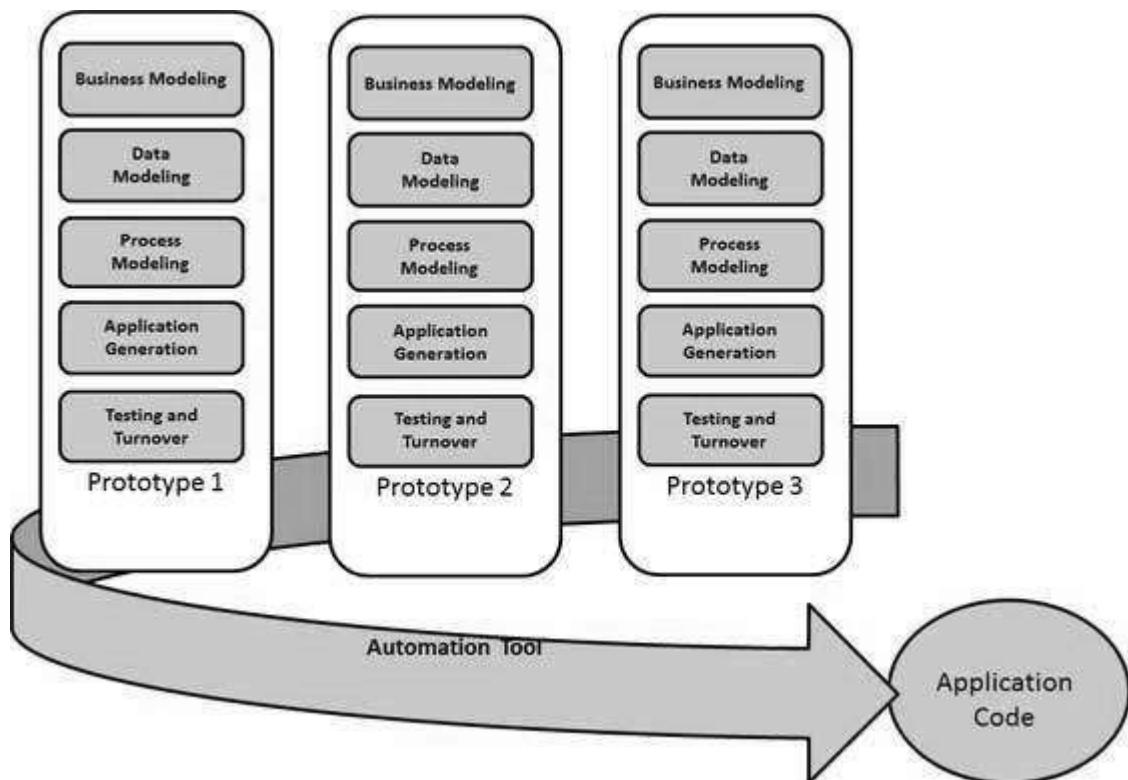
Advantages:

- Promotes teamwork and cross training.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Functionality can be developed rapidly and demonstrated.
- Delivers early partial working solutions.
- Enables concurrent development and delivery within an overall planned context.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements
- Little or no planning required.
- Easy to manage.

Disadvantages:

- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Not suitable for handling complex dependencies.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- There is a very high individual dependency, since there is minimum documentation generated.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

E. RAD Model



RAD stands for Rapid Application Development model. It is based on prototyping and iterative development when no specific planning involved. The process of writing the software itself involves the planning required for developing the product. The process of

Music streaming from a web-server to an Android app and Website.

writing the software itself involves the planning required for developing the product. Rapid Application Development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes (components), continuous integration and rapid delivery. In this model we favour Rapid Prototyping and use minimal planning.

Applications:

- It should be used if there is a high availability of designers for Modelling.
- RAD should be used only when a system can be modularized to be delivered in an incremental manner.
- RAD model should be chosen only if domain experts are available with relevant business knowledge.
- It should be used only if the budget permits use of automated code generating tools.
- Should be used where the requirements change during the project and working prototypes are to be presented to customer in small iterations of 2-3 months.

Advantages:

- Changing requirements can be accommodated.
- Iteration time can be short with use of powerful RAD tools.
- Productivity with fewer people in a short time.
- Progress can be measured.
- Reduced development time.
- Quick initial reviews occur.
- Encourages customer feedback.
- Increases reusability of components.
- Integration from very beginning solves a lot of integration issues.

Disadvantages:

- High dependency on modelling skills.

Music streaming from a web-server to an Android app and Website.

- Inapplicable to cheaper projects as cost of Modelling and automated code generation is very high.
- Management complexity is more.
- Dependency on technically strong team members for identifying business requirements.
- Only system that can be modularized can be built using RAD.
- Requires highly skilled developers/designers.
- Suitable for systems that are component based and scalable.
- Requires user involvement throughout the life cycle.
- Suitable for project requiring shorter development times.

F. Software Prototype Model

In this model we build software application prototypes which displays the functionality of the product under development, but might not actually include the exact logic of the original project.

Software prototyping is becoming very popular as a software development model, as it enables to understand customer requirements at an early stage of development. It helps get valuable feedback from the customer and helps software designers and developers understand about what exactly is expected from the product under development.

Applications:

- High level of user interactions.
- Projects with form fields which users will fill up.

Advantages:

- Increased user involvement in the product even before its implementation.
- Reduces time and cost as the defects can be detected much earlier.
- Quicker user feedback is available leading to better solutions.

Music streaming from a web-server to an Android app and Website.

- Since a working model of the system is displayed, the users get a better understanding of the system being developed.
- Missing functionality can be identified easily.
- Confusing or difficult functions can be identified.

Disadvantages:

- Users may get confused in the prototypes and actual systems.
- Risk of insufficient requirement analysis owing to too much dependency on the prototype.
- Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- The effort invested in building prototypes may be too much if it is not monitored properly.
- Developers may try to reuse the existing prototypes to build the actual system, even when it is not technically feasible.

With all the information in mind about the models mentioned above. We selected appropriate model for development of our project.

PERT & GANTT chart

PERT Chart

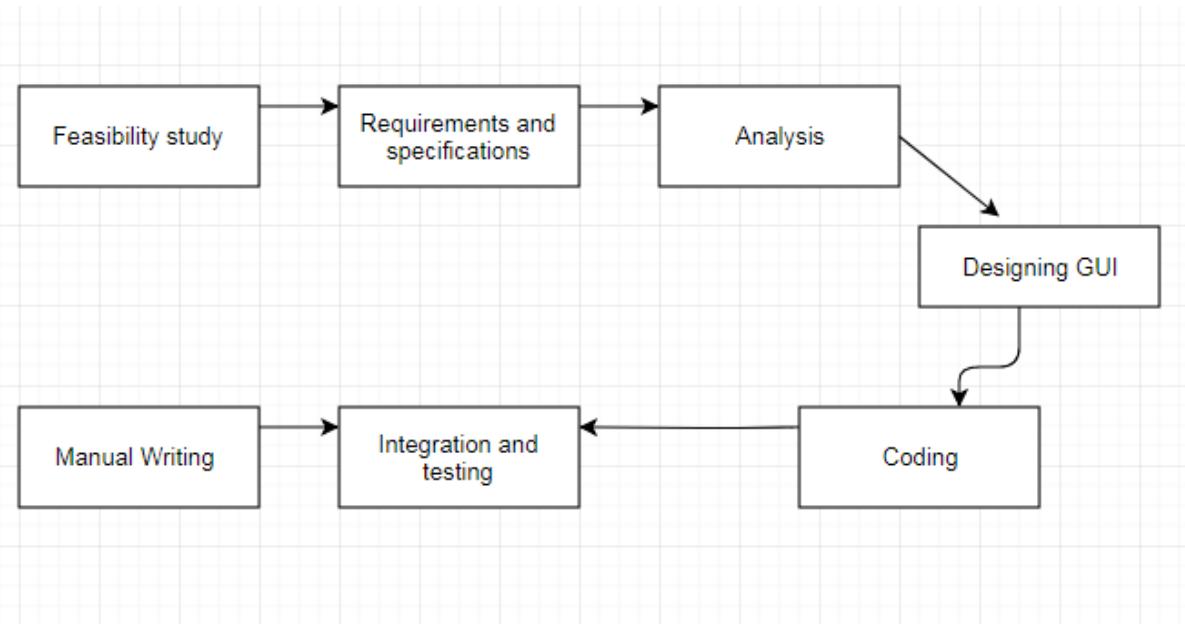
PERT stands for Program Evaluation Review Technique. A PERT chart is a project management tool that provides a graphical representation of a project's timeline. It breaks down the individual tasks of a project for analysis. PERT charts are often more difficult to interpret.

PERT charts were first created by the U.S. Navy's Special Projects Office in 1957 to guide the Polaris nuclear submarine project.

Music streaming from a web-server to an Android app and Website.

A PERT chart uses circles or rectangles called nodes to represent project events or milestones. These nodes are linked by vectors, or lines, that represent various tasks.

A PERT chart allows managers to evaluate the time and resources necessary to manage a project.



PERT offers a number of advantages to project managers. For example, it forces us to organize and quantify project information and provides us with a graphic display of the project. It also helps us to identify which activities are critical to the project completion time and should be watched closely, and which activities involve slack time and can be delayed without affecting the project completion time. The chief disadvantages of PERT lie in the nature of reality. Complex systems and plans, with many suppliers and channels of supply involved, sometimes make it difficult to predict precisely what will happen. The technique works best in well-understood engineering projects where sufficient experience exists to predict tasks accurately in advance.

GANTT chart

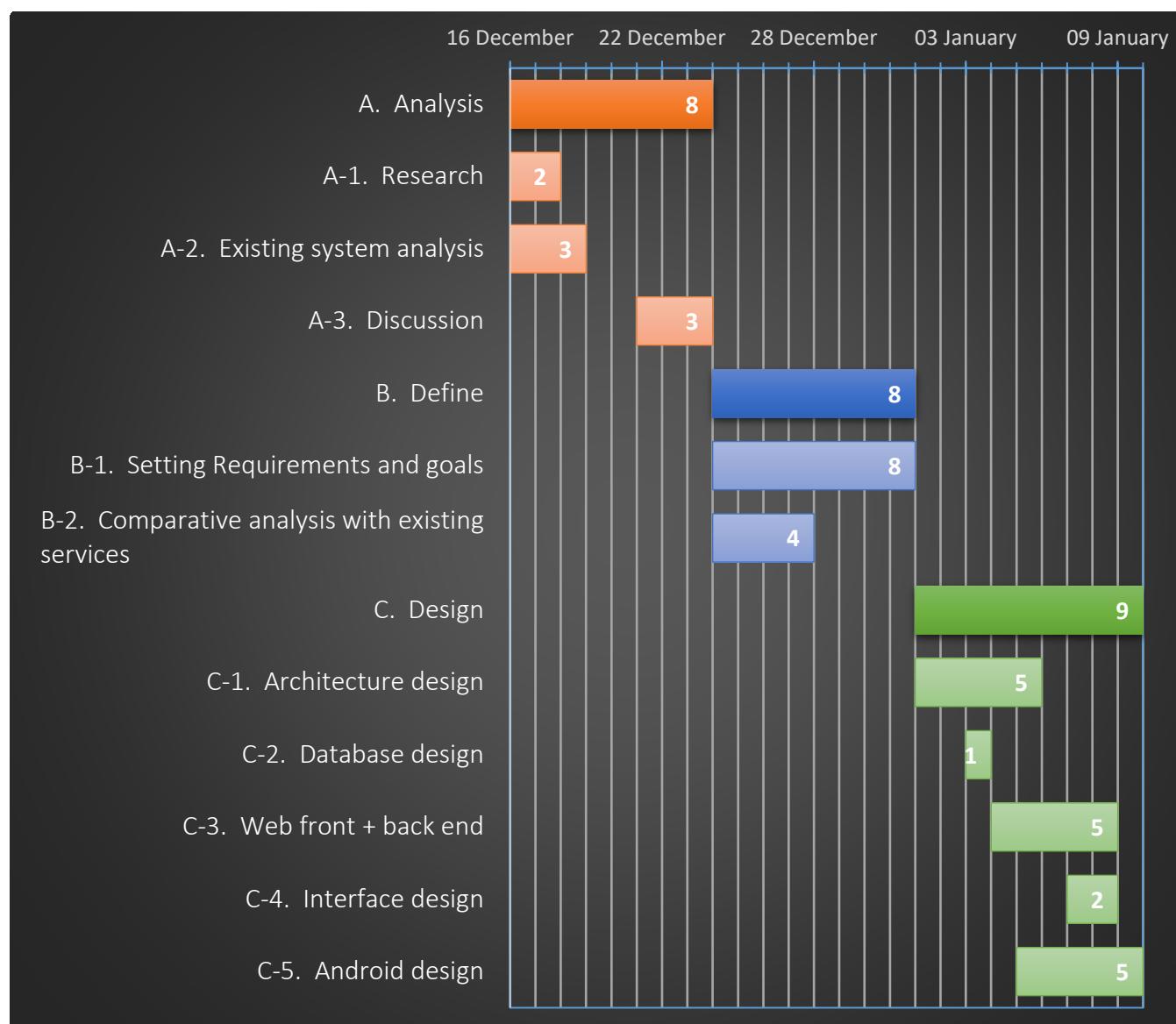
A Gantt chart, or harmonogram, may be a sort of bar graph that illustrates a project schedule. GANTT chart lists the tasks that we have to perform on the vertical axis, and time intervals on the horizontal axis. The width of the horizontal bars represents the duration of each activity. Gantt charts illustrate the beginning and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements constitute the

Music streaming from a web-server to an Android app and Website.

work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts are often wont to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.

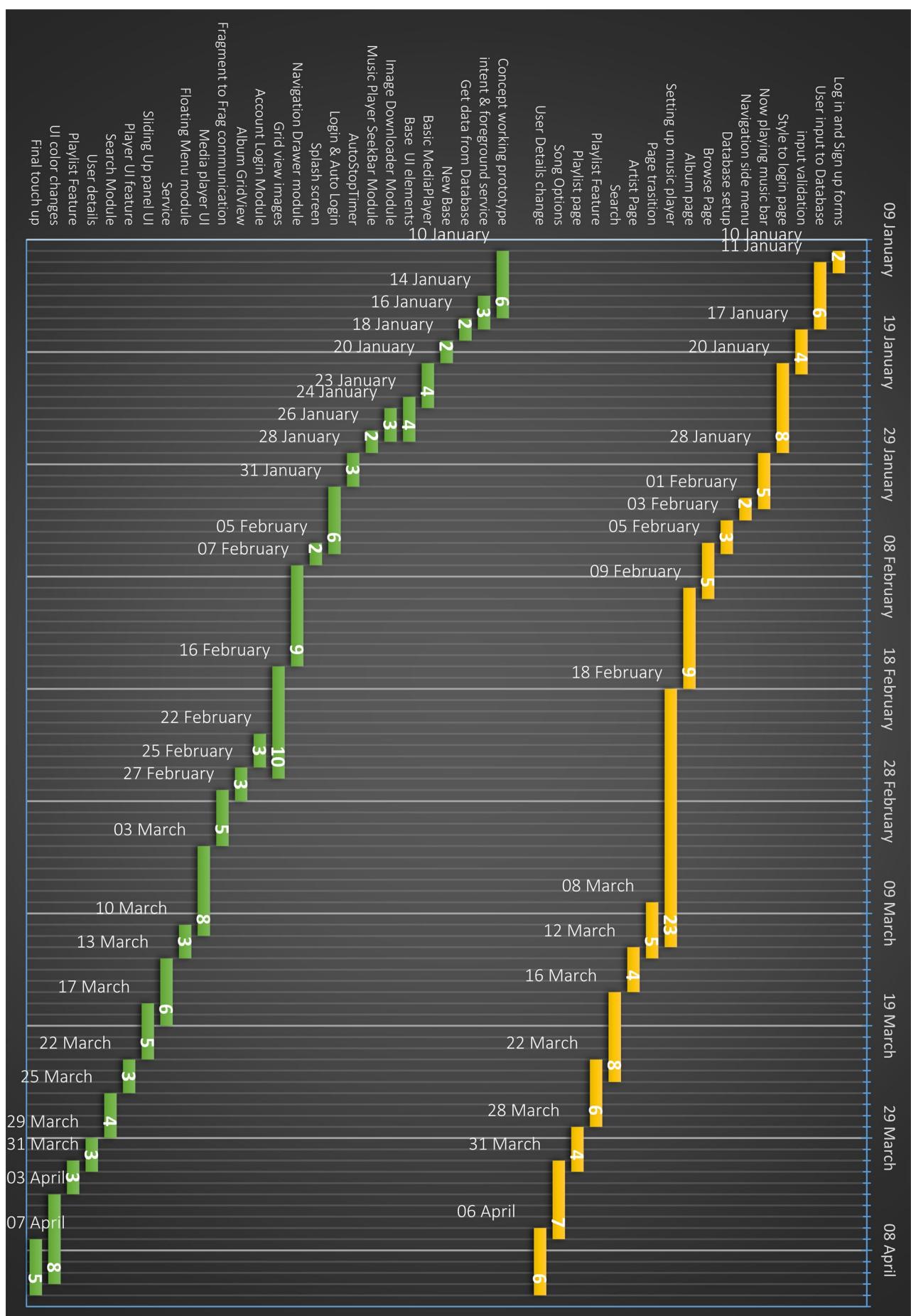
Gantt charts are sometimes equated with bar charts. Gantt charts are usually created initially using an early start time approach, where each task is scheduled to start out immediately when its prerequisites are complete. This method maximizes the float time available for all tasks.

Our project's GANTT CHART



Gantt Diagram: **Analysis, Definition, Design Stage (16 December 2019 ~ 10 January 2020)**

Music streaming from a web-server to an Android app and Website.



Music streaming from a web-server to an Android app and Website.

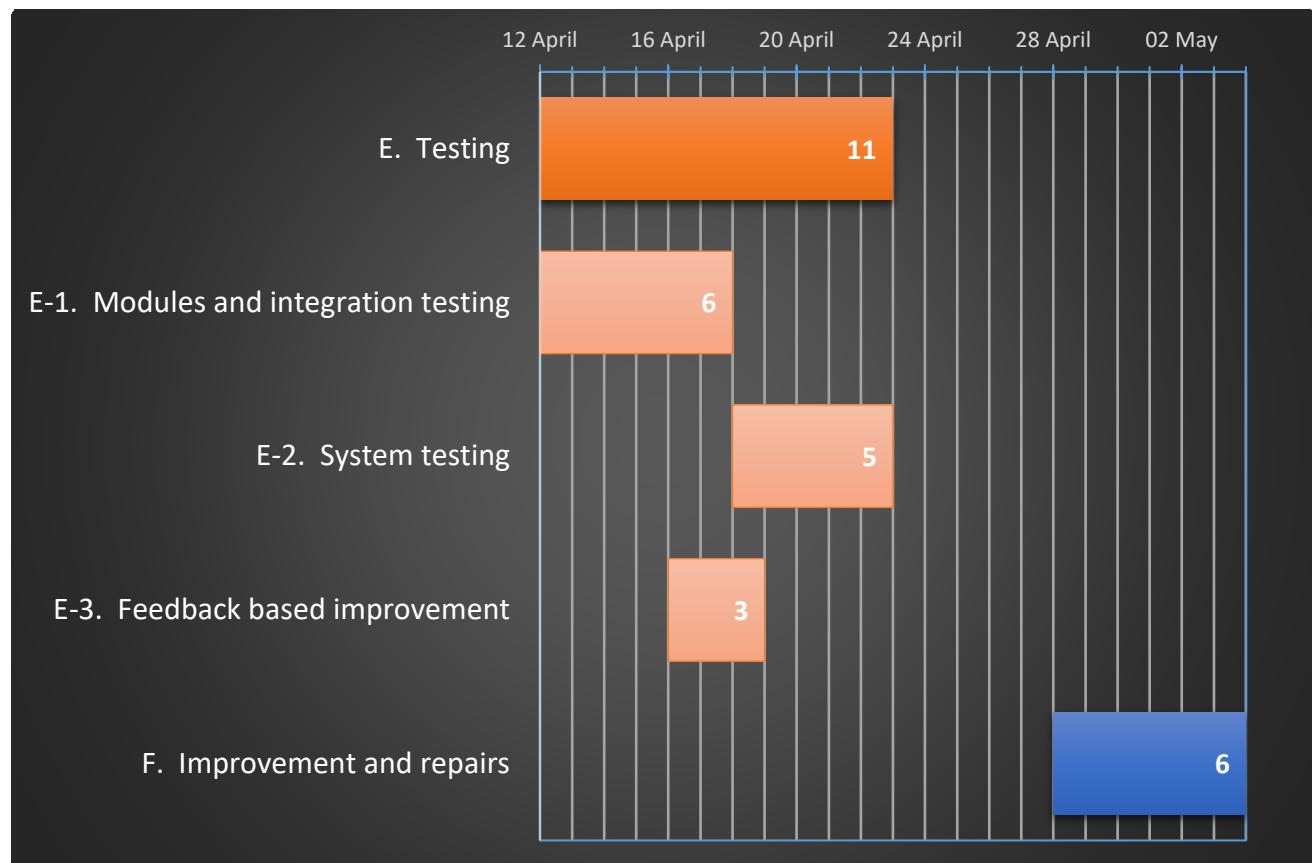
Previous Gantt Chart: **Development stage (09 January 2020 ~ 13 April 2020)**

Green: Android

Yellow: Web - Front End & Back End

We have only mentioned a few steps and features.

The project was live, and it was developed and tested while being deployed.



Gantt Diagram: **Testing Stage (18 April 2020 ~ 04 May 2020)**

In all of the grant charts, the distance between each vertical line represents one day.

CHAPTER 6

REQUIREMENT ANALYSIS

Overview

We study the Goals of our project, and clearly define what requirements are from our system? And what it needs to do? We determine the needs or conditions to meet for our project.

Now we'll see important Features and Cost that we need to consider before we build our project.

Features and Cost

A. Features:

- a. Music player
 - i. Ultra Essential: Pause, Play
 - ii. Essential: Next, Previous
 - iii. Repeat, Repeat single
 - iv. Shuffle, Random
 - v. Extra / Advanced Features:
 - 1. Auto Next
 - 2. Auto Stop Timer
 - 3. Server managed synchronised current music queue
- b. User registration.
- c. Search functionality.
- d. Browsing via categories.
- e. My Music or Playlists.
- f. Settings, and editing user profile.

B. Technology Stack

C. Price and Costs

A. Features

Features can be divided into Essential, and Advanced grouping. First we'll make Essential features available and then the rest. These are our project's functional requirements

As per analysis of other services we found these features:

a. Music player

This is the Main software component on our front page on website, and in our app on mobile device which plays or streams the given music file from our server. There are multiple methods of streaming, we can have it preloaded wholly, or send it to the device or front end bit by bit.

i. Ultra Essential: Pause, Play

This is the functionality to pause the track being played in the middle, and when user presses play the track shall resume from that point.

ii. Essential: Next, Previous

This is the functionality play the next or previous file when users wants to, disregarding whether the current track has been completed.

iii. Repeat, Repeat single

From surveyed music services and device based music players, many offer ability to either repeat the entire list after the end has reached OR to repeat the same track over and over.

iv. Shuffle, Random

This is the functionality where it randomly plays any track from the current queue after the playing track has completed.

v. Extra / Advanced Features:

These are the features we'll implement at last, as many of them are non-essential. Some of these features are unique to our project for example Auto Stop Timer, and Server managed synchronised music queue.

- Server managed synchronised current music queue

We plan to implement a music queue or list which is synchronised with the server, and hence the same user account may be used simultaneously on multiple devices, and they all will get to synchronise their lists.

- Auto Next

This is option to Enable or Disable if the player plays the next track (if any) in the list. Usually this options is never shown and is enabled in most of the apps and services. But we try to give a on or off option. And let the listener decide if they wish to go to next track or not automatically.

- Auto Stop Timer

This is rather unique feature, in all of the streaming services only a few have this option. Spotify recently added this in April 2020. Some of the Music Players have this feature. This will let the user decide after how many minutes they wish for their music to stop.

b. User registration.

This is essential feature if we wish to have user profile in our service. By having user profile we can manage their data, which will enable us to provide better service to the user, with giving them access to Playlist features, and possible expansion with Machine Learning in the future.

Different services have different approaches to how they allow user to create their profile. Some have form fields, while some let this be handled by Facebook or Google totally. This makes the security responsibility less on the service's end.

For our service, we will let user keep their data with us in our database. And for security we'll use MD5 hashing for their password, so no one with access to our database may read and know the password.

c. Search functionality.

Search is nearly very basic needed feature. Each time the browse page may differ with time, and so to find the favourite track user can use this, they can use it to find favourite artist and albums too.

d. Browsing via categories.

Browse page allows user to discover some new Album or Artists or Genre or a Track which may be inside of these category.

e. My Music or Playlists.

This is a feature which can be considered important. This gives ability to user to make a list of the tracks as per user liking, and have it in any way they wish as per user's discretion. Without this feature having user profiles will make less sense.

f. Settings, and editing user profile.

This allows the user to change their Email and password. This feature is also a must, because email Id may change for a user. Or sometimes users wishes to change their password for better security.

Design

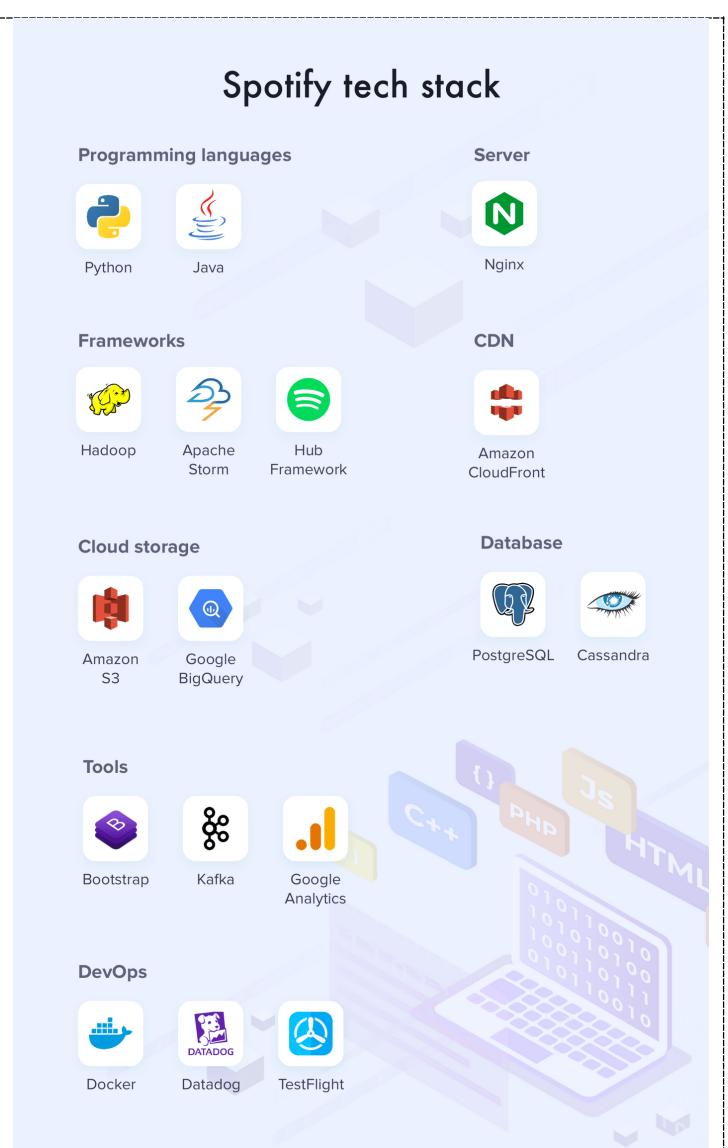
We must pay attention to the UI/UX design. Our goal is to make the service as intuitive and user-friendly as possible. Both the platforms that we target, Web from desktop browser, and Android app should have uniformity in their design.

B. Technology Stack

Music streaming services require a scalable technology stack. Additionally we need a scalable database.

Let's consider Spotify tech stack.

- A. Programming languages: Python, Java
- B. Frameworks: Hadoop, Apache Storm, Hub Framework
- C. Server: Nginx
- D. Cloud storage: Amazon S3, Google BigQuery
- E. CDN: Amazon CloudFront
- F. Database: PostgreSQL, Cassandra
- G. Tools: Bootstrap, Kafka, Google Analytics
- H. DevOps: Docker, Datadog, TestFlight



Choosing the tech stack for a music streaming service, we need to either consult with professionals, people who have experience or do proper research. The final tools and tech stack that we use always depends on our project's technical needs and project's requirements.

Our project's Tech Stack:

- Programming languages: Java, PHP, AJAX, JS
- Server: XAMPP Stack, Unix Socket, PHP engine
- Database: MySQL, MariaDB on UNIX socket
- Front End Web: HTML, CSS, JS, AJAX
- Android Application: Java, XML for android design

Android Libraries:

- Androidx
- MDC
- amitshekhar.android:android-networking
- picasso:2.71828
- sothree.slidinguppanel:library:3.4.0

- Storage: 000webhost server based limited storage

Programming languages



Server



Front End



Database



Android



Hosting storage



Music streaming from a web-server to an Android app and Website.

After studying tech stack that is required or essential for our whole project to work, we can figure out the software and hardware requirements.

- Software Requirements:

- Android (For App):

- Android API 25 (Android 7.1) or better.

- For Web version (For Website):

- Any HTML5 supported modern Web browser.
 - Chromium based browser preferred.
 - Preferably Desktop browser.

- Hardware Requirements:

- Android (For App):

- 3 GB of ram or more.
 - 1 GB of free space.

Total ~30 MBs Space requirement for app when it is installed.

15.11 MB App (Actual Storage requirement for our app.)

299 KB User Data (For various user preferences.)

9.13 MB Cache (Android caches data that is repeatedly needed.)

- Preferred SOC: MediaTek Helio P22 MT6762
 - 6 or more cores preferred.
 - Communication module(s):

- WLAN OR LTE / Cellular Data.

- For Web version (For Website) Desktop preferred:

- 4 GB of ram or more.
 - 6 GB of free space.
 - Dual-Core CPU (Intel 3rd generation or above).
 - Communication module(s):

- WLAN OR LAN.

C. Price and Costs

We need hosting for our server

Types of hosting:

- Shared Hosting
- VPS Hosting
- Dedicated Hosting
- Cloud Hosting
- WordPress Hosting

Hosting price range

Type of hosting	Minimum	Monthly	Maximum	Monthly
Shared Hosting	2.5 \$	190 ₹	15 \$	1,200 ₹
Dedicated Hosting	80 \$	6,000 ₹	730 \$	55,000 ₹

For our project least costly shared hosting can also work for developing, testing, and demonstration purposes.

For this project we have selected: 000webhost.com and it gives option to host for free, with least limitations. And after review they provide all the services and technology that we need.

- We have examined the time requirement for components of our project in Schedule Feasibility.

ABOUT TECHNOLOGY THAT WE HAVE USED

- A. Android OS
- B. JAVA
- C. Android SDK
- D. Dalvik VM / ART

Music streaming from a web-server to an Android app and Website.

- E. Android Studio
- F. XML
- G. Androidx
- H. MDC
- I. amitshekhar.android:android-networking
- J. picasso:2.71828
- K. sothree.slidinguppanel:library:3.4.0
- L. PHP
- M. AJAX
- N. JavaScript
- O. XAMPP
- P. UNIX socket
- Q. MySQL
- R. MariaDB
- S. HTML5
- T. CSS
- U. jQuery

A. Android OS

We are going to target Android OS based devices, because of the wide user base.

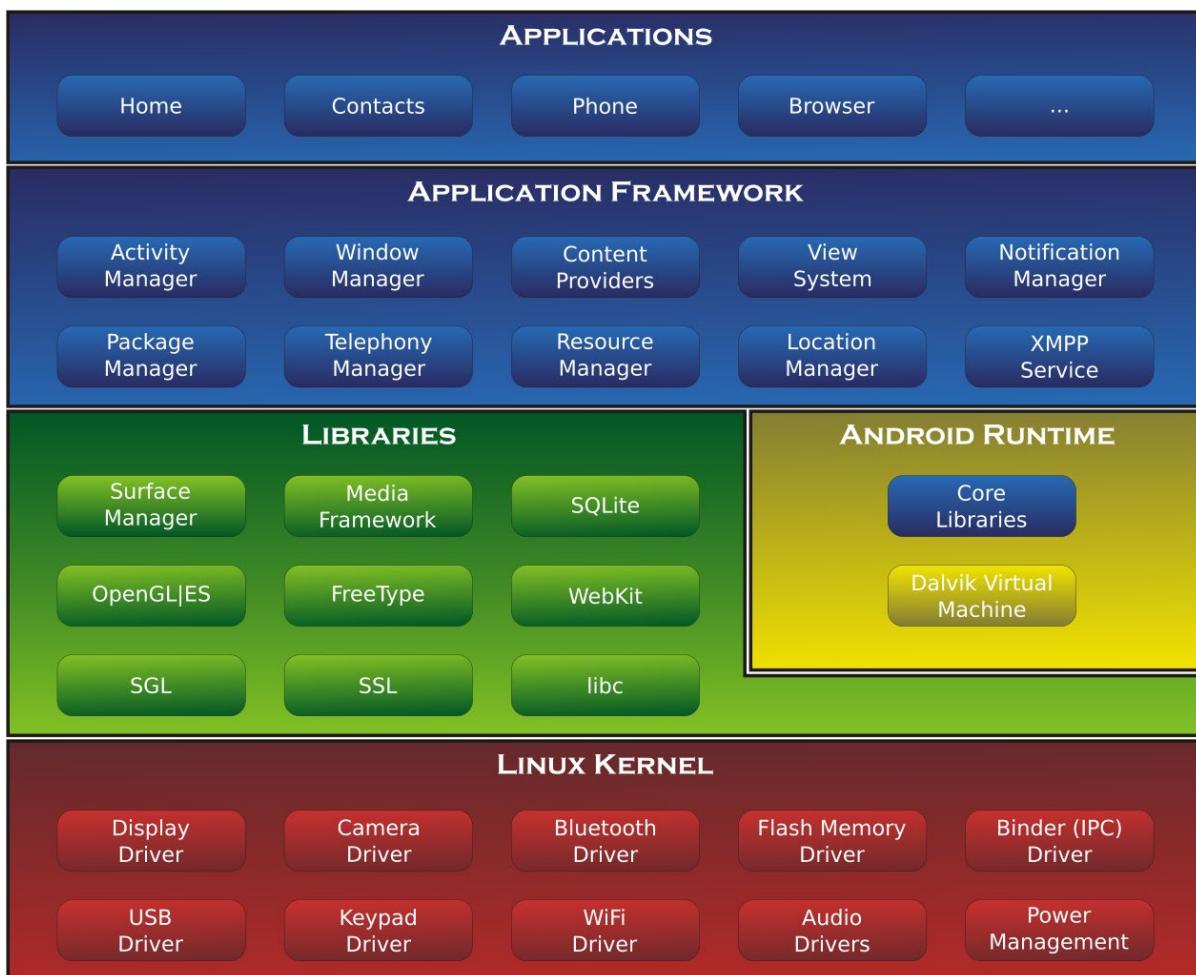
Android is Linux based Open source operating system. It is made for portable mobile handset devices, which should be using processors with RISC architecture. Although Android OS has versions made for x86 Architecture hence can be used on x86 and x86_64 processors (CISC). Android is largely targeted at touch input based handsets.

It is developed and managed by OHA - Open Handset Alliance. Google is commercial sponsor and also a part of OHA. However most of the android devices which we the end user uses are Android mixed with custom soft wares. This customisation is from Google and our device manufacturers. About 70 percent of Android smartphones run Google's ecosystem.

Java and Kotlin are used for developing Android apps. Along with a few other languages, framework, and technologies. Kotlin is a JVM based language.

Music streaming from a web-server to an Android app and Website.

Android System Stack:



B. JAVA

Java is the technology of choice for building applications using managed code that can execute on mobile devices. Java used to be and still is very popular, although in recent years (2015~2020), the popularity of JAVA is decreasing. Android applications can be developed by using the Java programming language and the Android SDK. We are choosing to use Java for android app development. Java emerged in mid 1990s, made by Sun Microsystems, which was later bought by Oracle.

Java programs are secure because they run within a sandbox environment. Programs written in Java are compiled into intermediate code known as bytecode. This bytecode is then executed inside the context of the Java Virtual Machine.

C. Android SDK

A SDK is Software Development Kit, Which are collection of tools to develop software for a specific framework or platform. Usually we can install the entire collection from single installation file. For android, Android SDK is provided. Google is developer behind the Android SDK. This closely works with Android Studio, which is official IDE for android development since 2015.

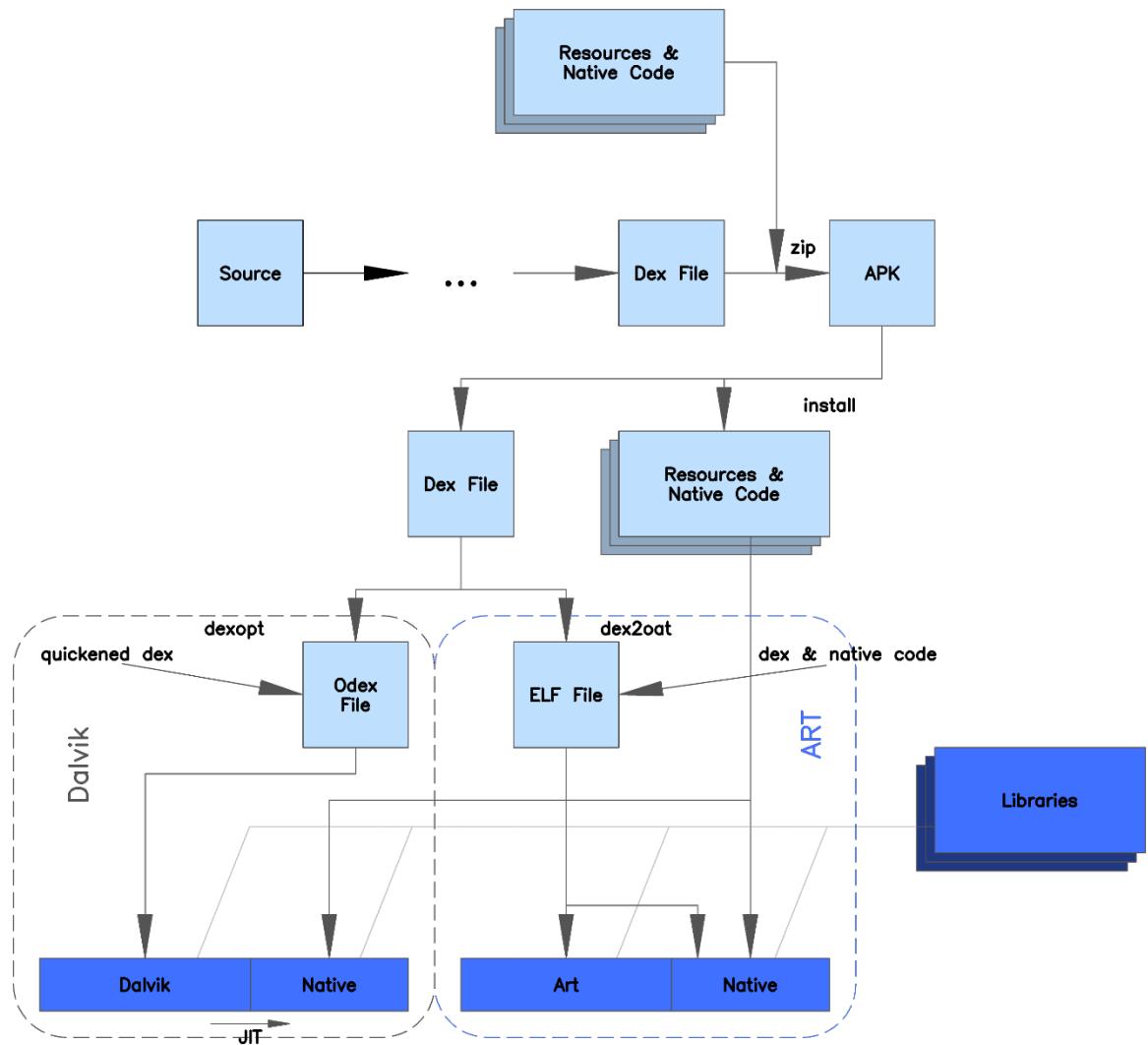
This includes debugger, libraries, may include handset emulator, documentation, and some sample code. Some other components which we get along with SDK are: adb-Android Debug Bridge. SDK works alongside Android NDK (Native Development Kit). This SDK is made available for Windows, macOS, Linux.

D. Dalvik VM / ART

Dalvik is a Virtual machine in android, which is no longer used, instead currently Android RunTime (ART) is used. Dalvik's bytecode format is still used. Dalvik was integral part of Android OS till version 4.4. When ART replaced Dalvik VM, it stopped use of .odex files, however .dex files (Both from dalvik) are still supported. This was done so the performance improvement would be transparent to the end user. ART performs the translation of the application's bytecode into native instructions that are later executed by the device's runtime environment. Unlike Dalvik, ART introduces the use of ahead-of-time (AOT) compilation by compiling entire applications into native machine code upon their installation. By eliminating Dalvik's interpretation and trace-based JIT compilation, ART improves the overall execution efficiency and reduces power consumption, which results in improved battery autonomy on mobile devices.

Music streaming from a web-server to an Android app and Website.

A comparison of Dalvik and ART architectures:



E. Android Studio

This is the officially supported IDE since 2015, from google. Currently 4.0 version is latest. This IDE is based on IntelliJ IDEA. This scheme of tools includes everything that is recommended for android development. When Android Studio was released it was meant to replace eclipse studio, which used to be officially supported with the help of plugins. Android Studio supports all the same programming languages of IntelliJ (and CLion) e.g. Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin.

We will be using Android Studio 3.5 and above if any update becomes available.

F. XML

Android layouts are written in eXtensible Markup Language, also known as XML.

Much like HTML (or HyperText Markup Language), XML is also a markup language. It was created as a standard way to encode data in internet-based applications. However, unlike HTML, XML is case-sensitive, requires each tag is closed properly, and preserves whitespace. To alter any layout we use JAVA.

Resources are the additional files and static content that an application requires, such as animations, color schemes, layouts, menu layouts. We use multiple layouts from Android and some animations all written in XML for our project.

Common Root level layouts:

- Linear layout: For Vertical OR Horizontal Oriented layout
- Relative layout: For displaying child elements relative to the position of parent layout
- Frame layout: Used as a placeholder.

G. Androidx

AndroidX is Android Extension Library. AndroidX is a redesigned library to make package names more clear. It is a library, in which many components required for android development are defined, with use of this, we can import the code for an element that we chose or require to use. AndroidX is the open-source project that the Android team uses to develop, test, package, version and release libraries within Jetpack.

AndroidX is a major improvement to the original Android Support Library. Like the Support Library, AndroidX ships separately from the Android OS and provides backward-compatibility across Android releases. AndroidX fully replaces the Support Library by providing feature parity and new libraries.

H. MDC

After introduction of the Material UI guidelines by android team, people tried to stay really close to the guidelines and by doing so they basically replicated Google's designs. And

so google realised that many android apps end up looking like differently coloured variant of Google's apps, hence they made MDC library. In this they have made a lot of android's UI elements available, with more flexibility and have published guidelines along with examples by creating case studies.

Due to less stability of MDC we will prefer older stable UI libraries and will use MDC whenever necessary or possible.

I. amitshekhar.android:android-networking

'Fast Android Networking' is a library made by Amit Shekhar, Who has made it open source and available via GitHub. This library uses other relevant popular libraries, hence we can add one library instead of all those libraries, for example: OkHttpClient, Okio. We require this library for JSON parsing, network data parsing, and to send and receive data to and fro network. Also removal of HttpClient in Android Marshmallow (Android M) made other networking library obsolete, hence this library is better suited option.

J. picasso:2.71828

This is a powerful image downloading and caching library for Android. It is very popular and used widely, and hence it is supported more than other similar libraries. We use it to download and display image content in our app.

K. sothree.slidinguppanel:library:3.4.0

This library provides a simple way to add a draggable panel which slides up in our Android application as seen in Google Music app and Google Maps app. We house our music player User Interface inside the activity which uses this as a layout. This is the same library being used by: DropBox Paper app, and Snaptee app.

L. PHP

PHP stands for Hypertext Pre-processor. It is an open source scripting language, and it is widely-used. The script are executed on the server. PHP is free to use. PHP file can contain:

text, HTML, CSS, JavaScript and PHP code itself. When we as a client use a browser to connect to a web address. We are essentially connecting to the web server and we are requesting the file for that page. So when the server has PHP installed, and the file which we want has PHP code in it, then it is executed and the result is sent back to our client device, along with everything else that may be in the file, i.e. HTML, CSS, JS code. That is not changed by the PHP engine, unless it is inside the PHP code. PHP files have .php extension. PHP can open, read, write, close files on server, and can connect and interact with the Database. PHP is used many times to collect data and process if required. PHP can generate dynamic page content. PHP also handles sending and receiving of cookies. PHP is compatible with most current servers, and supports wide range of databases.

Because of all these abilities, we are going to use PHP in our back end. We use it for front end website, and to connect, retrieve transfer data between our android app and database.

M. AJAX.

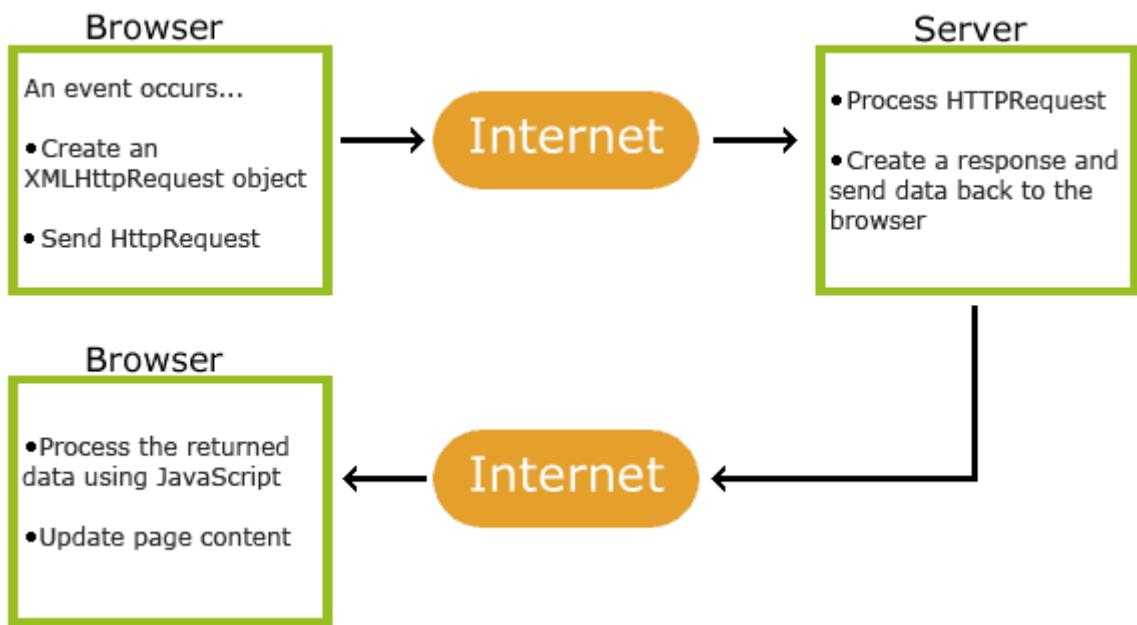
AJAX stands for Asynchronous JavaScript and XML. It is not considered a programming language. AJAX just uses a combination of: A browser built-in XMLHttpRequest object (to request data from a web server) and JavaScript and HTML Document Object Model. AJAX allows us to update only a part of our webpage with the data from our server, without reloading entire server.

In our web front end pages, we'll be using AJAX to retrieve, and process data.

Steps of AJAX process:

1. An event occurs in a web page (the page is loaded, a button is clicked).
2. An XMLHttpRequest object is created by JavaScript.
3. The XMLHttpRequest object sends a request to a web server.
4. The server processes the request.
5. The server sends a response back to the web page.
6. The response is read by JavaScript.
7. Proper action (like page update) is performed by JavaScript.

Music streaming from a web-server to an Android app and Website.



N. JavaScript

JavaScript is a lightweight, just in time compiled programming language. It used to be only used in front end of webpage. Now a days it's being used even in back end, for DB handling, and even outside the field of web programming. There are frameworks which allow us to make JavaScript code be executed on our Android or iOS device, or both. ECMAScript is a standard of JavaScript, made by ECMA, which is European Computer Manufacturer's Association. They set the standards. Latest is ECMAScript 2020, 11th edition.

We will only be using JavaScript on our web pages.

O. XAMPP

XAMPP is an open source, cross platform web server solution stack collection. It is developed by Apache Friends. Consists of components: Apache HTTP server, MariaDB, PHP and Perl interpreters. XAMPP is one of the most popular stack packages. XAMPP allows us to install every necessary component at once.

We will be using XAMPP at our web server.

P. UNIX socket

Music streaming from a web-server to an Android app and Website.

Sockets are communication points on the same or different computers to exchange data. Sockets are supported by Unix, Windows, Mac, and many other operating systems. A Unix Socket is used in a client-server application framework. A server is a process that performs some functions on request from a client. Most of the application-level protocols like FTP, SMTP, and POP3 make use of sockets to establish connection between client and server and then for exchanging data.

We'll be using this in our backend.

Q. MySQL

MySQL is the most popular Open Source Relational SQL Database Management System. MySQL is one of the best RDBMS being used for developing various web-based software applications. MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. It uses a standard form of the well-known SQL data language. It supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but we can increase this (if the operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

A database is an independent application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

R. MariaDB

MariaDB is an open source relational database management system (RDBMS) that is a compatible drop-in replacement for the widely used MySQL database technology. The developers who were important in development of MySQL created MariaDB as fork to MySQL. This was done because Oracle bought MySQL. It is based on SQL and supports ACID-Style date processing. It guarantees ACID for transactions. It also supports JSON APIs, parallel data replication and multiple storage engine. MariaDB was made in a way that it can be considered to have parity with MySQL. Basically the creators wanted a MySQL clone, because Oracle Inc. bought the parent company.

Music streaming from a web-server to an Android app and Website.

MariaDB can support multiple Storage Engines. We have used InnoDB as storage engine. For our project the version that we use is: 10.3.16-MariaDB.

S. HTML5

HTML stands for Hyper Text Mark-up Language. It is the standard markup language for Web pages. We design the web page with HTML, and style it with CSS. This is the standard to make web page. HTML5 is the latest evolution of HTML. However HTML5 May refer only to HTML or sometimes it is used as a reference to HTML5 and friends, this will include the latest technologies bundled with HTML5. This standard is recommended by World Wide Web Consortium, and it managed by Web Hypertext Application Technology Working Group.

For each webpage in our front end site we have used HTML and HTML5 standard to construct and build web pages.

T. CSS

CSS stands for Cascading Style Sheet, the standard we used is CSS3, however CSS4 does exist, but there's no single standard specification. CSS is used to give style and to customise HTML elements. Without CSS HTML structure on its own wouldn't be alluring. With proper implementation of CSS one can build stunning sites.

U. jQuery

jQuery is a small, fast and feature-rich JavaScript library. It gives us easy to use API, with which we simplify HTML document traversal, manipulation, event handling, animation, and AJAX. It is supported in many current browsers. It is considered a lightweight, "write less, do more", library. It simplifies a lot of things, by converting what would be a multiline code in standard JavaScript into a single line of Code.

We use it a few places in our web pages.

CHAPTER 7

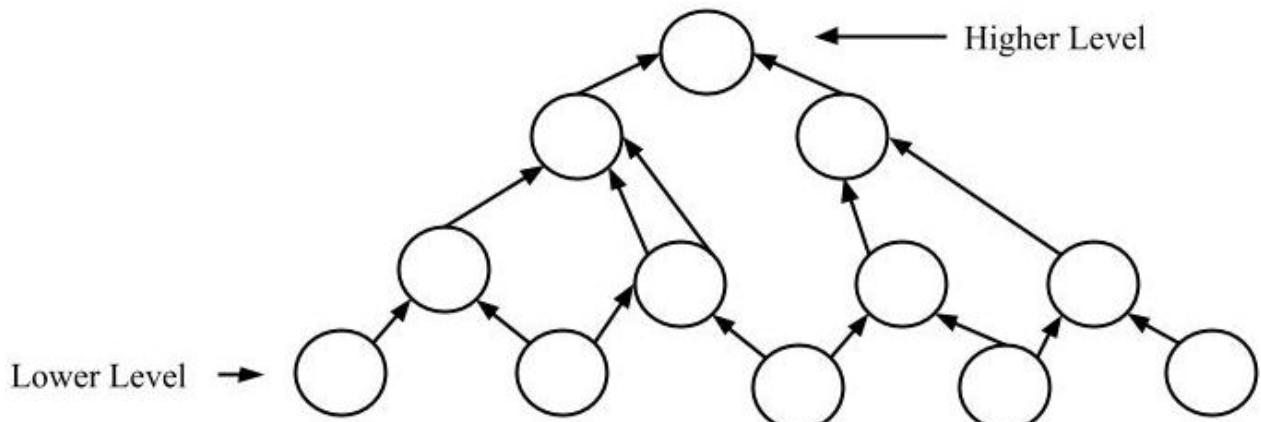
SYSTEM DESIGN AND ANALYSIS

Overview

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. System design is really important step. With proper design we can make a software which can achieve its goals and needs. While if we don't correctly make the system design, then it could have adverse effects, the result could be loss of efficiency because we didn't plan the architecture correctly, or full breakdown, data leakage, or even time wastage in development process itself because we the developer don't have clear or correct idea of what we are supposed to build and how.

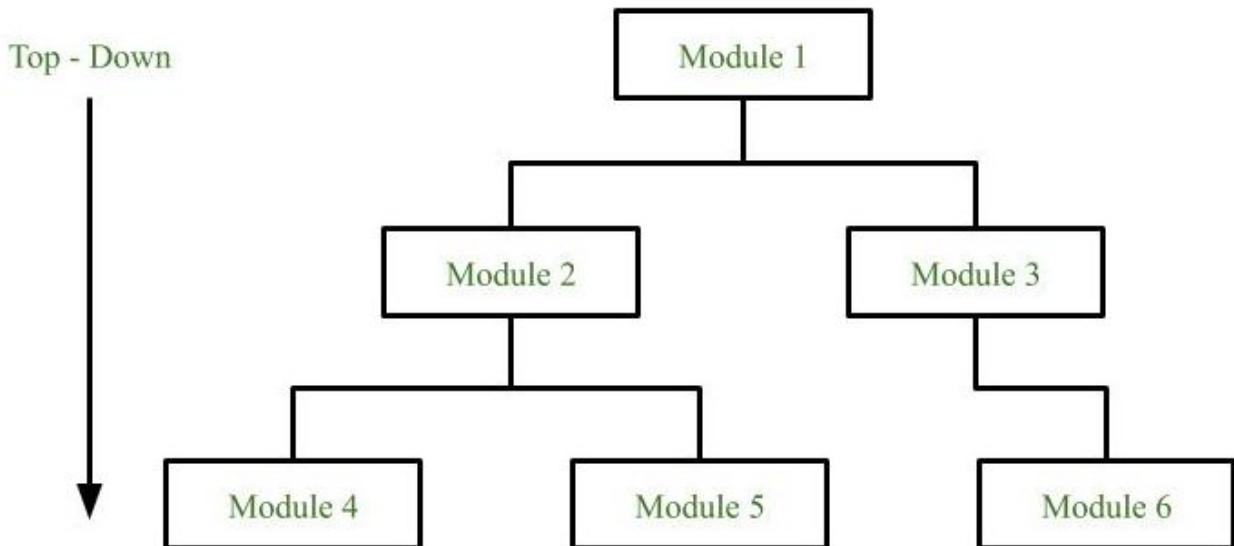
There are 2 approaches

- A. Bottom-up approach:



Music streaming from a web-server to an Android app and Website.

B. Top-down approach:



C. Hybrid Design

No Diagram available.

Each approach has its advantages and disadvantages.

Some tools for System analysis and design

- Data Flow Diagram
- Structure Charts
- HIPO Diagram
- Pseudo-Code
- Entity-Relationship Model

Further for system design and analysis we use UML

UML, an Overview

UML Stands for Unified Modelling Language. It is a general purpose, developmental, modelling language. It is used in Software industry, for development. It is intended to provide a standard way to visualise design of the system. It isn't only limited to Software industry, it is also used by banking, finance services, transportation, retail, enterprise information systems.

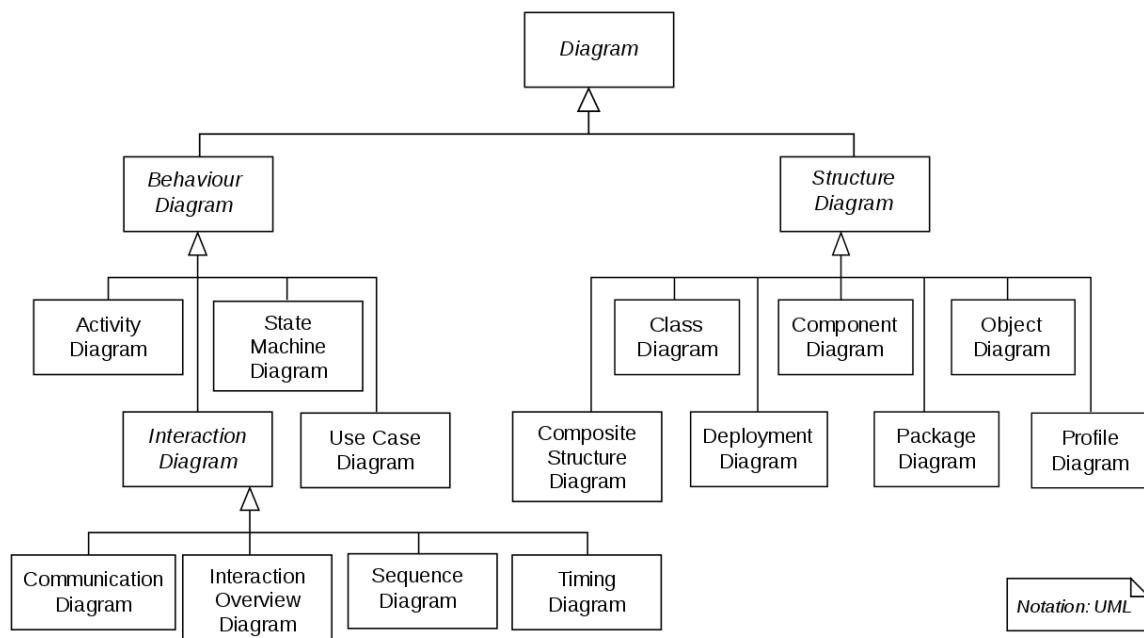
Music streaming from a web-server to an Android app and Website.

UML 2 has many types of diagrams, which are divided into two categories. Some types represent structural information, and the rest represent general types of behaviour, including a few that represent different aspects of interactions.

List of few UML Diagrams

Structural UML diagrams	Behavioural UML diagrams
Class diagram	Activity diagram
Component diagram	Communication diagram
Composite structure diagram	Interaction overview diagram
Deployment diagram	Sequence diagram
Object diagram	State diagram
Package diagram	Timing diagram
Profile diagram	Use case diagram

Representation of all the UML diagrams shown as per their categories

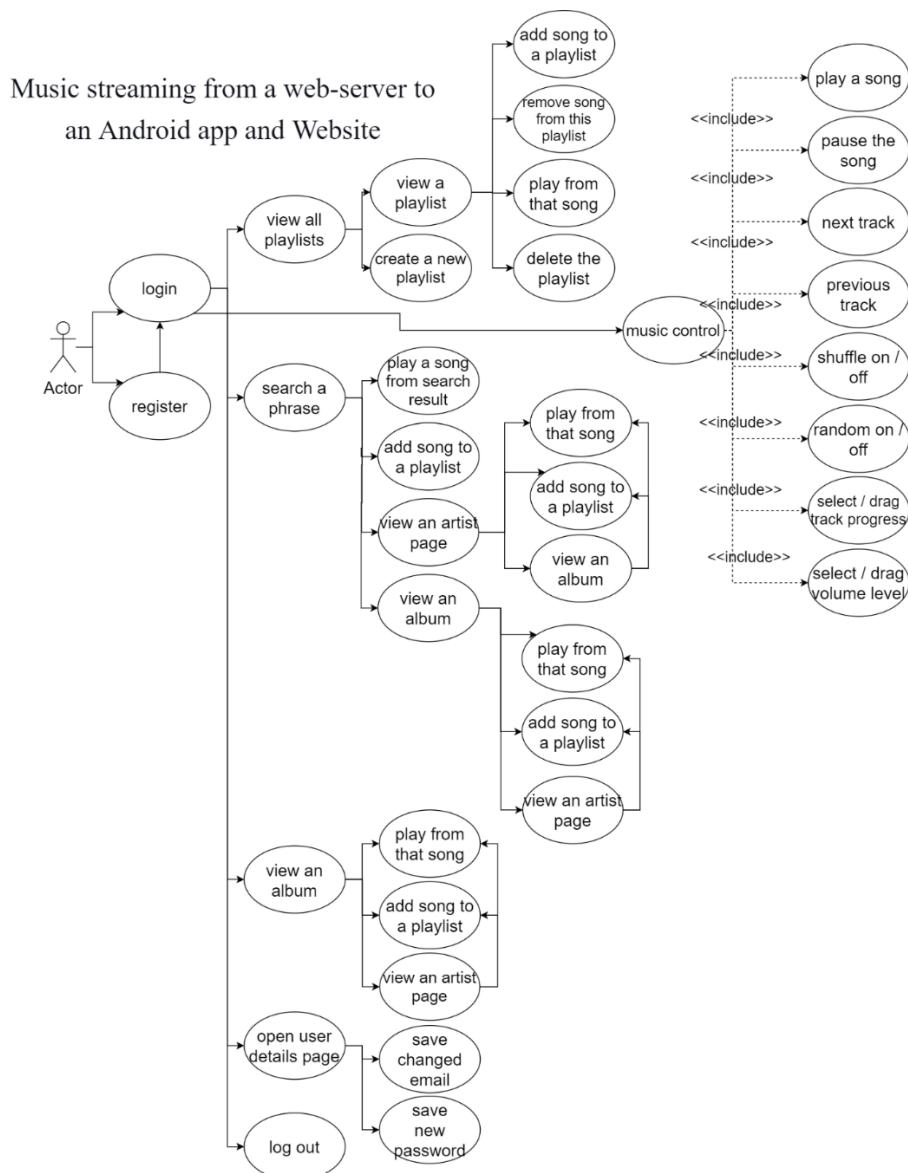


Now we'll see several diagrams for our project, they represent how our system is designed, built, structured and how each components / modules interacts with each other.

A. Use case diagram

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.

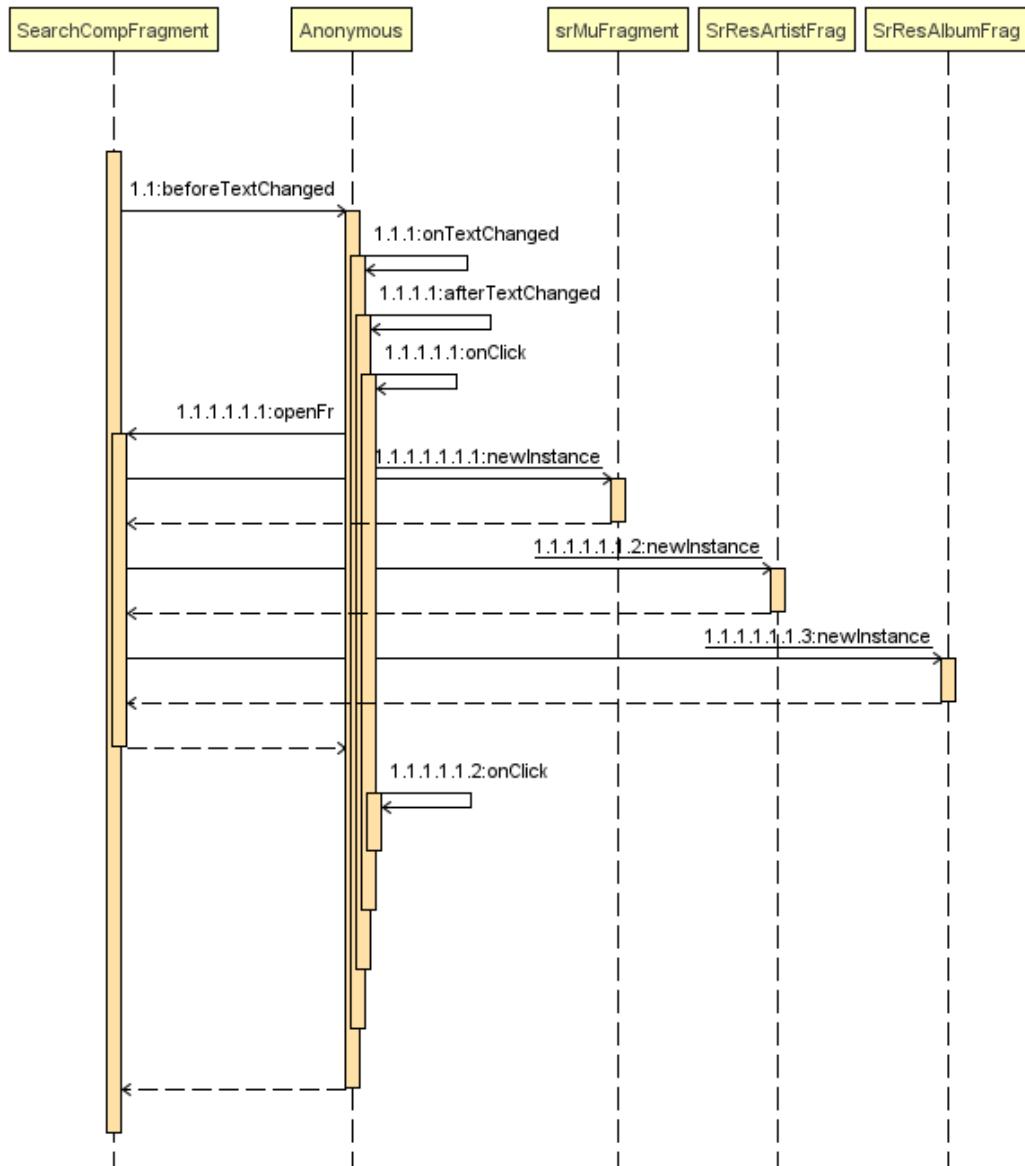
Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified.



Use Case Diagram for our project

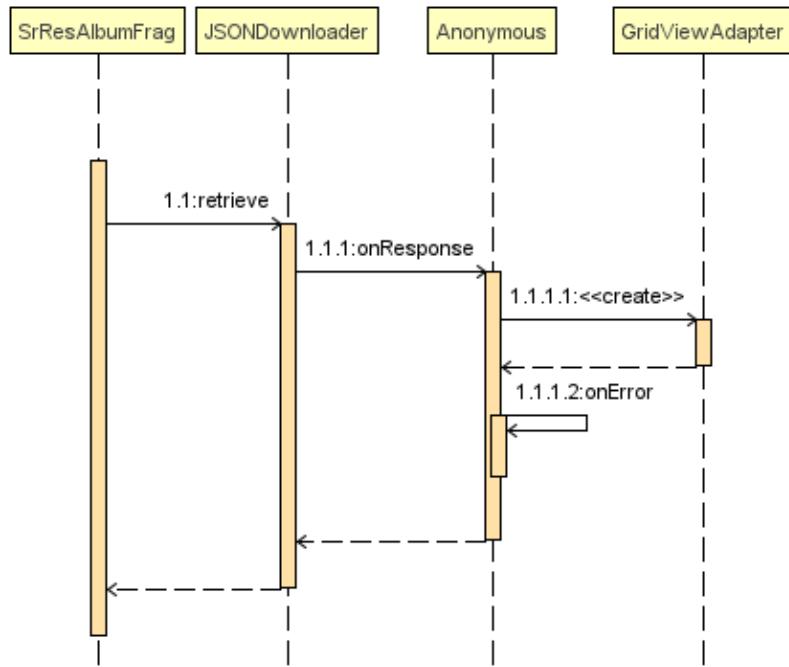
B. Sequence diagram

We have inserted only a few sequence diagrams (All are from Android app project)

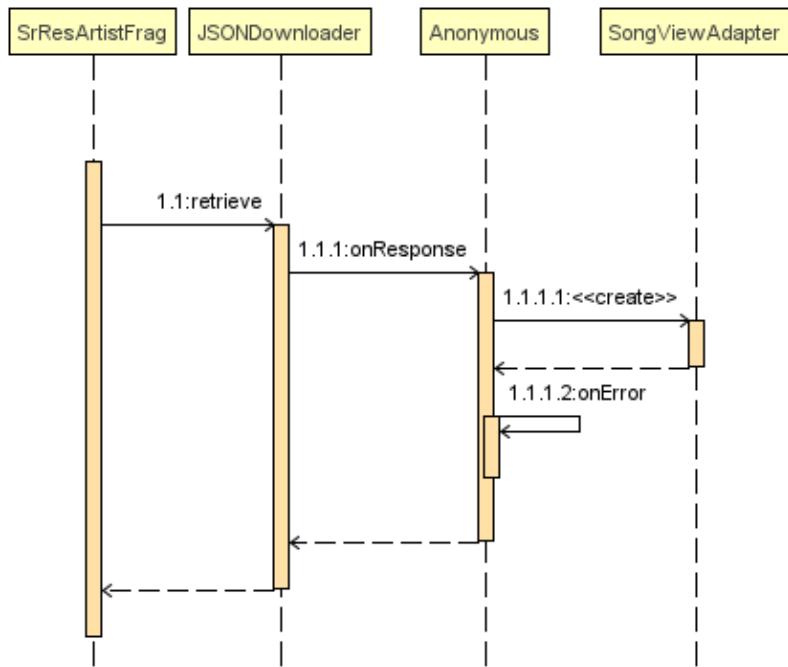


- This is Sequence diagram of `SearchCompFragment` class. It inflates the layout of Search Bar. Essentially this is only the search bar.
- We pass input which is search string to ‘openFr’ when the search button is pressed. This search input can be empty too.
- Then ‘openFr’ creates newInstances of ‘`srMuFragment`’, ‘`SrResArtistFrag`’, ‘`SrResAlbumFrag`’. And we pass the search phrase to them.
- ‘`srMuFragment`’ is to display the list of songs when they match to the searh string.
- If empty string is passed then our php file returns All the entries from the table.

Music streaming from a web-server to an Android app and Website.

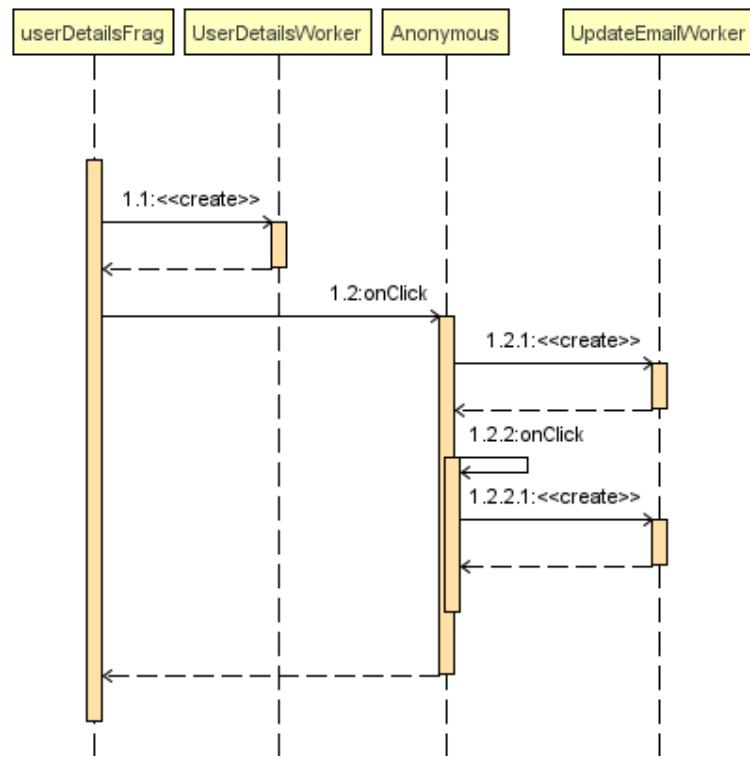


- This is Sequence diagram of SrResAlbumFrag class. It displays Search Result for search query for category Album.
- It uses JSONDownloader and then passes that data to Grid View Adapter.

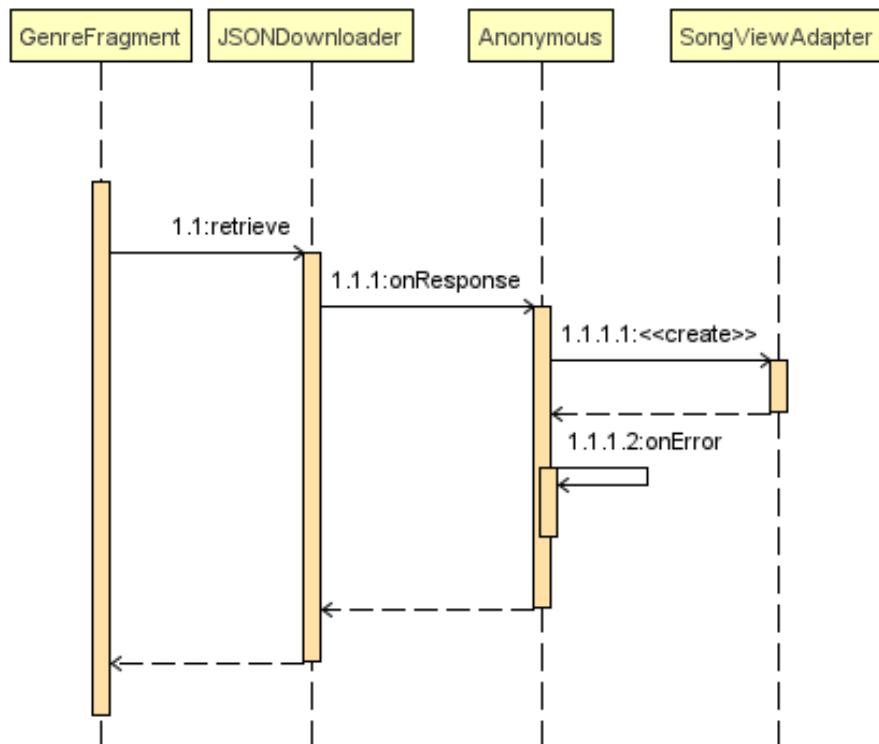


- This is Sequence diagram of SrResArtistFrag class. It displays Search Result for search query for category Artist.
- It uses JSONDownloader and then passes that data to List View Adapter (SongViewAdapter is what we have named it).

Music streaming from a web-server to an Android app and Website.

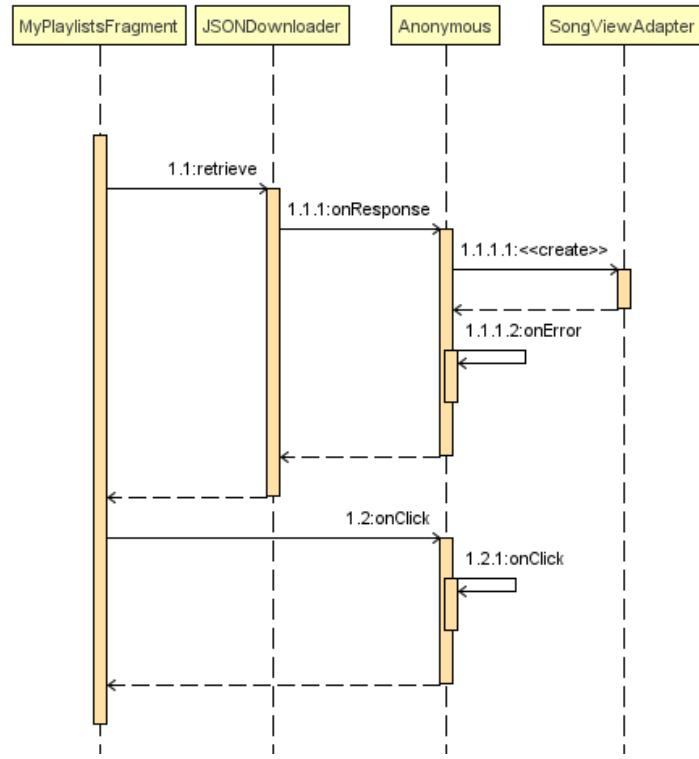


- This is Sequence diagram of userDetailsFrag class. It is used to update the Email OR Password.

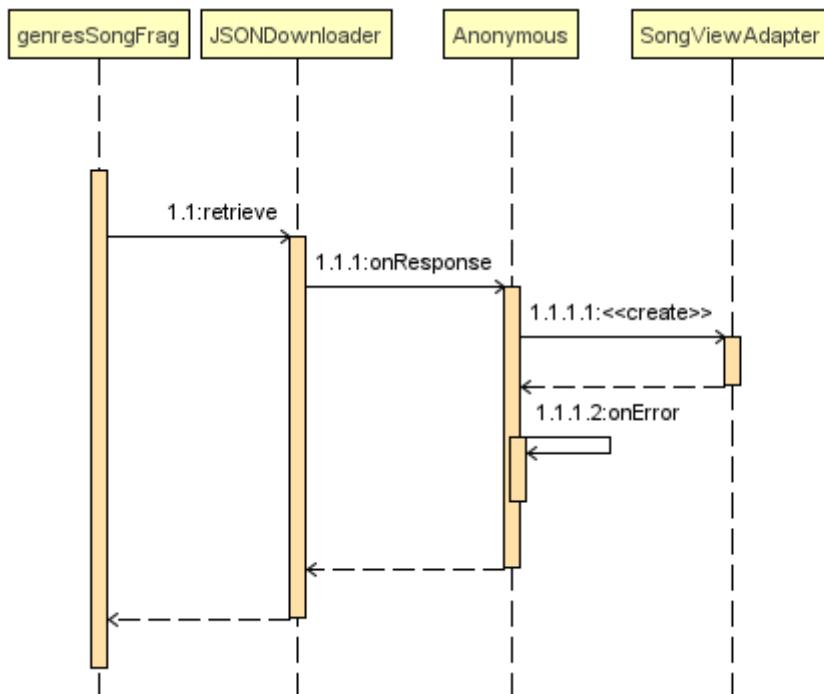


- This is Sequence diagram of GenreFragment class. It is used to download the list of genres via JSONDownloader, and then it passes data to SongViewAdapter to be displayed.

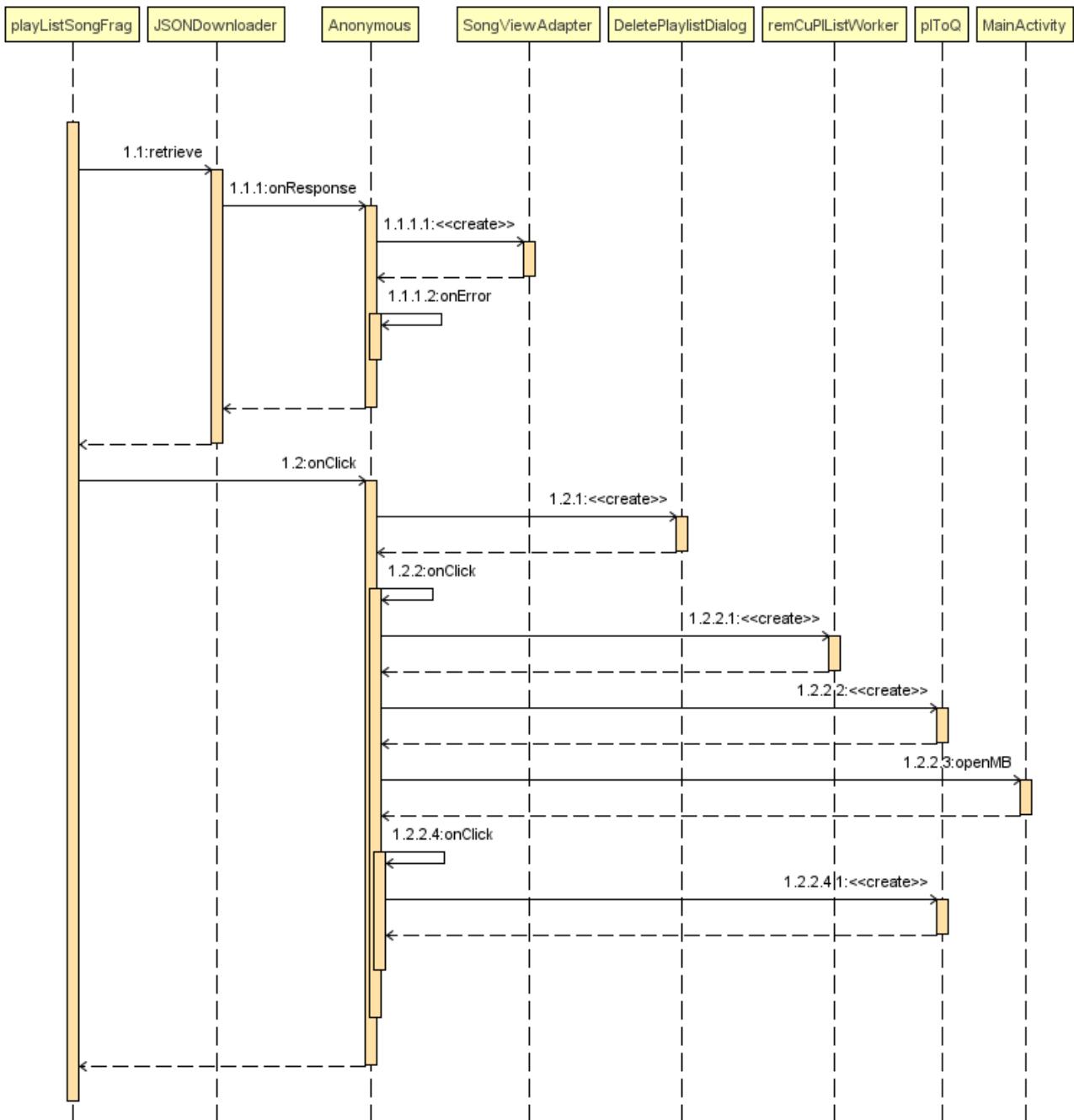
Music streaming from a web-server to an Android app and Website.



- This is Sequence diagram of MyPlaylistsFragment class. It is used to download the list of playlist for the logged in user via JSONDownloader, and then it passes data to SongViewAdapter to be displayed.
- Here we also give option for user to reload their list of playlist. (needed after creating new.)

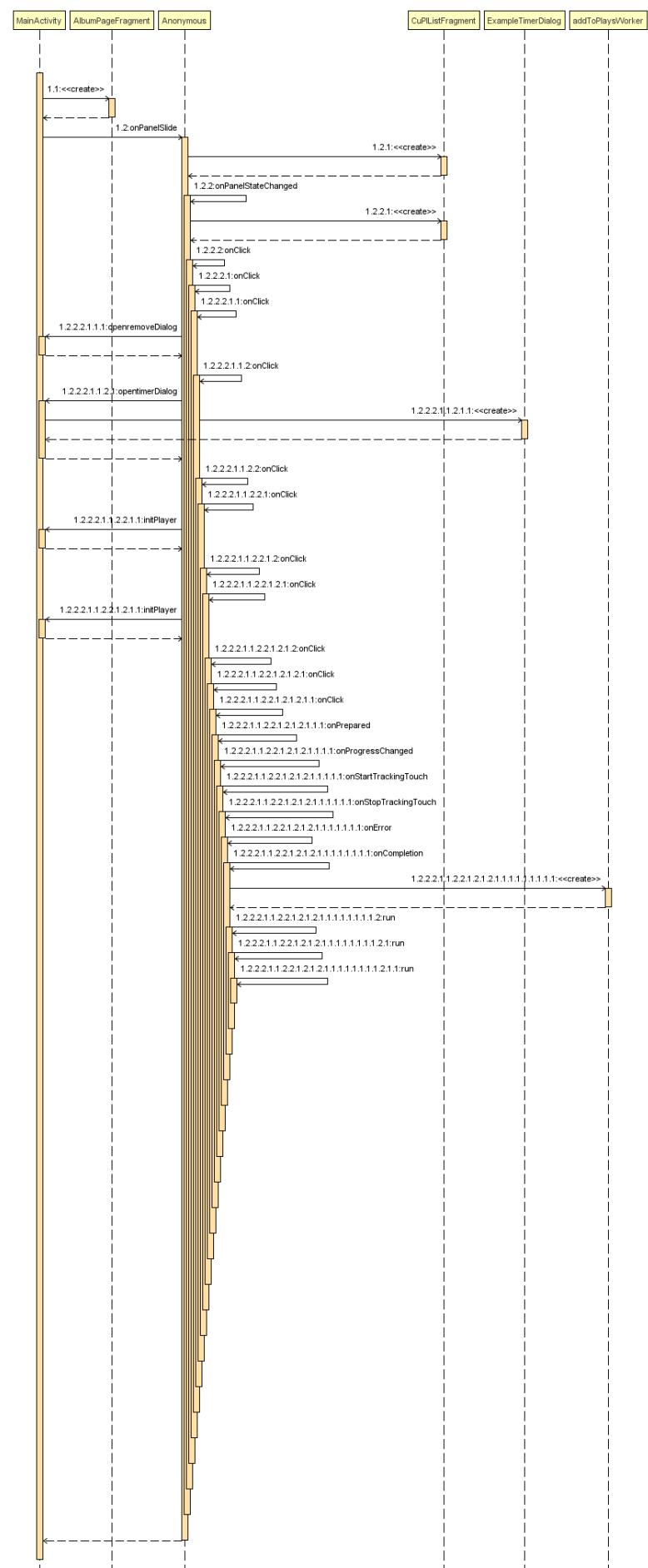


Music streaming from a web-server to an Android app and Website.



- This class retrieves list of songs in the selected Genre.
- This class retrieves list of songs in the selected Playlist.
- However we give options to the users to delete the playlist (DeletePlaylistDialog), or add that playlist to the current playing list (plToQ). Those options can be seen.
- DeletePlaylistDialog is a alert dialog box, inside user may select to delete or not to.
- remCuPlListWorker is an option, through which user can empty the current playing list. User can't select this option on their own from this fragment class. This used when user presses on any song. Then the current playing list is replaced by the selected playlist.

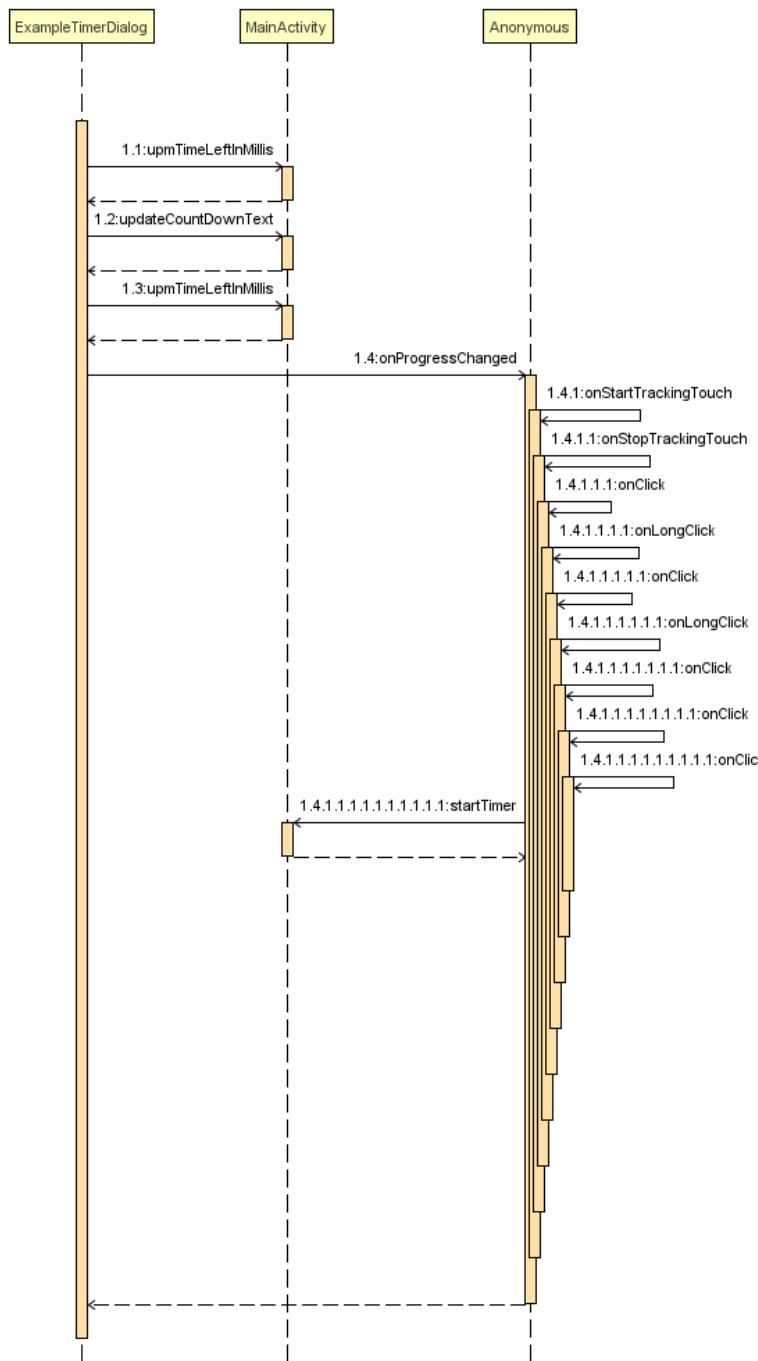
Music streaming from a web-server to an Android app and Website.



This is the
Sequence Diagram
for MainActivity.

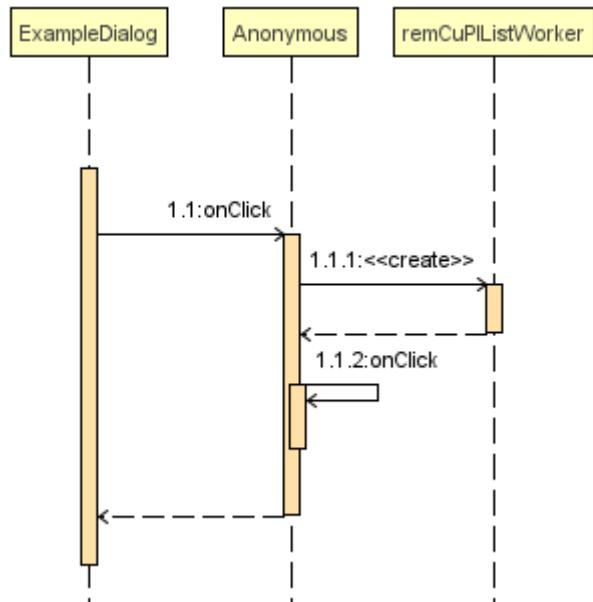
It shows how every other class is interacted by and from, the MainActivity.

Music streaming from a web-server to an Android app and Website.

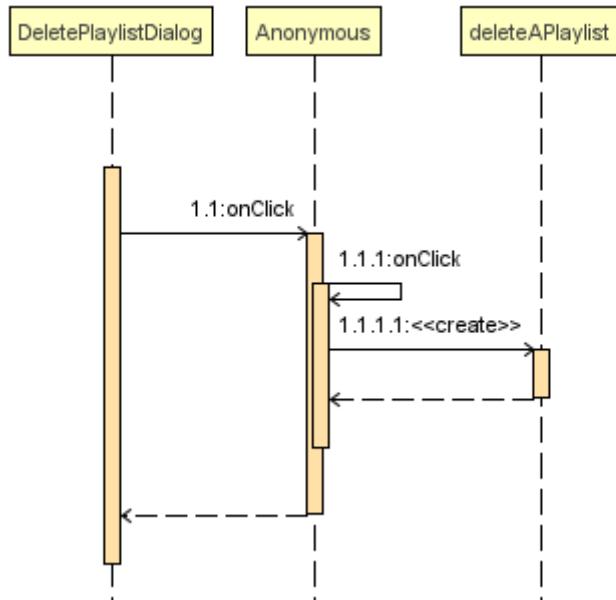


- This class gives option for user to select a predecided time, in minutes, after which the music will stop.
- It creates a Dialog Box, inside of which we have a seek bar, where user can select minutes. As can be seen, ExampleTimerDialog class passes and receives time with use of method ‘upmTimeLeftInMillis’ which is defined in MainActivity.
- ‘updateCountDownText’ is method with which we update the remaining time in the Dialog Box.
- We use ‘onStopTrackingTouch’ (belongs to SeekBar) to get the value of the seek thumb, we display it inside the Dialog box.
- ‘startTimer()’ is initiated only when the Positive button of the Dialog box is pressed.

Music streaming from a web-server to an Android app and Website.

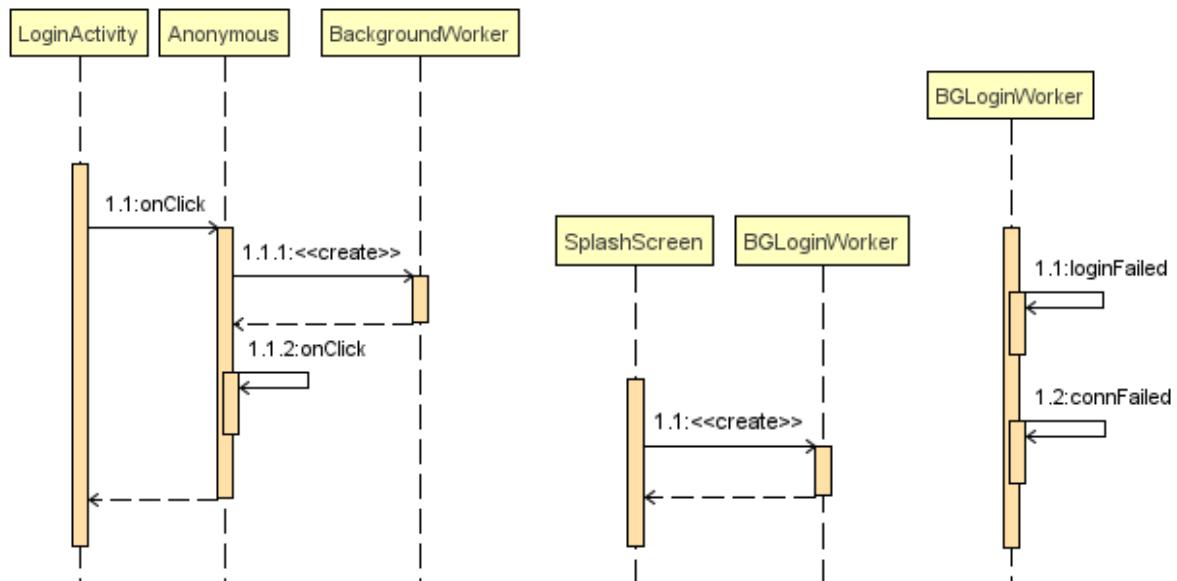


- With this class we let users empty their current playing list. This class creates a Dialog Box, when the positive button is pressed, ‘remCuPIListWorker’ is run and it access a php file though which the table entrieys are removed.

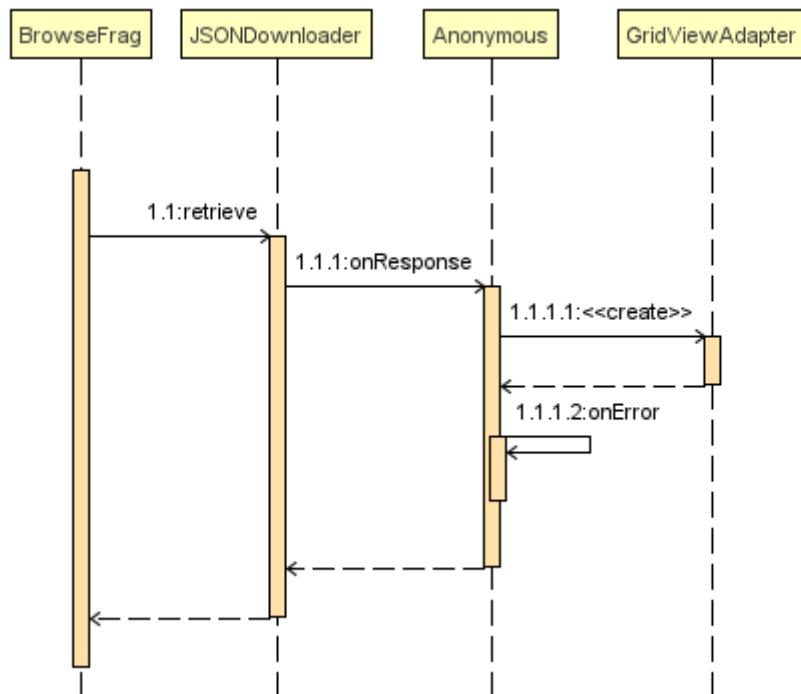


- With this class users can delete a playlist that they have created. This class creates a Dialog Box, when the positive button is pressed, we pass the playlistId to ‘deleteAPlaylist’ and it passes that data to php. Along with this, we also take user one screen back as it would attract issues if user is allowed to see a playlist which we have deleted on server, yet it is visible on the device. As any opion might trigger issues.

Music streaming from a web-server to an Android app and Website.

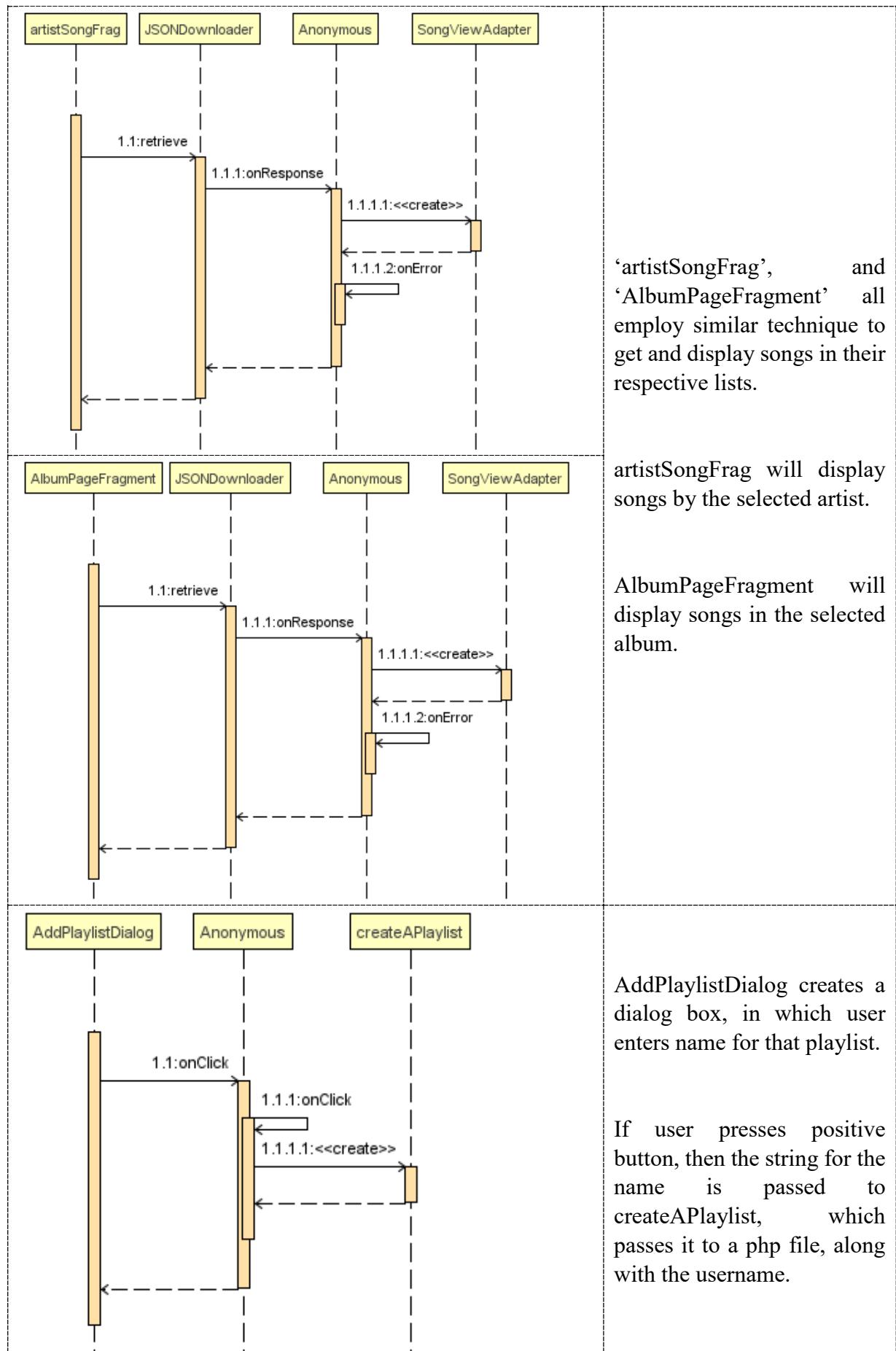


- Explained, these three classes in the class diagram section.



- BrowseFrag, is a fragment class, it is supposed to display available albums.
- It uses JSONDownloader to download the list, store it in ArrayList of custom class, then pass it to GridViewAdapter to populate.
- We use GridView for Albums, because Albums have album art.

Music streaming from a web-server to an Android app and Website.



'`artistSongFrag`', and '`AlbumPageFragment`' all employ similar technique to get and display songs in their respective lists.

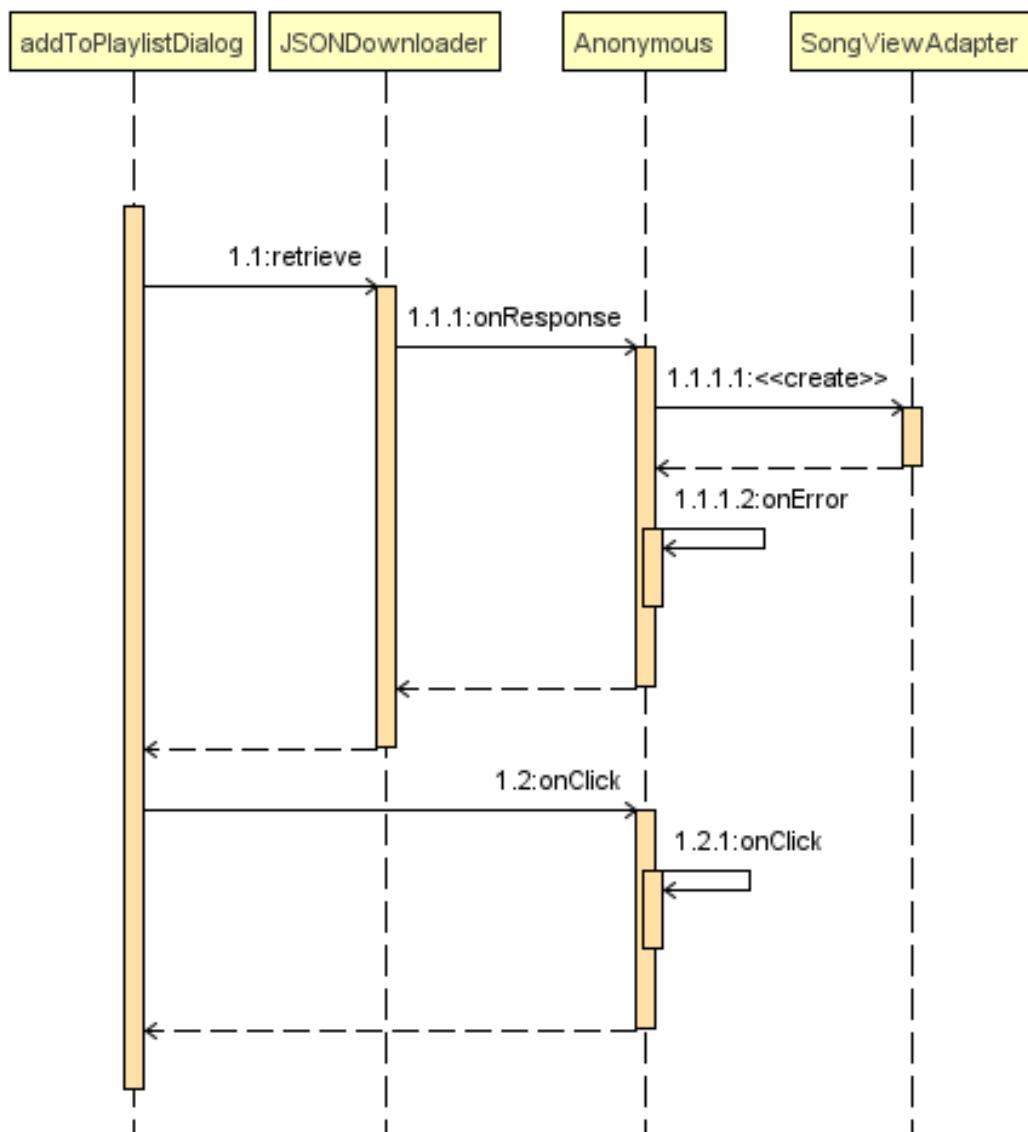
`artistSongFrag` will display songs by the selected artist.

`AlbumPageFragment` will display songs in the selected album.

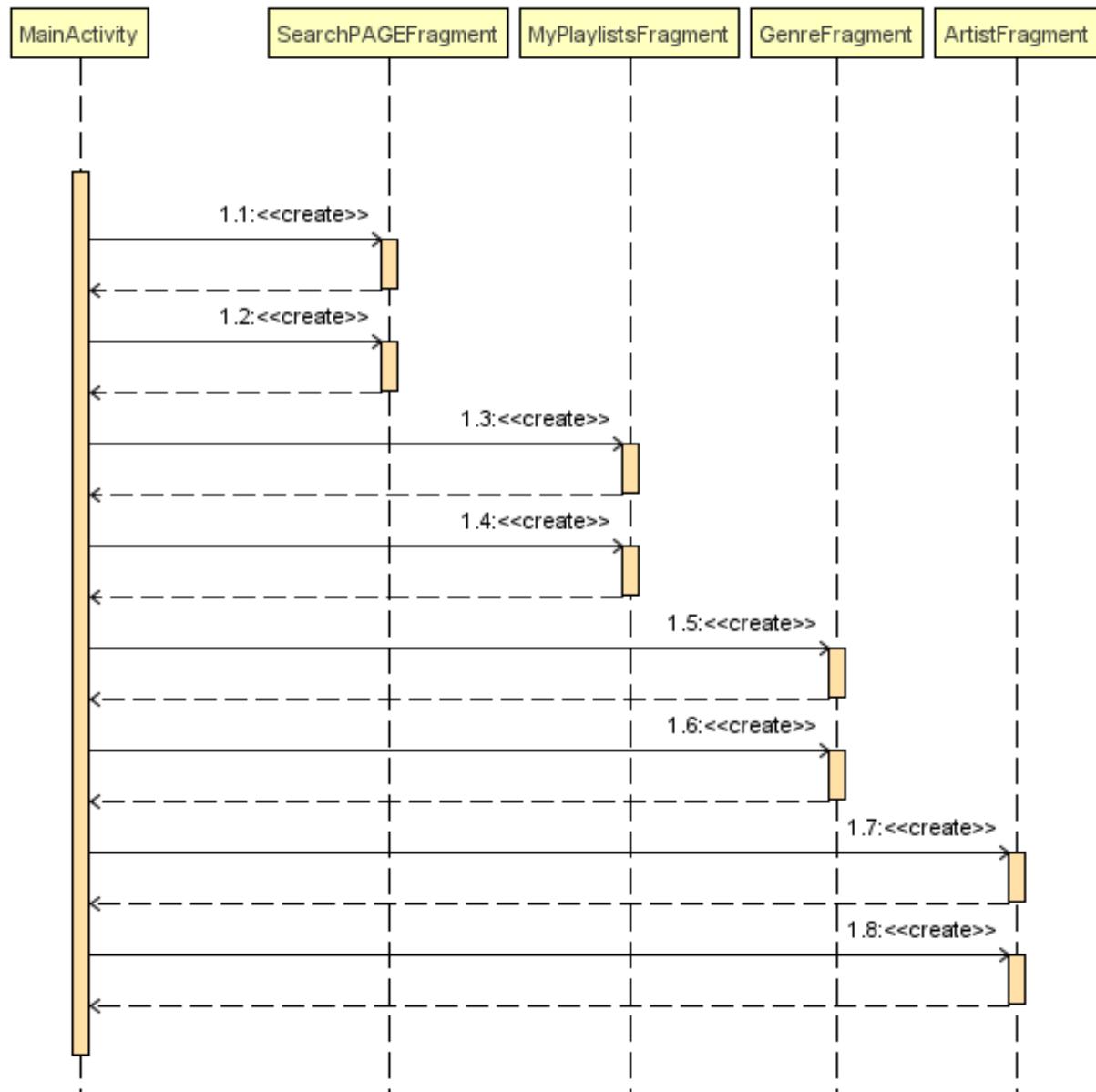
`AddPlaylistDialog` creates a dialog box, in which user enters name for that playlist.

If user presses positive button, then the string for the name is passed to `createAPlaylist`, which passes it to a php file, along with the username.

Music streaming from a web-server to an Android app and Website.

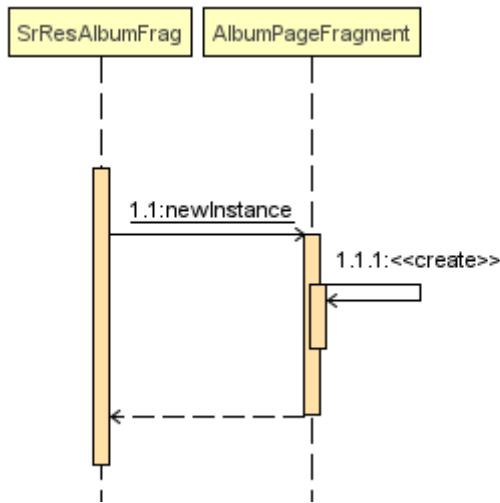


- `AddToPlaylistsFragment` gives user a list of playlists which are created by the user. This is triggered when a user wishes to add a song into a playlist.
- For that User has to press and hold the song name and options will appear.

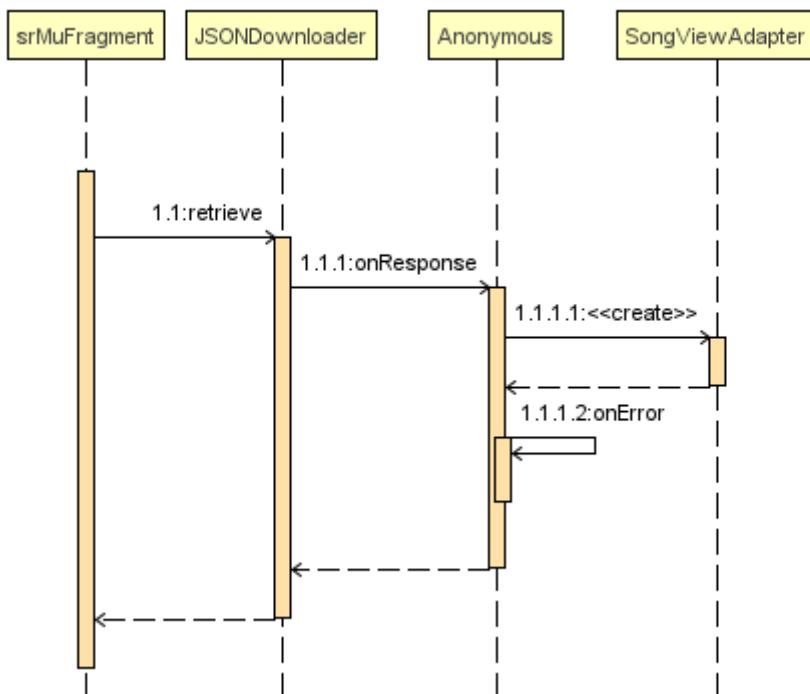


- This is sequence diagram of how Navigation Menu works.
- Navigation menu is part of MainActivity.
- When user presses Search section it takes user to 'SearchPAGEFragment'.
- When user presses Playlist section it takes user to 'MyPlaylistsFragment'.
- When user presses Genres section it takes user to 'GenresFragment'.
- When user presses Artist section it takes user to 'ArtistFragment'.
- Important to note that, when the app initiates MainActivity we are in Browse section. In the MainActivity we have a frame layout. And we populate 'BrowseFrag' in it.

Music streaming from a web-server to an Android app and Website.



- ‘SrResAlbumFrag’ display list of album which match the search phrase. And when user selects one album it launches newInstance for ‘AlbumPageFragment’.



- ‘SrMuFragment’ display list of songs which match the search phrase.
- Just like all the other Activities and Fragment Classes which get and display contents from our backend. ‘SrMuFragment’ also uses JSONDownloader to get the contents from php and then passes it to SongViewAdapter to populate the layout.

C. Database Tables in graphical diagram

These are all the tables in our database.

songs	albums	genres	artists
id	id	id	id
title	title		
artist	artist		
album	genre		
genre	artworkPath		
duration			
path			
albumOrder			
plays			

currentPlayingList	users
id	id
songId	username
owner	firstName
playlistOrder	lastName

playlists	playlistSongs
id	id
name	songId
owner	playlistId
dateCreated	playlistOrder

Data Base Tables

Description:

- songs

This table has all the given information that we require about all the songs that are available in our project's offering. This table is really essential. In this we store ids for artist, genre, album which are present in their own respective tables. We also keep track of sequence of songs for their respective albums, with albumOrder. We store the duration and path which is essential for playback.

- albums

This table has information for album, like title, artist, and genre of that album. And has artworkPath, which is address to the album art.

Music streaming from a web-server to an Android app and Website.

- genres

This table only has id and the corresponding name of that genre.

- artists

This table only has id and the corresponding name of that artist.

- playlist

We store, id of the playlist, username of the creator, and the date when it was created.

- playlistSongs

In this table the songs exist which are in their respective playlists. They are correlated with playlistId column. We store the sequence of songs in that playlist.

- users

Here we store all the information about the users. Information including first, last names, email Id, username. Date of signup, their profile pic. And we store their password. Password is processed by the php to be converted from text format to md5 hash. So anyone with access to our database can't guess the password.

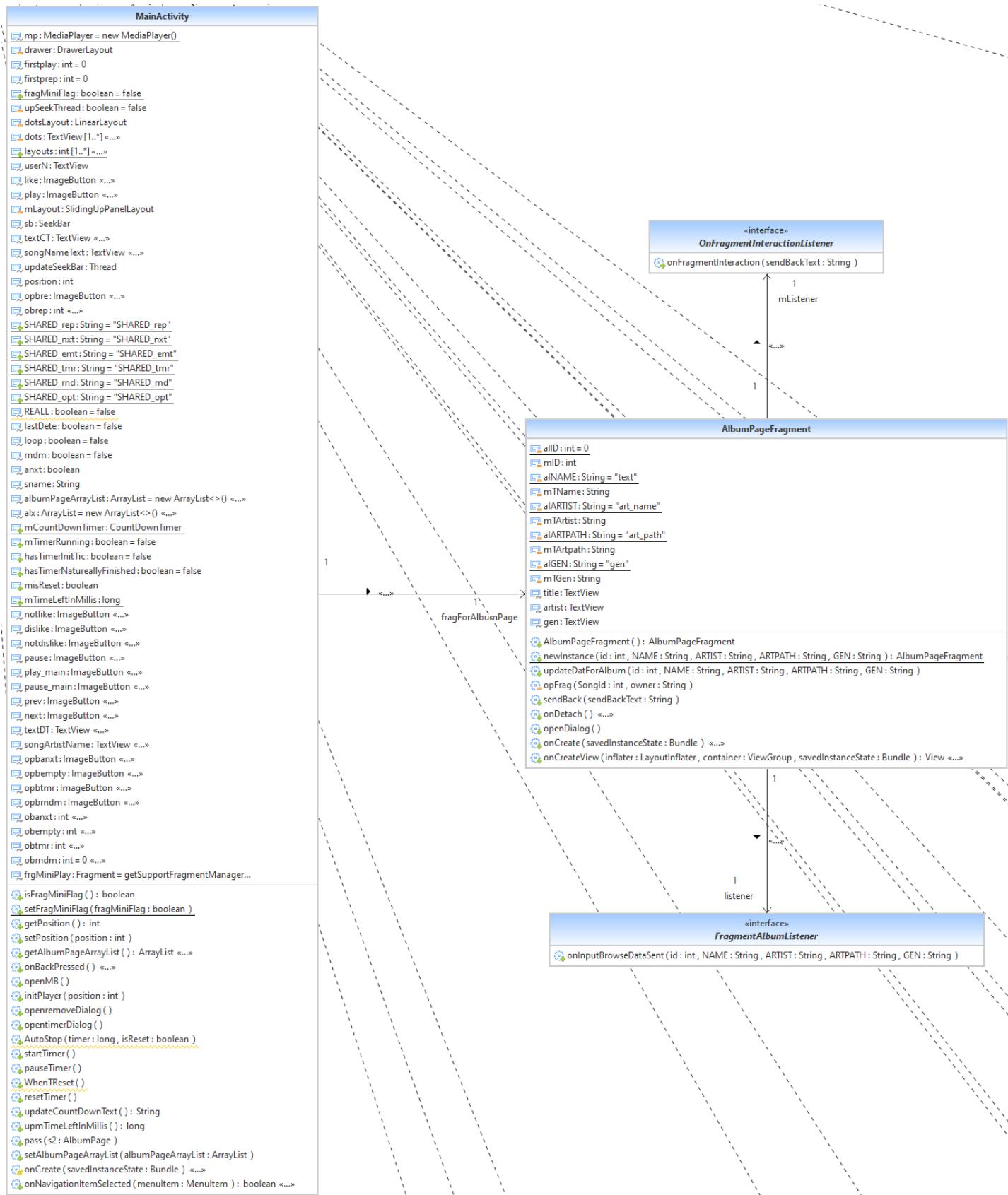
- currentPlayingList

This was added much later in the project. This is a table, in which a playlist of songs is kept, with the sequence, and owner name which is the username.

D. Class diagram

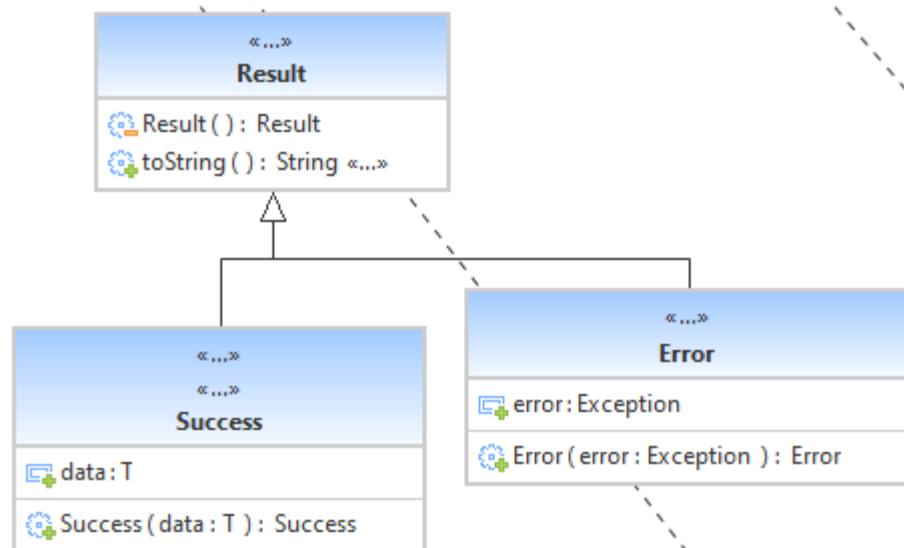
We have inserted only a few selected class diagrams (All are from Android app project).

Music streaming from a web-server to an Android app and Website.

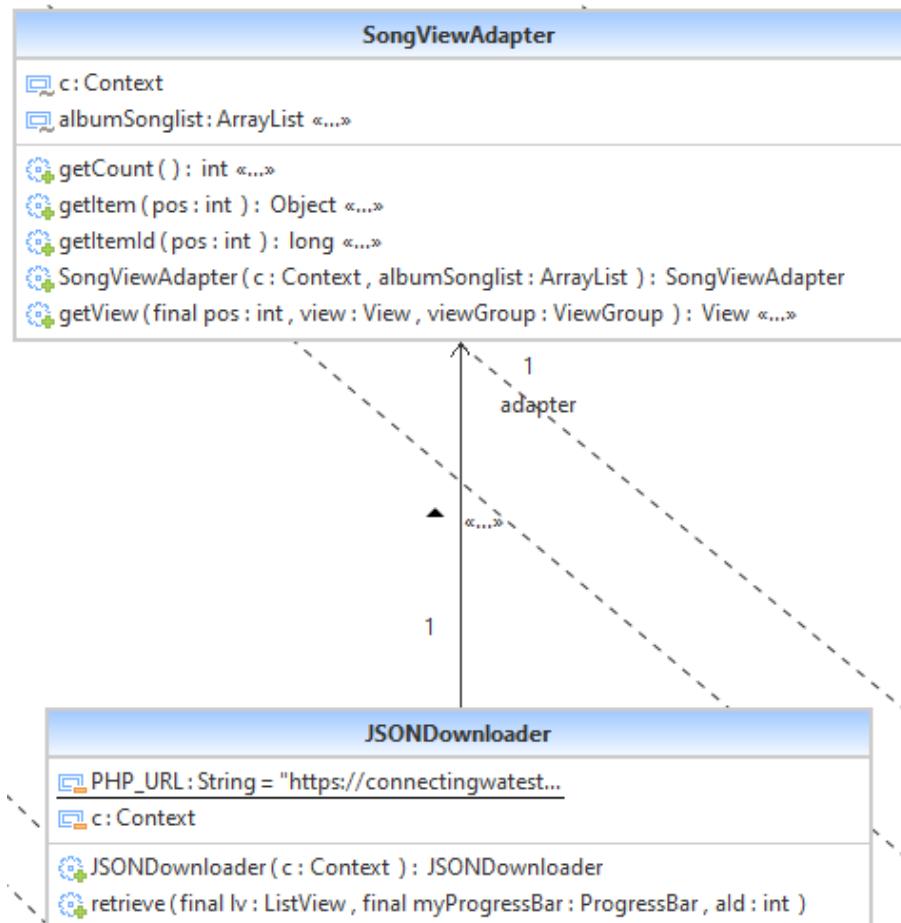


Class Diagram of MainActivity, AlbumPageFragment

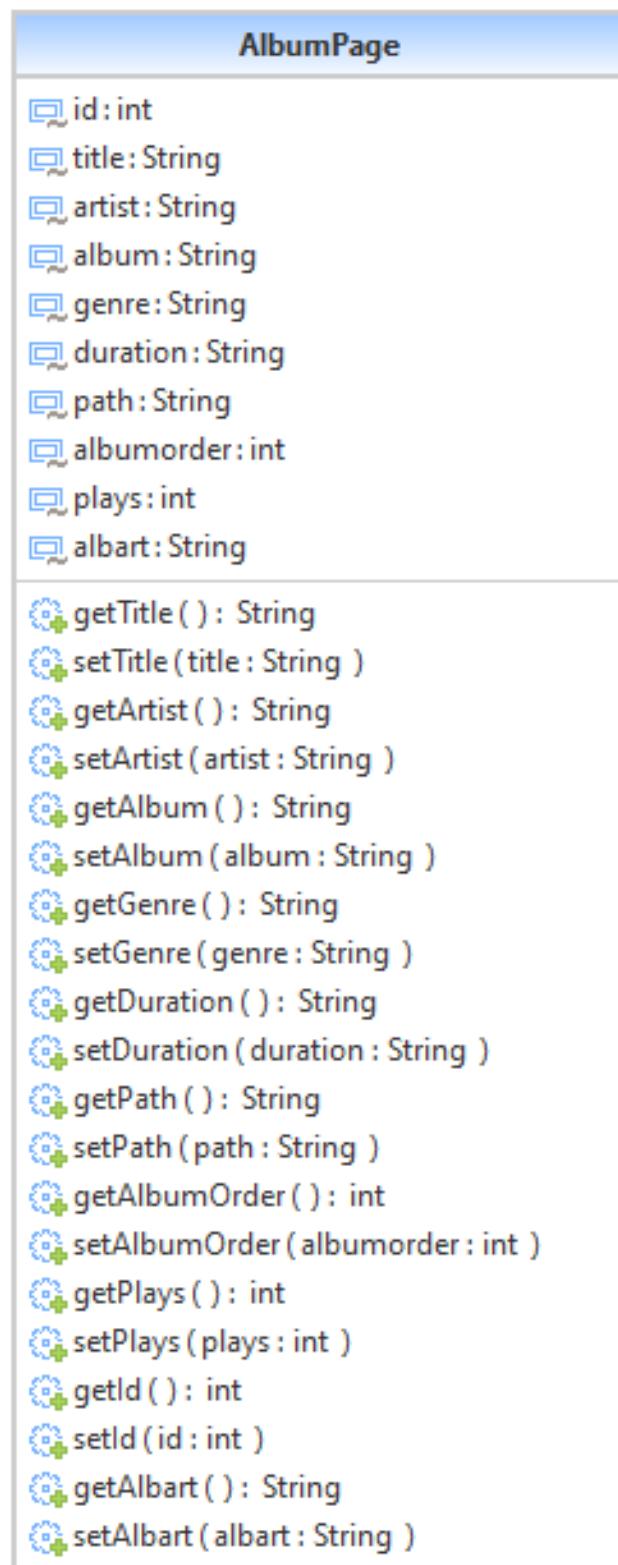
Music streaming from a web-server to an Android app and Website.



Class Diagram of Login Class.



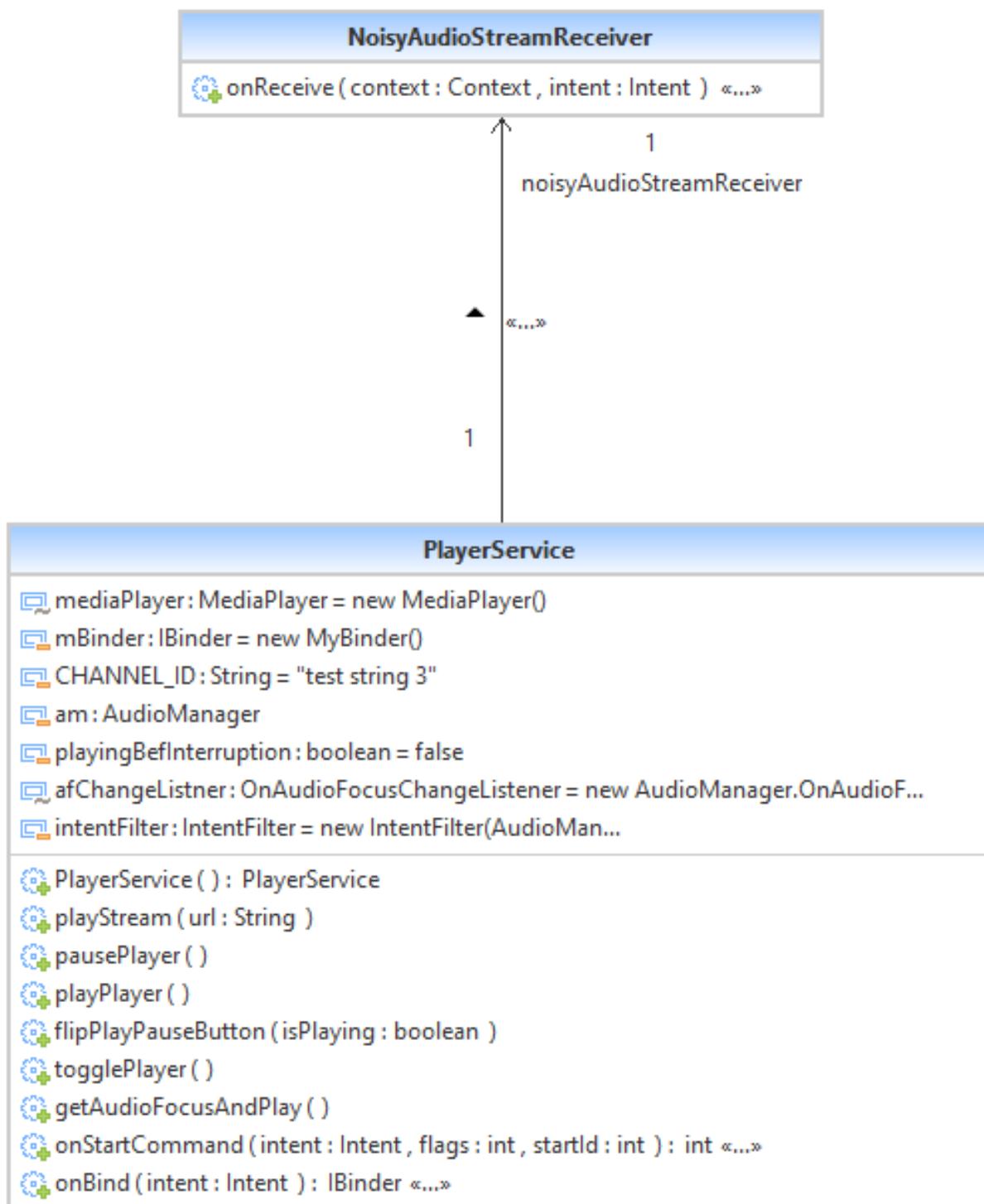
An Example of class diagram of many activity & fragment classes which use **JSONDownloader** to retrieve data. The diagram shows how **JSONDownloader** passes data to **SongViewAdapter** (in some cases **GridViewAdapter**). Then it populates the List view or Grid view.



Class Diagram of AlbumPage.

AlbumPage is an activity inside of which we display all the songs, which are contents of the album which was clicked / opened.

Music streaming from a web-server to an Android app and Website.

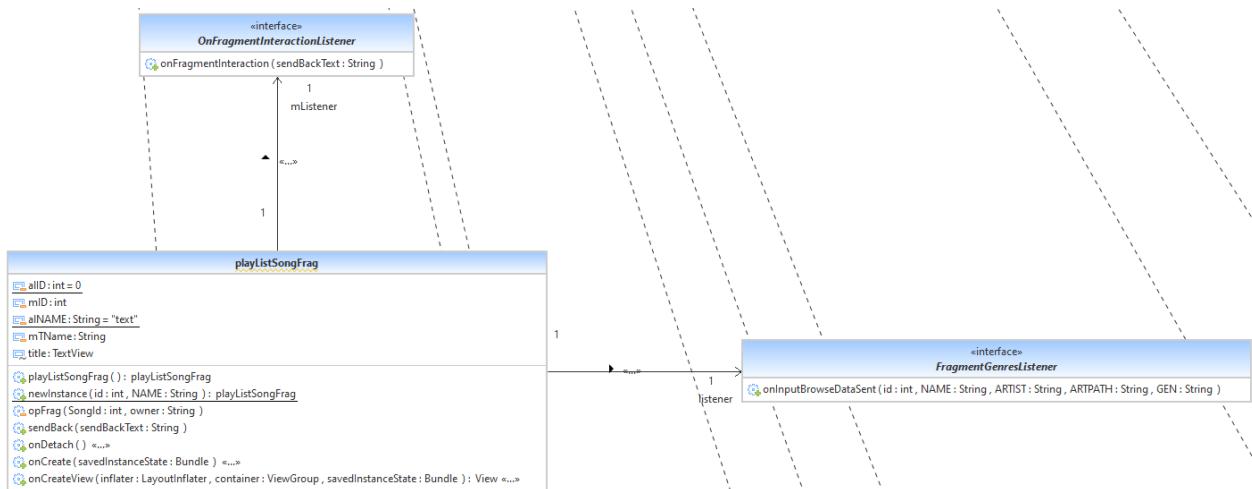


This is class diagram of PlayerService.

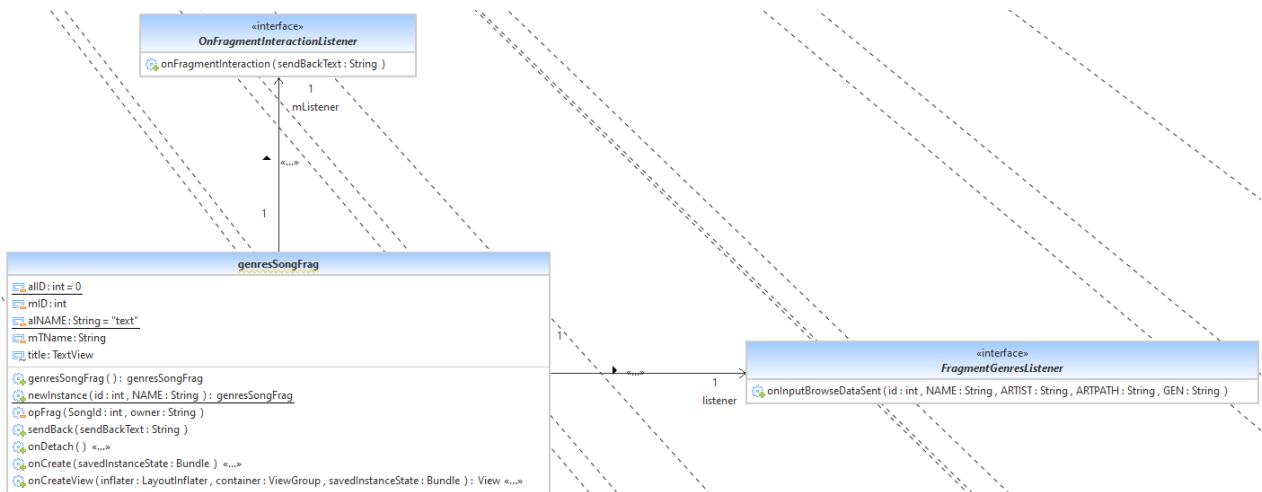
MediaPlayer class is the primary API for playing sound and video in Android. It is capable of playing: local resources, internal URI (Device based files), external URL from internet. For internet media, the app that we make will require permission declaration to access internet.

We only use and invoke MediaPlayer class.

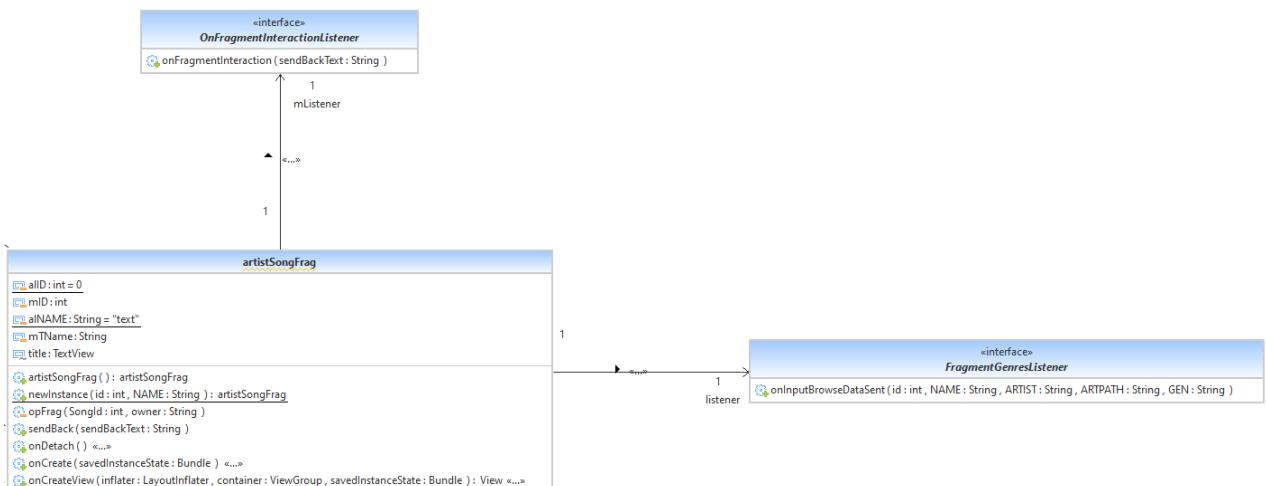
Music streaming from a web-server to an Android app and Website.



Class diagram of `playListSongFrag`. It is a fragment class.



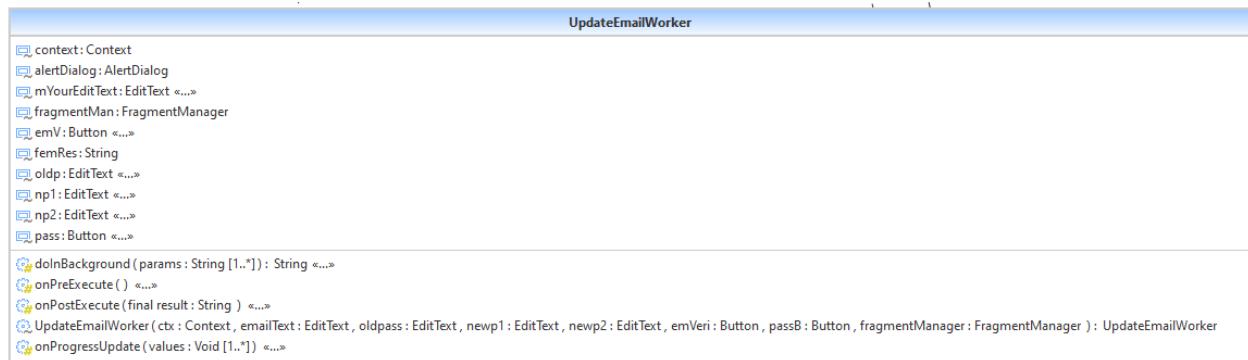
Class diagram of `genresSongFrag`. It is a fragment class.



Class diagram of `artistSongFrag`. It is a fragment class.

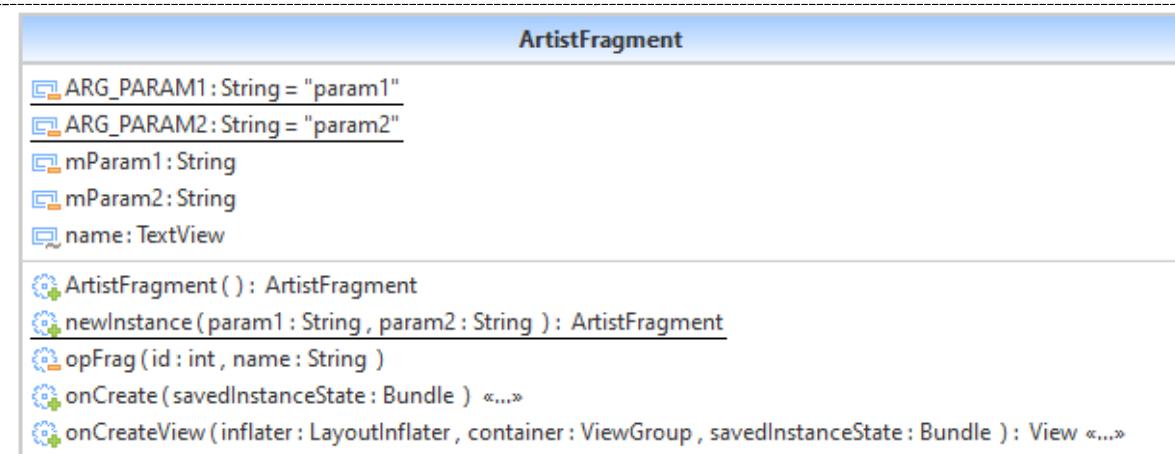
In all of these classes seen above, we are initiating them when any item is clicked in the previous fragment. For a playlist it is name of the playlist, for an artist, it is name of the artist. When that is clicked it passes id and name `newInstance()` is initiated for each of these classes.

Music streaming from a web-server to an Android app and Website.

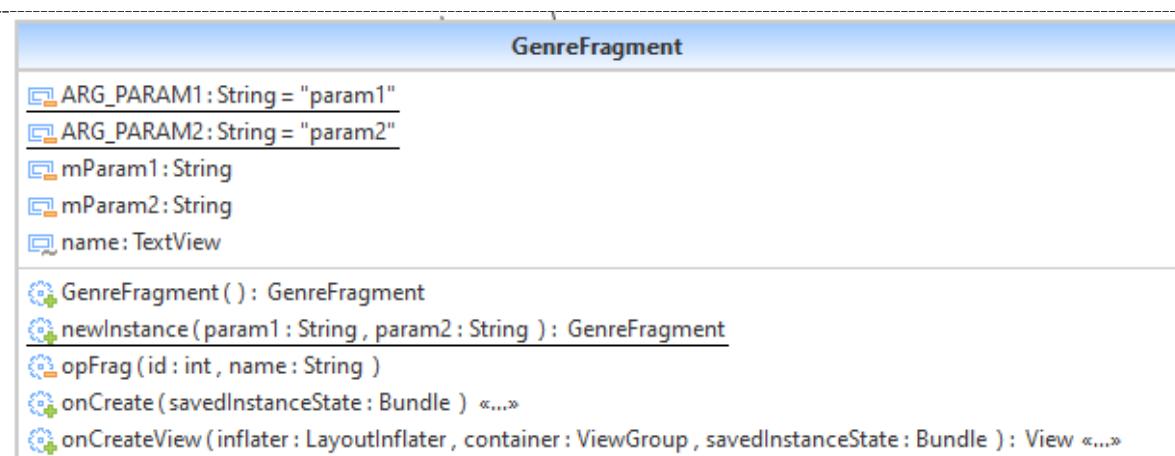


Class diagram for `UpdateEmailWorker`

This class connects to and updates Either Email or Password. It needs username. When email wishes to change the Email id, it passes that to the .php file and then returns the text results. And in case of Password, it passes current password, and new password, it can do so only if both new passwords are a match. It passes old and new password to .php files and returns result.

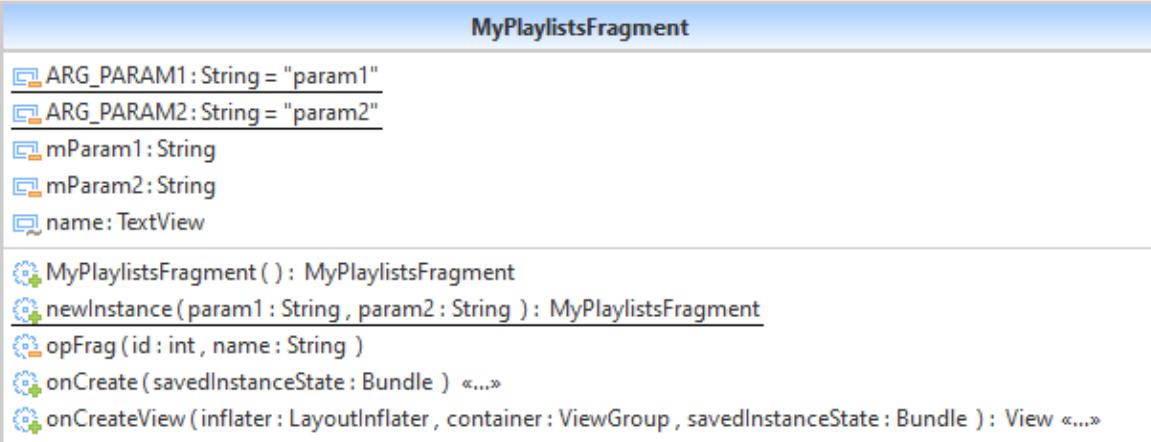


This is a Fragment class which displays list of Artists we have in our database.

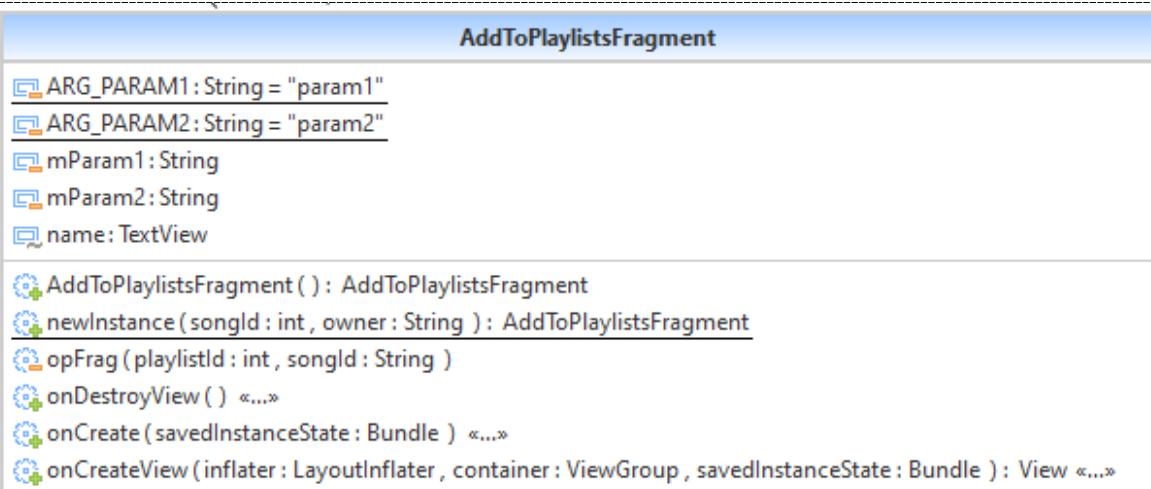


This is a Fragment class which displays list of Genres we have in our database.

Music streaming from a web-server to an Android app and Website.



Passing username to .php page, returns list of Playlists made by that user.

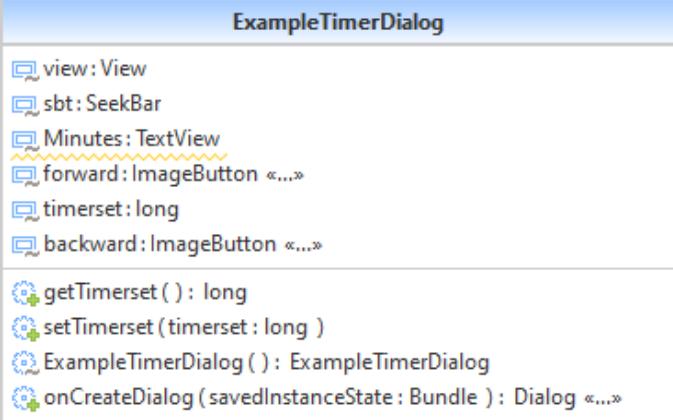


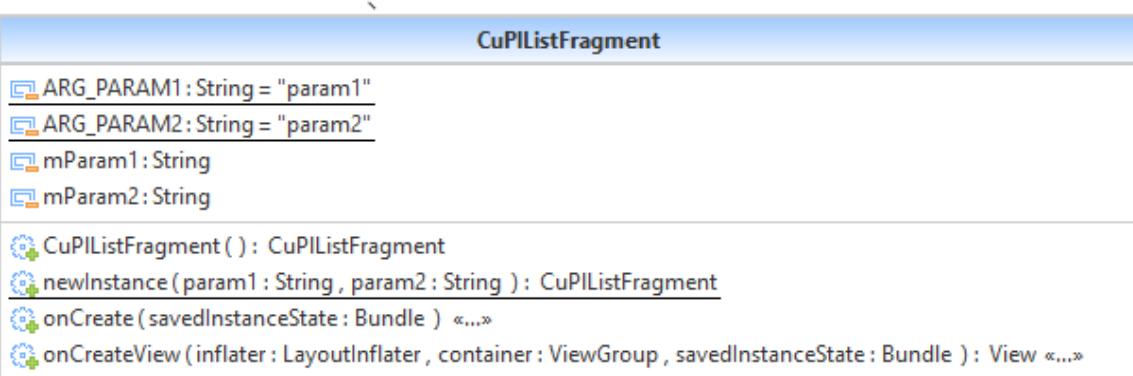
AddToPlaylistsFragment receives songId and username.

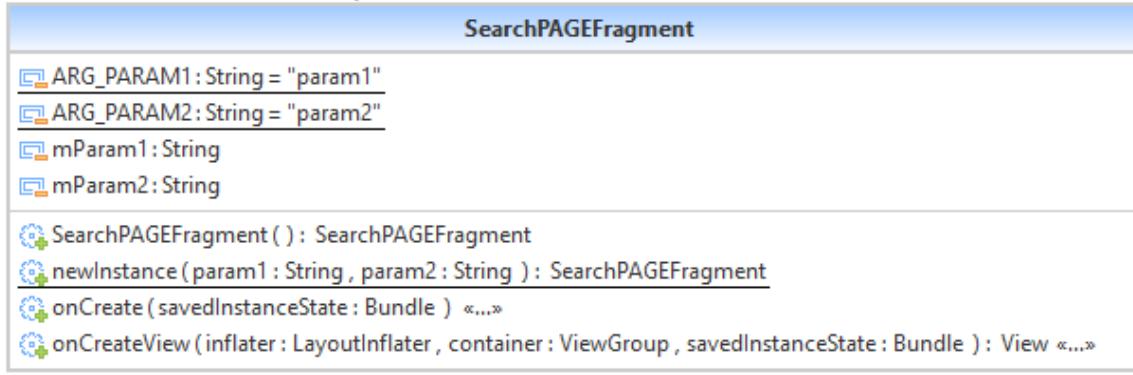
Then it retrieves the available playlists made by that user, if any.

Then it passes the songId and playlistId which user selected to ‘AddToPlayListWorker’

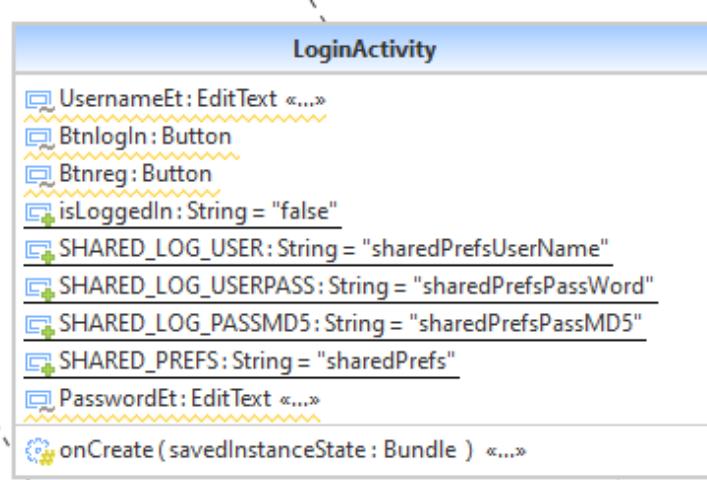
Music streaming from a web-server to an Android app and Website.

	<p>This class creates a dialog box, Inside of which user can set time (after which the media player will be stopped). The value is set in minutes, then it is passed to MainActivity.</p>
---	---

	<p>In class CuPIListFrag, this uses JSONDownloader to get the Current Playing list from our database for our user. When user slides up the music player slider, inside of this it is displayed.</p>
---	---

	<p>SearchPAGEFragment class, we only replace a FrameLayout and start the search bar fragment in it. After the user presses search button, Other FrameLayouts of this class are populated with results for our search.</p>
--	---

Music streaming from a web-server to an Android app and Website.

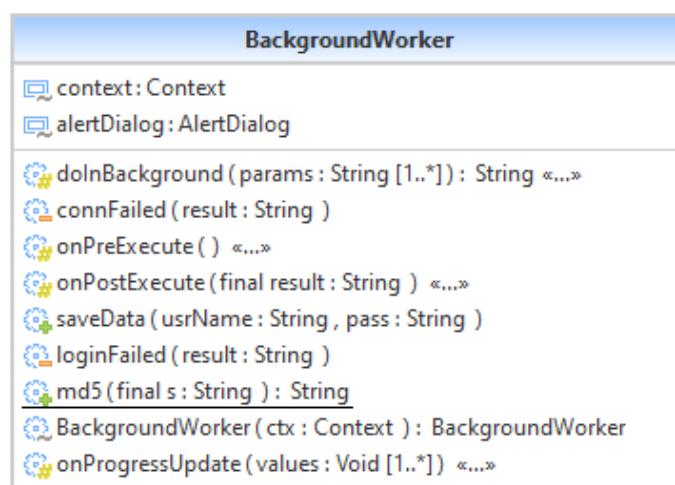


In LoginActivity we have 2 edit text boxes, in which user of our app may enter username and password. Along with them this class has 2 buttons, Login and Register.

User may press either.

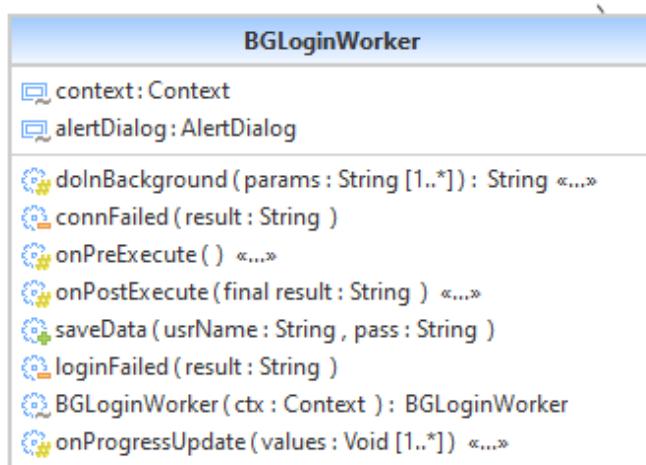
If login is pressed then string contents of 2 edit text boxes are sent BackgroundWorker. And the clear text result is displayed in AlertDialog Box.

If user presses register button. Then the user gets our web login and registration page in an android WebView.



This class receives and then passes username and password to the .php files. And when the result returns in clear text it displays it, in AlertDialogBox.

If the email id and password is a match, then saveData() and MainActivity is triggered.



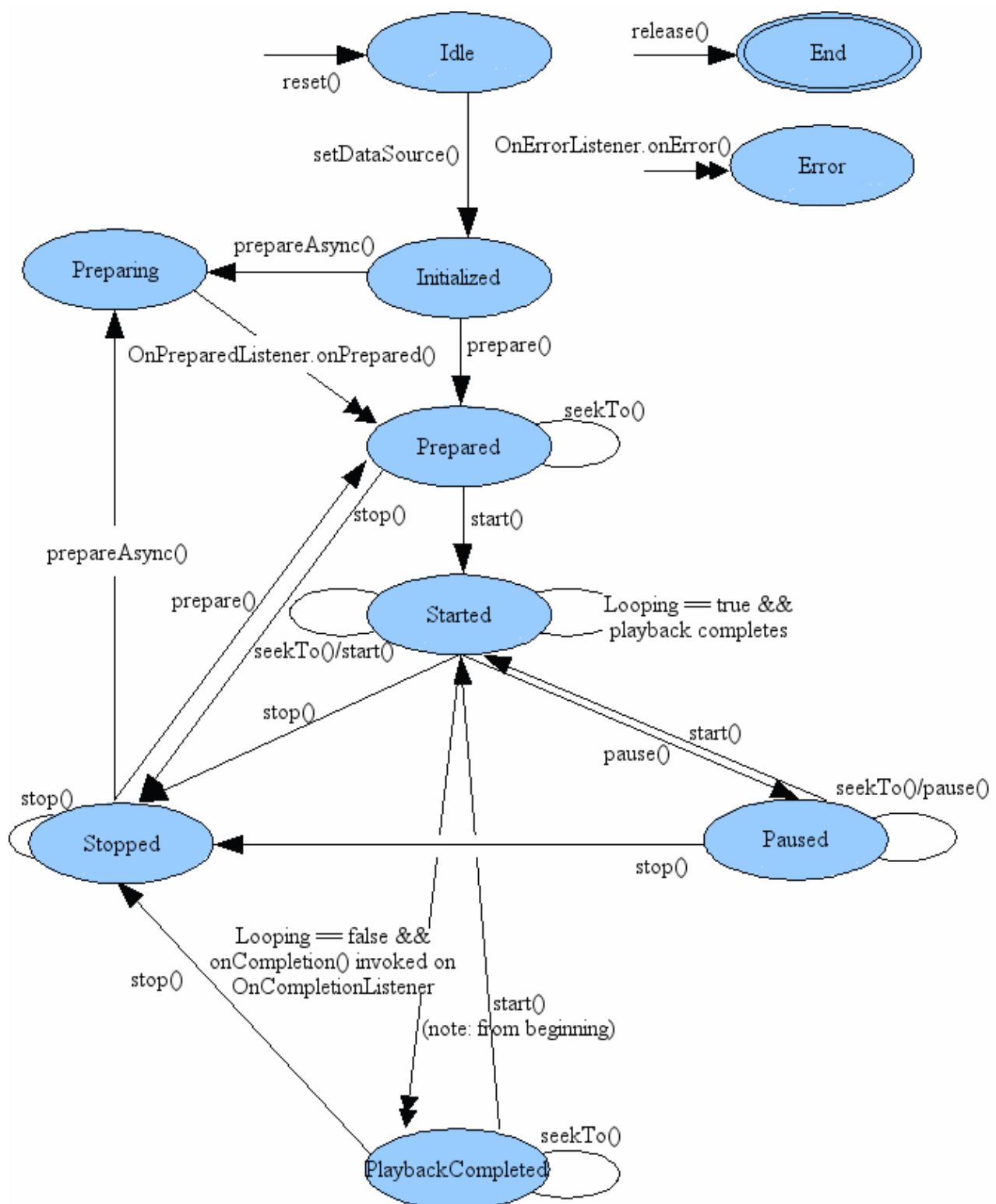
This class is essentially the same as `BackgroundWorker`. However it uses username and password (stored in md5 hash). Both of these are stored in app's shared preferences.

We use this to reverify that username and password has not changed, And that we have access to our backend.

This is triggered during the Splash screen. If user has previously logged in, then to skip the manual login we use this.

E. State diagram

State diagram of MediaPlayer class



CHAPTER 8

IMPLEMENTATION

Overview

This is the most critical step, we attempt to develop our project with all the analysis in mind, to achieve all of the essential and non-essential goals of our project. We will turn the theoretical design into a working system.

Modules specification

- A. Music player
 - a. Ultra Essential: Pause, Play
 - b. Essential: Next, Previous
 - c. Repeat, Repeat single
 - d. Shuffle, Random
 - e. Extra / Advanced Features:
 - i. Auto Next
 - ii. Auto Stop Timer
 - iii. Server managed synchronised current music queue
- B. User registration.
- C. Search functionality.
- D. Browsing via categories.
- E. My Music or Playlists.
- F. Settings, and editing user profile.
- G. Storing user preferences.

Above mentioned are the Features in our project, now we will review how we implemented each module

- A. Music player

This is the Main software component on our front page on website, and in our app on mobile device which plays or streams the given music file from our server.

Music streaming from a web-server to an Android app and Website.

- For the web page we use HTML DOM Audio Object.
- For android application, we Import ‘`android.media.MediaPlayer;`’ and then use MediaPlayer class. It can be used to control playback of audio or video files and streams.

a. Ultra Essential: Pause, Play

This is the functionality to pause the track being played in the middle, and when user presses play the track will resume from that point.

Web page:

- To play audio track which we have passed using PHP to convert the result into a JSON array, we use Audio Object Method ‘`.play()`’. To pause we use method ‘`.pause()`’. This DOM Object provides several methods to interact with audio object and control the music track we wish to play on our web page.

Android:

- We pass parameter in method like this:
`‘.setAudioStreamType(AudioManager.STREAM_MUSIC);’` This is used to identify the volume of audio streams for music playback. This informs the MediaPlayer to prepare and expect STREAM. For alarm and ring other options exist.
- Then we use ‘`.setDataSource(url);`’, ‘`.prepareAsync()`’. With URL string we pass the full qualified address of our music file which is on server. And then we call ‘`.setOnPreparedListener(--)`’ inside this method we use method ‘`.start()`’. Here `setOnPreparedListener()` is a method which is only triggered when we have called `prepareAsync()` and it has finished its job of preparing, because unlike `prepare()`, `prepareAsync()` doesn’t hold up the whole thread. `prepare()` holds the thread till the media is fully loaded / prepared making the app or that thread to freeze, especially if we have used MediaPlayer class on UI thread then our app will seem to be frozen. Inside `setOnPreparedListener()` we use `.start()` this starts the playback of the music.
- For pausing the playback, this class has `.pause()` method.

b. Essential: Next, Previous

This is the functionality play the next or previous file when users wants to, disregarding whether the current track has been completed.

Web page:

- In web page we have created 2 functions: ‘function nextSong()’ and ‘function prevSong()’ in both of these we use Audio Object function ‘setTrack()’ to pass the array index, that array index has the music track information.
- While pressing next, we test if shuffle flag is set to true or not, and so we shuffle is it is set true.
- When the prevSong() is initialised we check if the current duration of the song isn’t greater than 3 OR we check if the song is the first element in array, and if either of the conditions are true, we just simply restart the music track from the beginning. Otherwise it selects the previous track.

Android:

- When either the next or previous button is triggered with .setOnClickListener() we first use .stop() and .reset() on MediaPlayer object.
- After which we check if ‘rndm’ flag is true or not. And if it is true we randomise the Array List. Otherwise we either add or remove 1 from the array list index. We don’t do it when either the top or bottom end of the array list is reached.
- Whatever the ArrayList index may be, we have to do what we did when we press play. That is to .setDataSource(), prepareAsync().

c. Repeat, Repeat single

Web page:

- We use custom function. When triggered we set repeat flag to **true** or **false** and if the flag is **true**, whenever the playing music track ends, it keeps looping the song.

Android:

Music streaming from a web-server to an Android app and Website.

- Whenever a music track has completed its playback, ‘.setOnCompletionListener()’ is triggered. And when this is triggered, we check if the ‘REALL’ and ‘anxt’ are set to `true` along with ‘loop’ flag is `false`. If this condition is met we repeat reset the index to beginning when the end of array list is reached, and we triggered the `prepareAsync()` method.
- If the flags are exactly set opposite, that is ‘REALL’ and ‘anxt’ are set to `false` along with ‘loop’ flag is `true` then we simply pass `true` to ‘.setLooping()’ method. Then the song is restarted after it is completed
- REALL - Is Repeat the entire music queue / array list.
- anxt - Is auto next, to know if next song shall be played automatically or stop.
- loop - is to know if a single song has to be looped / restarted over and over.

d. Shuffle, Random

Web page:

- Just like repeat, we declare the shuffle flag true or false. And if `true` the `shuffleArray()` function is triggered when the playing music track ends, and then the random index is sent, so random element of array will be played.

Android:

- Just like repeat, or single repeat we set ‘rndm’ to true or false. And when the song is ended, ArrayList is copied and then is randomised, and random element is selected. By coping and then randomising the array we preserve the original order of the array list.

e. Extra / Advanced Features:

- We haven’t yet implemented these features in the web version.
 - i. Server managed synchronised current music queue
- We have implement a music queue or music list which is synchronised with the server database, and hence the same user account may be used simultaneously on multiple devices, and they all get to synchronise their lists.

Android:

- When a song name is touched for long duration, options are opened, and when ‘add to queue’ option is selected. The SongId of that song is appended to database table along with username. And that is retrieved in the music queue in the app.

ii. Auto Next

This is option to Enable or Disable if the player plays the next track (if any) in the list. Usually this option is never shown and is enabled in most of the apps and services. But we try to give a on or off option. And let the listener decide if they wish to go to next track or not automatically.

Android:

- If pressed the `anxt` flag is set `true`, if not it is `false`.

iii. Auto Stop Timer

This is rather unique feature, in all of the streaming services only a few have this option.

Android:

- Here we set a timer by using several inbuilt java methods and classes. User has to select on a ‘seek bar’, how many minutes they wish for the music to play. The range is between 1 to 180 minutes. When the time set by the user is elapsed, we stop the MediaPlayer object. And also set a few flag to false. If we don’t do this, we found that in during the testing songs would start playing, when the time elapsed in between 2 song tracks.

B. User registration.

Web page:

- We have used a form on our registration page. User shall enter details which we test and verify, we check whether the length of password isn’t greater than 30 or not smaller than 5. And if name has any unwanted characters. And if the string in the email has the pattern

of email id. We also check if the password, verification password and email and verification email matches with each other.

- If all the details are the way they should be and the email ID is unique (hasn't been used by other user,) then we send the data to database table. For security we use MD5 hashing for user's password. When the password is hashed with md5 reading the hash doesn't let any one entry into user's account.

C. Search functionality.

Web page:

- We have made a search page. When user enters any phrase and presses enter then a database query written inside the PHP code is triggered, it searches 3 tables for if the title or name columns have any part matching the string user has entered. The tables are: Songs table, Artist, and Album table.

Android:

- We have made a fragment, there are 4 sections in this layout, 1st is search bar.
- Search bar fragment has edit text field and search button.
- Then next 3 fragments are for Songs, Artist, and Albums respectively.
- Every time the user presses the search button, the string contents of the search bar are sent to 3 mentioned fragments. They each further pass it to PHP scripts of their own. Which queries in the similar fashion as for web version. And return the results, which are displayed by the fragment.
- If user sends an empty text query. The entire tables are returned as result.

D. Browsing via categories.

Browse page is designed in such a way that it allows users to discover some new Album or Artists or Genre or a Track which may be inside of these category.

Web page:

- We have made PHP pages with queries to retrieve and display relevant details for Browse page, Album Page, and for Artist.

Music streaming from a web-server to an Android app and Website.

- Browse page displays list of albums, and their album art.
- Album page displays songs in that album.
- Artist page displays that artist's songs and albums that they have worked on.

Android:

- Here we have created Browse, Genres, and Artist sections. When we click on an item displayed in these sections, may it be an album, or artist or genre name, the list of songs associated with them is retrieved from database. And it's displayed.
- Each Fragment is responsible to retrieve data for itself. For Browse, Artist, Genres pages we just retire names, IDs and Album art for albums.
- When a name is pressed, we pass the ID of that album / genre / artist to the fragment which will retrieve and display the contents.

E. My Music or Playlists.

This is a feature which can be considered important. This gives ability to user to make a list of the tracks as per user liking, and have it in any which way they wish as per user's discretion.

Web page:

- We have made a page to display available names of playlists, and have given an Add New Playlist button.
- PHP is used to retrieve and / or add a playlist.
- We have 2 tables in database. One with playlist name, id and creator's name. The other with id, song id, playlist id, and the order of sequence in which it is added.
- When we click open the playlist name, we get to see the songs which we have added into them.

Android:

- As mentioned above, we have 2 tables and have 2 PHP scripts to retrieve data from them.
- We have 2 buttons: refresh playlist list, and add new playlist. The first one, only reloads the fragment. While add new playlists, opens a dialog box where user needs to enter name for new playlist. And then that is sent to the PHP script, which adds it to the 1st table along with user name.

Music streaming from a web-server to an Android app and Website.

- First we need to only display names of playlist for the user which is logged in. If the user has any.
- When the playlist is pressed, we retrieve from the other table. We get SongID and Name of the song. We display list of the Names.
- To add any song to any playlist. Press and hold a song name, and choose add to playlist option.

F. Settings, and editing user profile.

We have given users ability to change their email ID and passwords.

Web page:

- When user clicks on the user's full name, they are presented with 2 buttons. They can either press Update Details button to open the page in which they can change their details. Or Logout button, to log out.
- When user click opens, Update Details page, they are presented with their current Email ID, where they can enter their new Email ID and press update, If it is a unique Email ID for our users table, then we let the user update. If they wish not to update then they can just let it be.
- On the same page they will see password section inside, they can enter their current password. And enter their desired new password twice. If their old password is correct and the newly entered password are matching with each other, then it is accepted.

Android:

- On Android when user selects User Details, From Navigation menu, User will see Email section and Password section.
- Email section has 1 EditText and 1 button. User's present Email ID is displayed, if user's wishes to they can change it or let it be, Change will be made, if that email ID is unique in our database's users table.
- In Password section, user will see 3 EditText boxes, they all have hints displayed in them so user won't have difficulty identifying what EditText is for what? User can replace their password, to do so, user has to enter their active password in the first box, and then add the new password in 2nd and 3rd boxes. The changes are made IF the current password is a match, and new passwords are same twice.

G. Storing user preferences.

Web page:

- We depend upon cookies with use of php to store user relevant data.
 - We use it to make sure user doesn't have to login each time, unless they have logged out.

Android:

-  In android, we store a few details in ‘SharedPreferences’.

For example:

- isLoggedIn It's a manual Boolean flag, set to true after successful login attempt.
With a successful login attempt we also save username and password in md5 hash format.

It is checked in Splash screen just after app is launched. If it is true then Login is skipped, and username and password is sent to BGLoginWorker. If it is set to false, Then user is taken to login screen.

- SHARED_LOG_USER Username is stored, after successful login attempt.
 - SHARED_LOG_PASSMD5 Password is processed through a method that we made. It converts string into MD5 hash. And then it's stored inside this.

- SHARED_rep We store the state of repeat feature for our music player.

It can be “REOFF”, which means the repeat function is set to off by the user.

Or “REAL”, which means to repeat the entire list of currently playing songs.

Or “REONE”, which means user wishes to repeat the single song over and over.

- **SHARED_nxt** We store if user wishes to play the next song automatically, or not.
Value can be “TRUE” or “FALSE”.
If the value is true, it’ll keep playing next song till the end of the list reached.
If false it’ll simply stop after playing the current song.
(Depends also upon value of ‘SHARED_rep’).

- **SHARED_rnd** We store the value as per user has set it, this is used to retain in memory if user turned shuffle on or off the last time.
Value can be “TRUE” or “FALSE”.
If the value is true, it’ll randomly select the song to play after the end of the last one.
If false it’ll select next as per the sequence.
(Depends also upon value of ‘SHARED_rep’). And affects value of ‘SHARED_nxt’.

CHAPTER 9

TESTING

Overview

Testing is the process of detecting errors, bugs, or unexpected results or events. Testing performs a very important role for quality assurance and ensuring the robustness of the software project. The results of testing can be used during maintenance. Testing is the process of trying to discover every possible fault or weakness in a software project. It provides a method through which we can check the functionality of components, sub-assemblies, assemblies and / or a finished project. It is also the process of exercising software with the intent of ensuring that the project meets the requirements, user expectations of that project, and does not fail in an unacceptable manner. There are various types of test that can be performed. Each type of test addresses a specific testing requirement.

Testing Objectives

The main objective of testing is to discover bugs, errors, in a systematic manner and with minimum effort, time and resources. Testing is a process where we execute the project with the intent of finding errors and bugs which exist.

- A successful test will be the one that discovers an undiscovered errors, bugs, functionality which may be sup par.
- A good test case will be the one that has a high probability of finding errors and bugs if they exist in our project. We must never deny the possibility of an error or bug to be present.
- Only a tested, and repaired / fixed software can be considered that of the higher quality and of the reliable standards.

Software Testing Types

- A. Unit Testing
- B. Integration Testing
- C. Regression Testing

Music streaming from a web-server to an Android app and Website.

- D. Smoke Testing
- E. Alpha / Beta Testing
- F. System Testing
- G. Stress Testing
- H. Performance Testing

A. Unit Testing

While in the development process, the programmer may test smallest unit of that project. To make sure that the unit is error free, and is working within expected parameters. Unit Testing doesn't have to be done only during the development process itself.

This is White Box testing approach.

B. Integration Testing

While individual units may be working as expected. But it is necessary for them to work in the correct manner with each other. Hence all the components of project are integrated together. And then they are tested.

C. Regression Testing

This is done after minor changes. If the product was working fine before we just test that it is performing in the similar fashion after the changes. Test cases which had worked as expected are re-executed to confirm the performance.

D. Acceptance Testing

This is the last step of testing. The user or client has to use and review our product. Because they are the target audience for whom we have made it. Disregarding the fact that it may be covering 100% of defined goals, yet if the user(s) / client don't like our software then the whole project is a failure. However it'll be very rare to get totally unacceptable review from users, usually it is something that we can handle, and repair or change to make it more pleasing to the final end user.

Two methods of acceptance testing:

a. Alpha Testing:

Alpha version of the software is when it's used as if it's an unstable final software, by the development team itself, or anyone who is approved by the team. Usually it's done in-house, trying to simulate how the users will use it.

b. Beta Testing:

Beta version of the software is when it's passed the Alpha phase, and then it is released in a limited fashion to some early users. This is done only for testing purposes. The developer team expects these users to highlight and bring to notice minor and major errors that may have been skipped till now.

E. System Testing

The software project is finally compiled as a product and then it is tested as an entire system.

Testing Approaches:

a. Functionality Testing Also known as Black Box Testing:

We tested all the functionalities, and goals from the project. In Black Box testing the tester is only interested in testing the functionality of the project. If there happens to be efficiency loss due to wrong method of implementation, and yet if the result is the same, with some loss of time... in that case a Black Box Tester won't be able to know and identify that bug.

b. Implementation Testing Also known as White Box Testing:

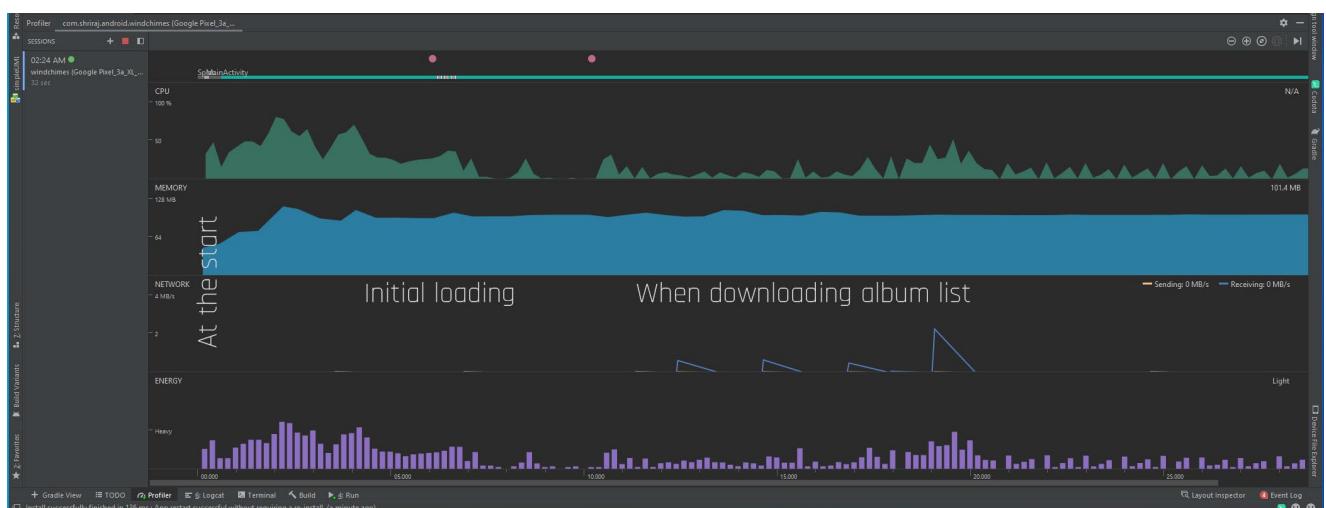
This test is done after the implementation, we try to find error or unexpected behaviour when implemented. We try to get gauge on the performance of our software project. We use load testing and stress testing. Here we tried to get output under various environmental conditions. In White Box Testing the tester also has access to the development tools and the code, so they can judge the implementation better.

Challenges and Solutions found as result of testing phase

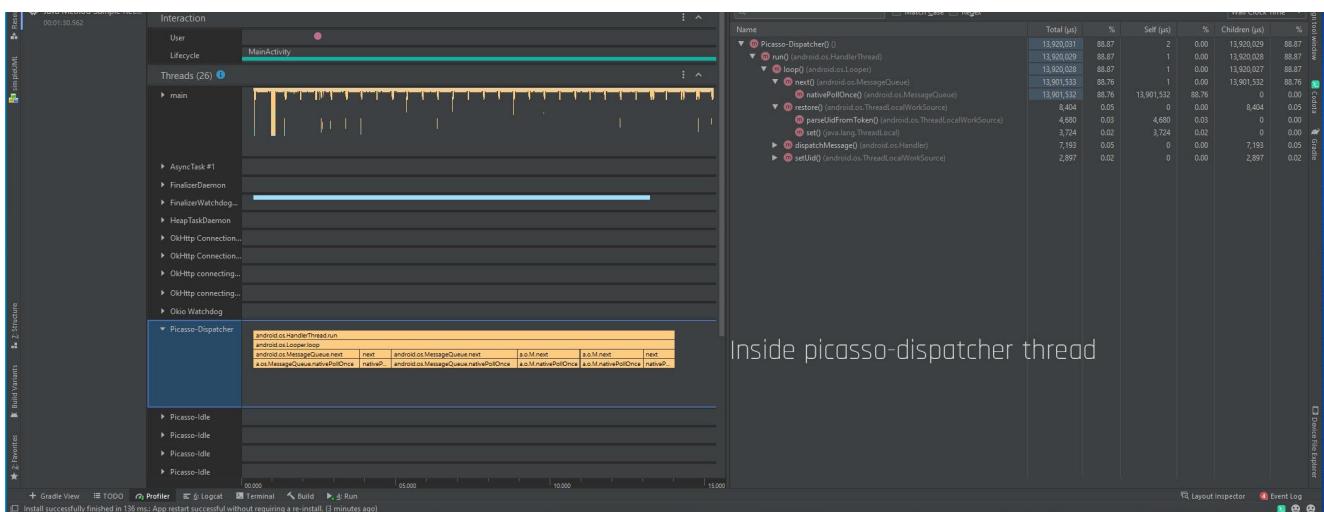
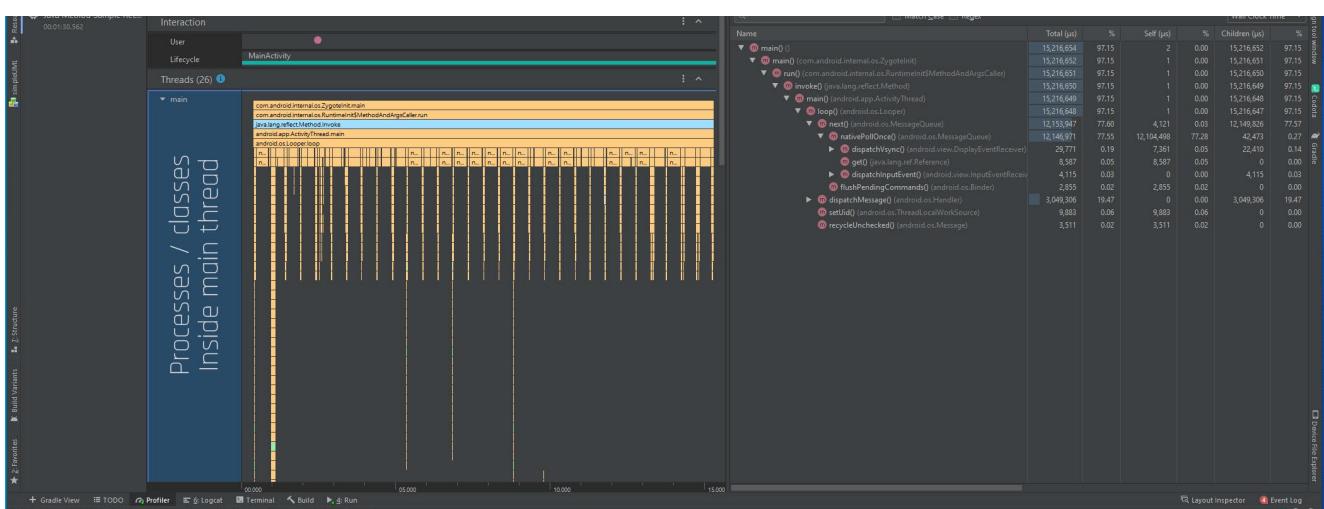
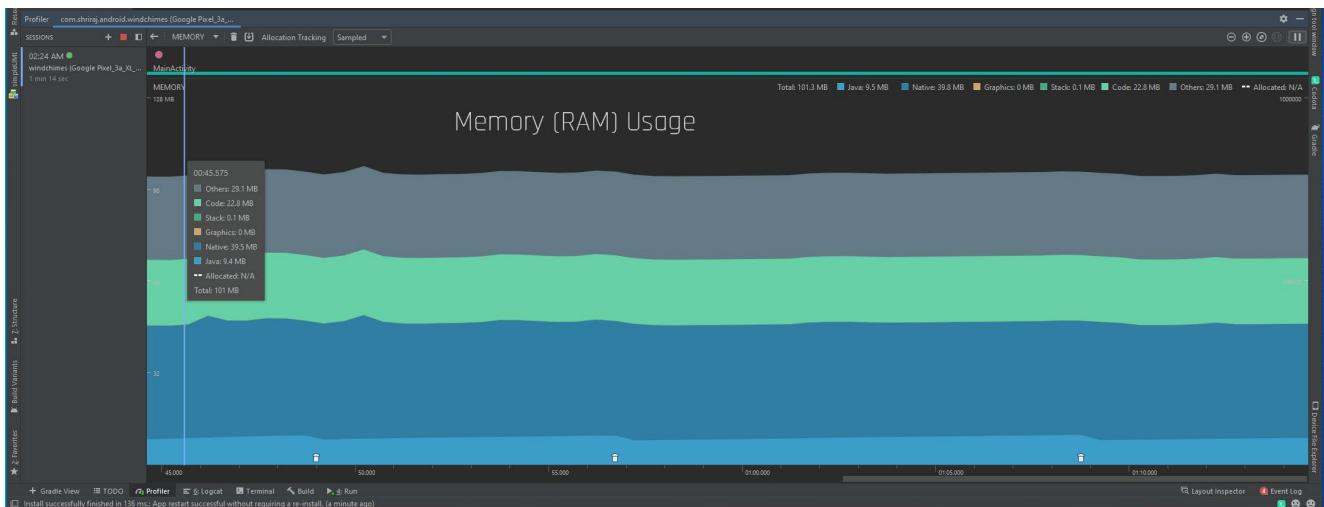
- We tested our project with unit testing, with integration, and with regression testing multiple times during and after the initial development phase itself. At these stages we found a lot of error and bugs, which we kept fixing along the way.
- After the first revision of the website was done, we encouraged our friends, and colleagues to perform a black box testing of the site. During which we identified that our website's Music Player wasn't accepting touch input from mobile devices. This was due to a minor JavaScript DOM feature that we had enabled.
- We were using image assets / resources which were very high in their size or their resolution or sometimes both. It is advised that while making app, developer should use image assets as small as possible in size and resolution. After fixing this issue the out of memory error was eliminated, and the app was supported on high number of devices. Before doing this, the app would run only on Emulators and devices running android API 28 or above.
- The User details form in android may be having layout related issues, on some devices.

Actual Performance and Resource requirement by our app

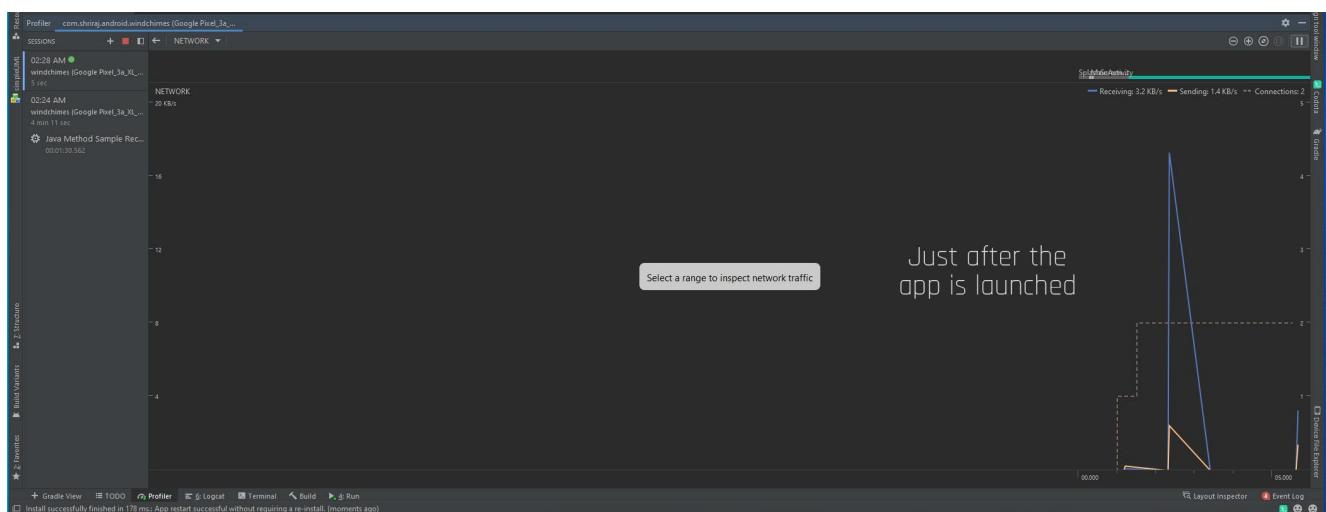
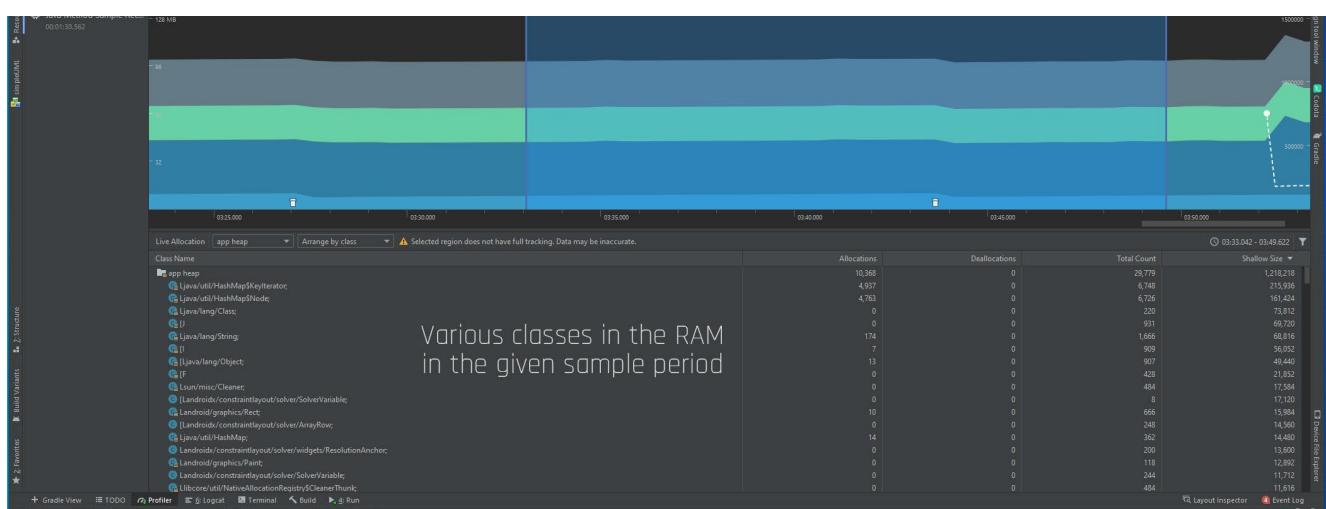
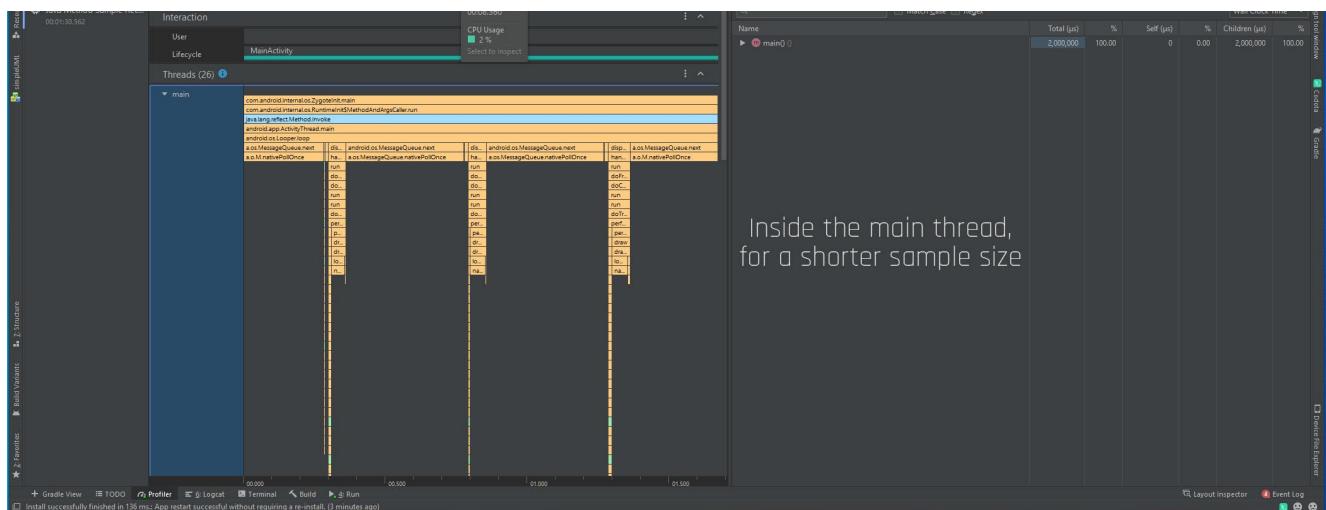
- We use profiler, a tool which is packed in Android Studio IDE.
- It attempts to give us live resources usage, and energy impact.



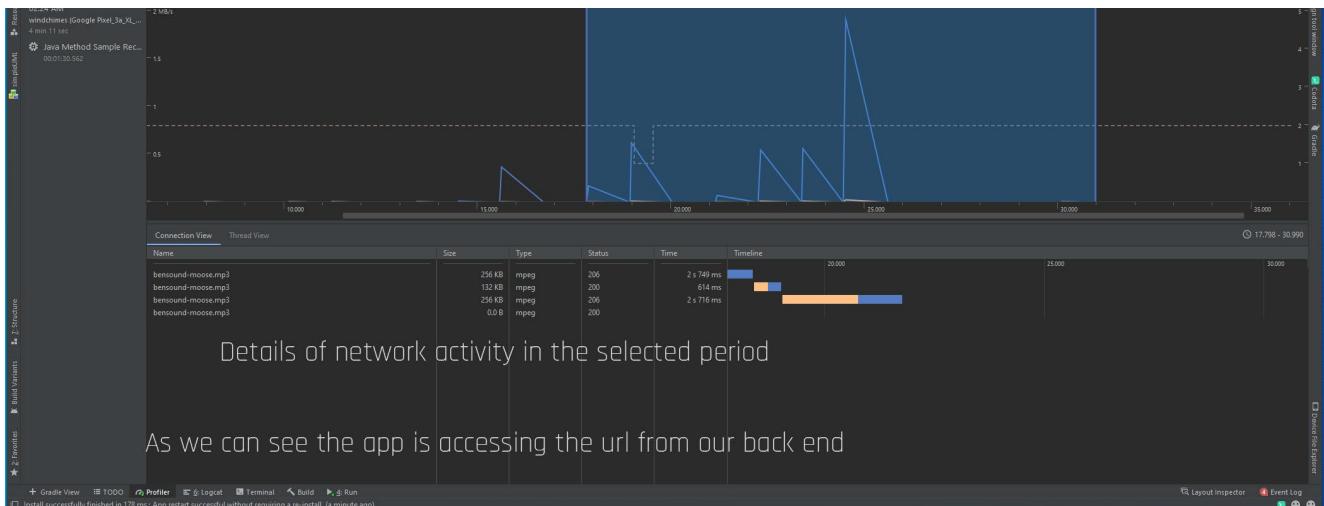
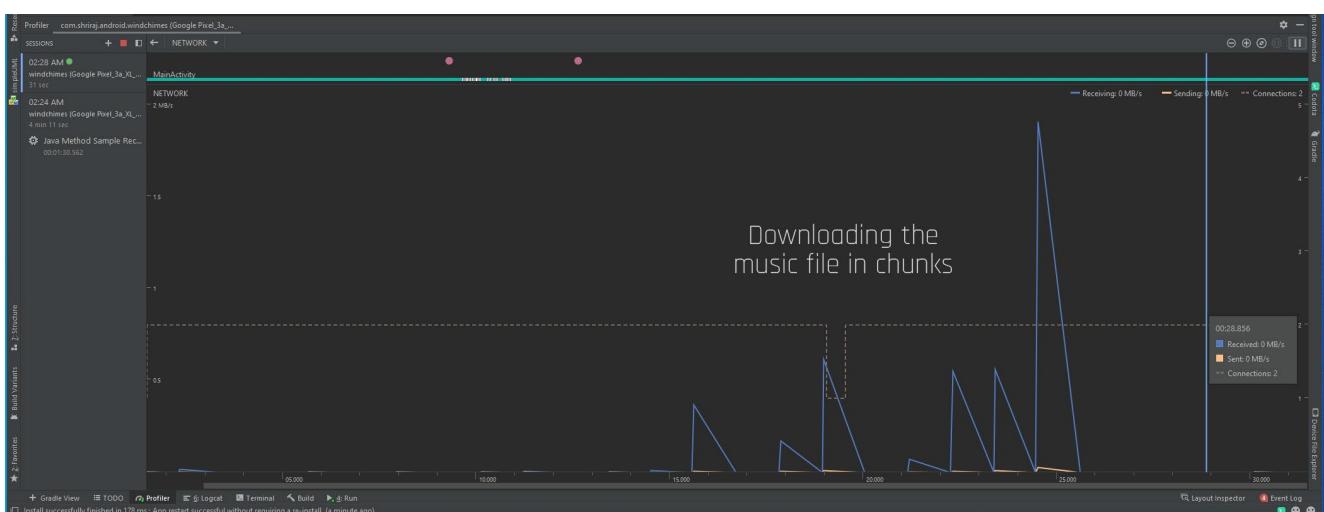
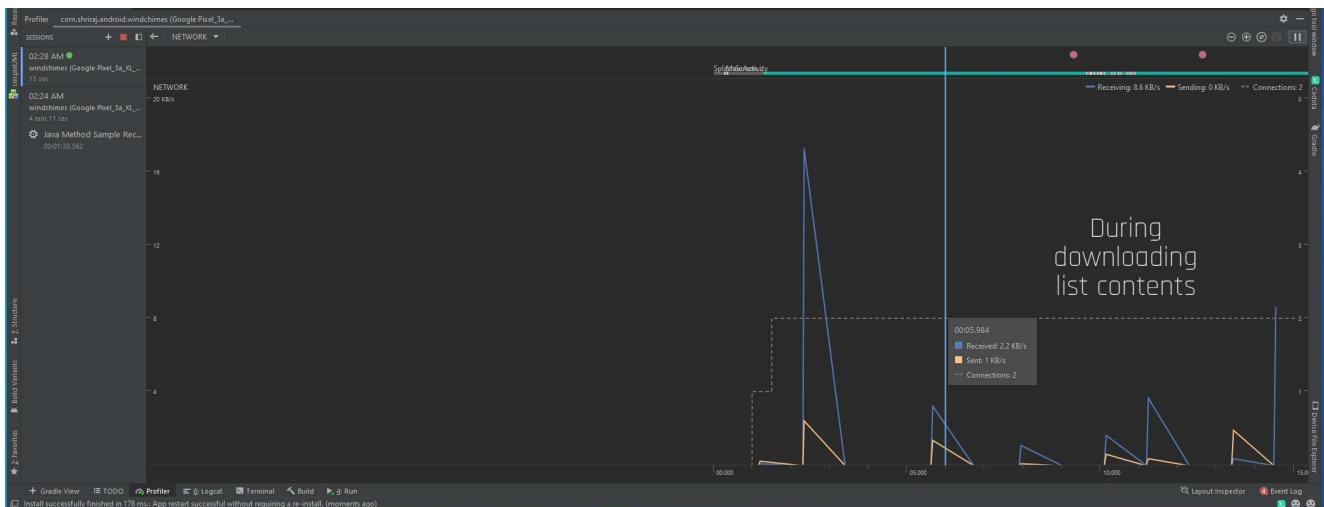
Music streaming from a web-server to an Android app and Website.



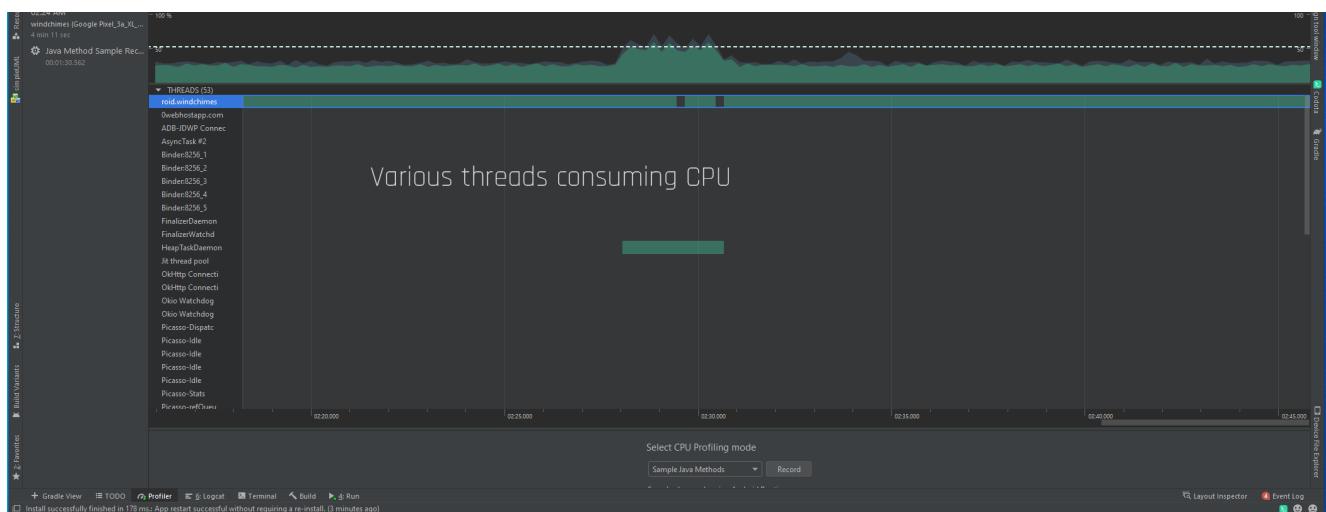
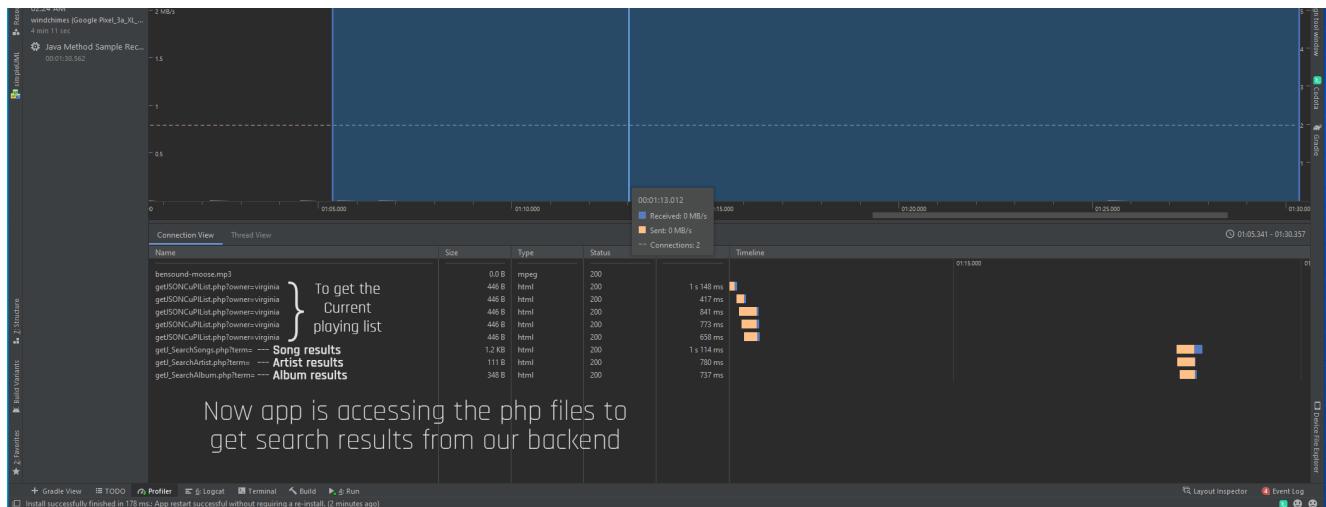
Music streaming from a web-server to an Android app and Website.



Music streaming from a web-server to an Android app and Website.

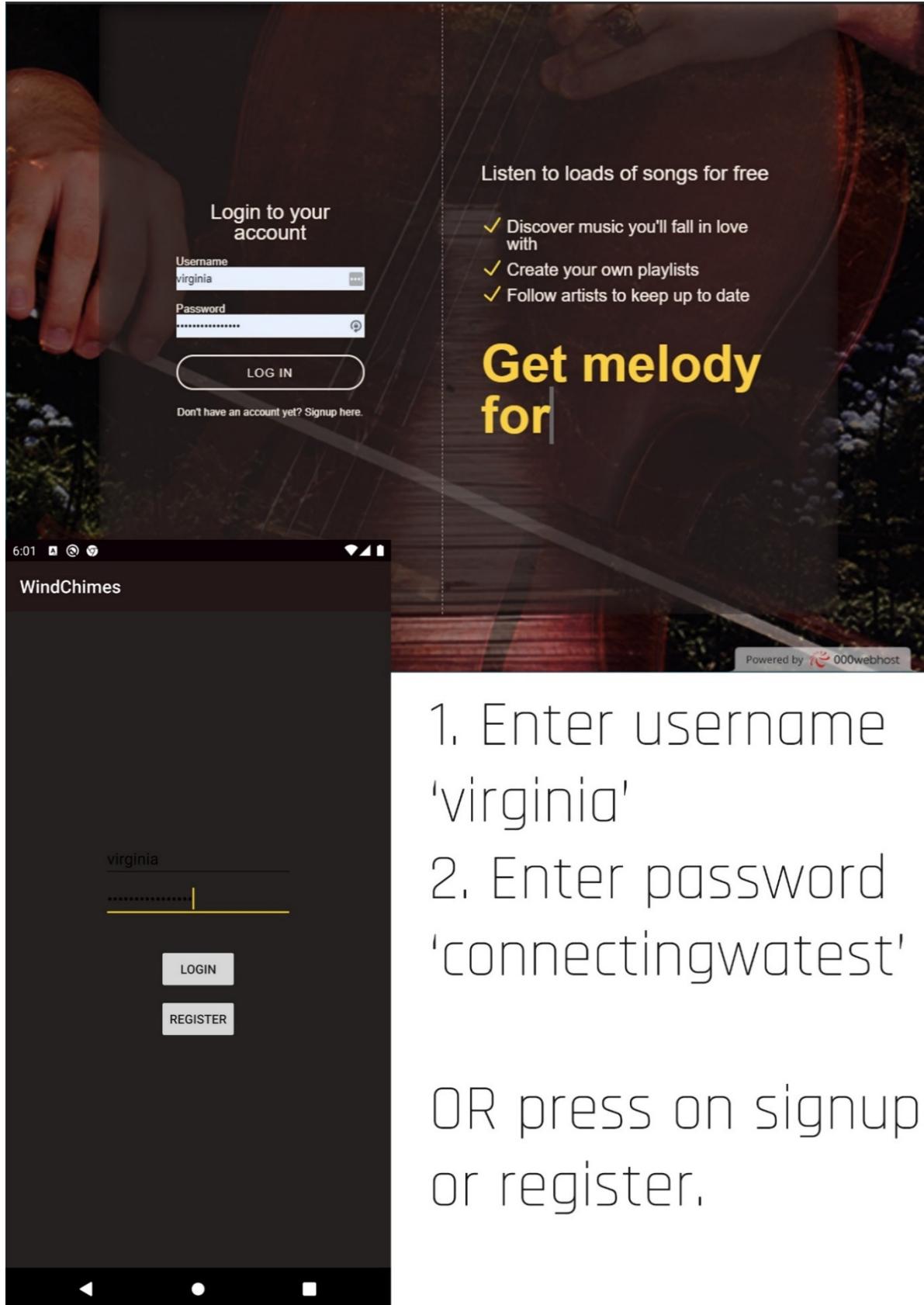


Music streaming from a web-server to an Android app and Website.



CHAPTER 10

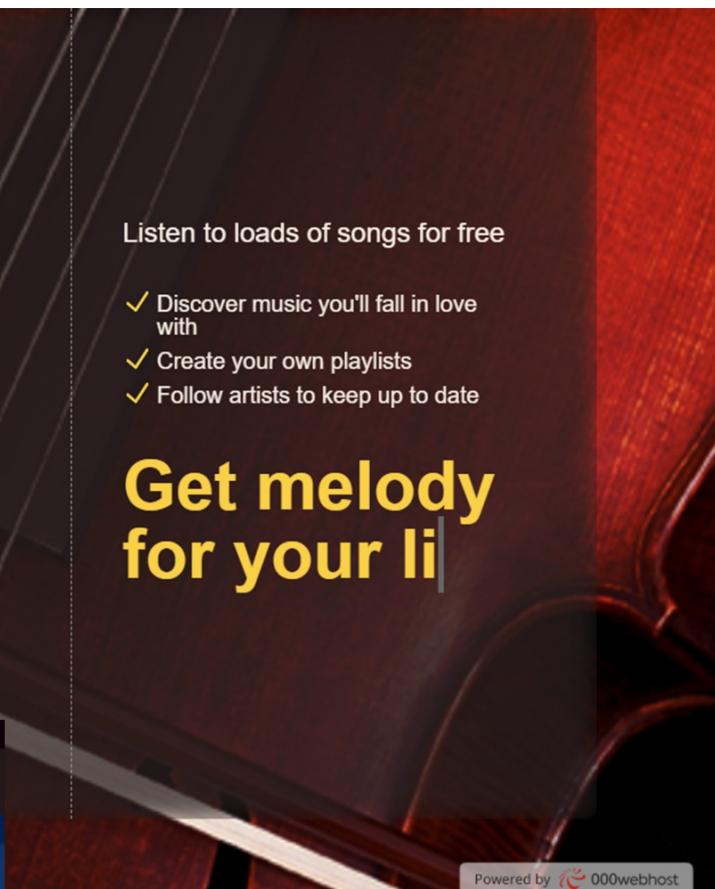
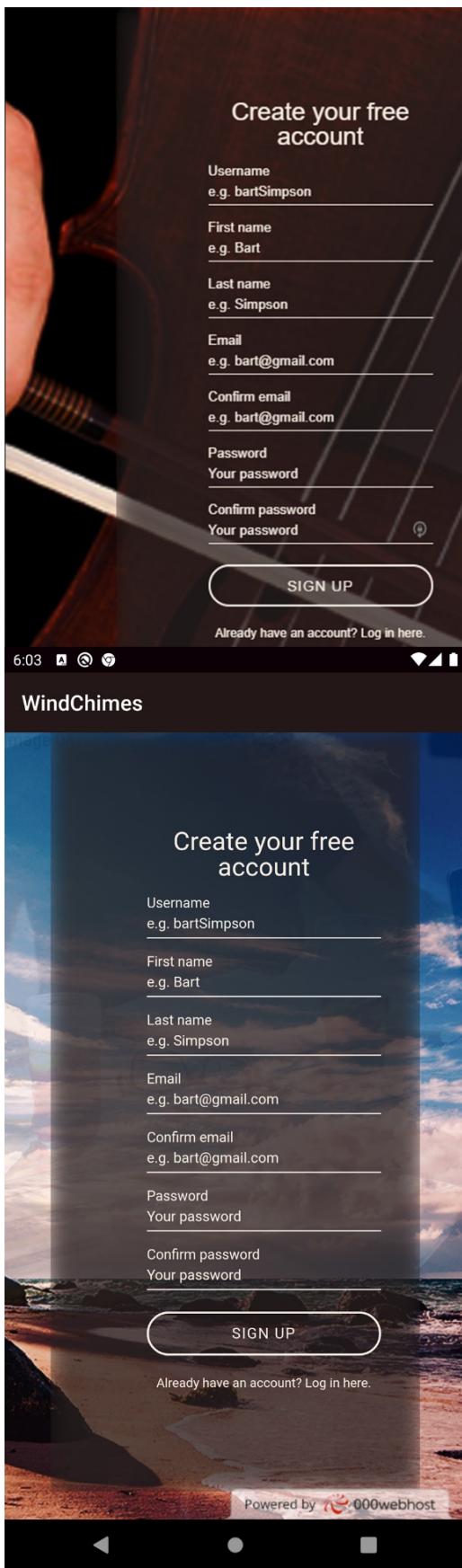
SCREENSHOTS



1. Enter username
'virginia'
2. Enter password
'connectingwatest'

OR press on signup
or register.

Music streaming from a web-server to an Android app and Website.



Enter relevant details and press sign up

In android device after signup please press return key.

Music streaming from a web-server to an Android app and Website.

The image displays a composite view of a music streaming application. On the left, a dark-themed web interface shows a sidebar with 'Search' and 'Browse' buttons, and a 'Your Music' section for 'Kirsten Wiseman'. A 'You Might Also Like' section features eight album covers: 'Pizza head', 'Bacon and Eggs', 'Best of the Worst', 'Summer Hits', 'Best beats', 'Hello World', 'The movie soundtrack', and 'WindChimes'. The central part of the image shows an Android mobile application window titled 'WindChimes'. It lists several albums: 'Best beats' (Homer, Techno), 'Hello World' (Bart Simpson, Classical), 'Best of the rack' (Mickey Mouse, Hip-hop), 'The m' (Goofy, Folk), 'Summer Hits' (Folk), and 'Pizza head' (Techno). A media control bar at the top of the app shows volume, battery, and time (5:08). A watermark 'Powered by 000webhost' is visible. To the right of the mobile app, large text reads 'Pres and open any album'.

The image shows a composite view of a mobile device and a computer monitor. On the left, a smartphone displays a music player application. The screen shows a list of songs from an album titled "Best of the Worst" by Mickey Mouse. The songs listed are "Clear Day", "Buddy", "Better Days", "A new beginning", and "Acoustic Breeze". Each song entry includes the artist name "Mickey Mouse" and its duration. At the top of the phone's screen, there is a navigation bar with icons for search, browse, and user profile. On the right, a computer monitor displays a music streaming website. The website has a dark theme and features a large image of a hand interacting with a digital interface against a blue sky with clouds. Below the image, the album title "Bacon and Eggs" is displayed, along with the artist "By Goofy" and "5 songs". A list of five songs follows, identical to those on the phone: "Going Higher", "Funky Element", "Extreme Action", "Energy", and "Dubstep", all by "Goofy". The bottom right corner of the computer screen shows a watermark for "Powered by 000webhost".

You are inside selected album.

Press any song to play.

Tap and Hold (android)
or
Press 3 dots (web)
To get options

The image shows a composite view of a mobile device and a computer screen demonstrating music streaming. On the left, a smartphone displays a dark-themed music player interface. At the top, there's a navigation bar with 'Search' and a magnifying glass icon, followed by 'Browse', 'Your Music', and 'Kirsten Wiseman'. Below this is a large album cover for 'Bacon and Eggs' by Goofy, featuring a hand interacting with a floating digital interface against a blue sky with clouds. The main area shows a list of five songs:

Rank	Song Title	Artist	Length
1	Going Higher	Goofy	4:04
2	Funky Element	Goofy	3:08
3	Extreme Action	Goofy	8:03
4	Energy	Goofy	2:59
5	Dubstep	Goofy	2:03

At the bottom of the phone screen, there are standard Android navigation icons (back, home, recent apps). On the right, a computer monitor displays a web-based music player. The title 'Best of the Worst' is visible, along with album art for 'Mickey Mouse' and 'Hip-hop'. A list of songs is shown, including:

Song Title	Artist	Length
Clear Day	Mickey Mouse	1:29
Buddy	Mickey Mouse	2:02
Better Days	Mickey Mouse	2:33
A new beginning	Mickey Mouse	2:35
Acoustic Breeze	Mickey Mouse	2:37

The computer screen also features a navigation bar at the top with 'Search' and a magnifying glass icon, 'Browse', and 'Your Music'. It includes a search bar with placeholder text 'Search' and a magnifying glass icon. The overall theme is a comparison between mobile and web-based music streaming experiences.

You are inside selected album.

Press any song to play.

Tap and Hold (android)

or

Press 3 dots (web)

To get options

The image displays a composite view of a mobile device screen. On the left, a music player application is shown with a dark theme. It features a sidebar with 'Search' and 'Browse' options, and a main area showing an album cover for 'Bacon and Eggs' by Goofy, which contains 5 songs. Below the album cover is a list of songs:

Rank	Song Title	Artist	Duration
1	Going Higher	Goofy	4:04
2	Funky Element	Goofy	3:08
3	Extreme Action	Goofy	8:03
4	Energy	Goofy	2:59
5	Dubstep	Goofy	2:03

At the bottom of the mobile screen, there is a navigation bar with icons for back, forward, and other controls, along with a volume slider and a 'Powered by 000webhost' watermark.

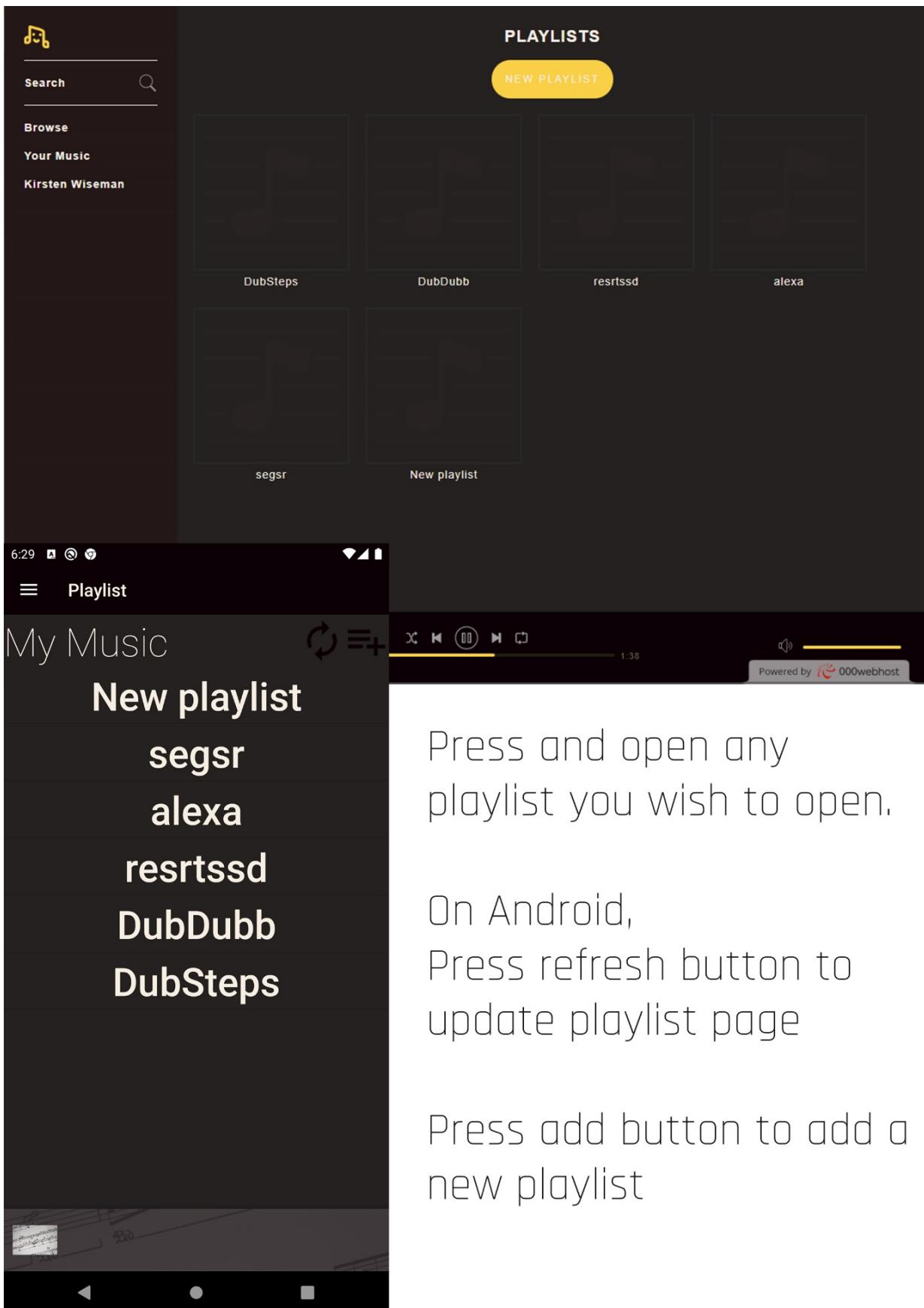
On the right side of the image, a separate web browser window is open, also displaying the 'Bacon and Eggs' album page, showing the same song list and album details.

You are inside selected album.

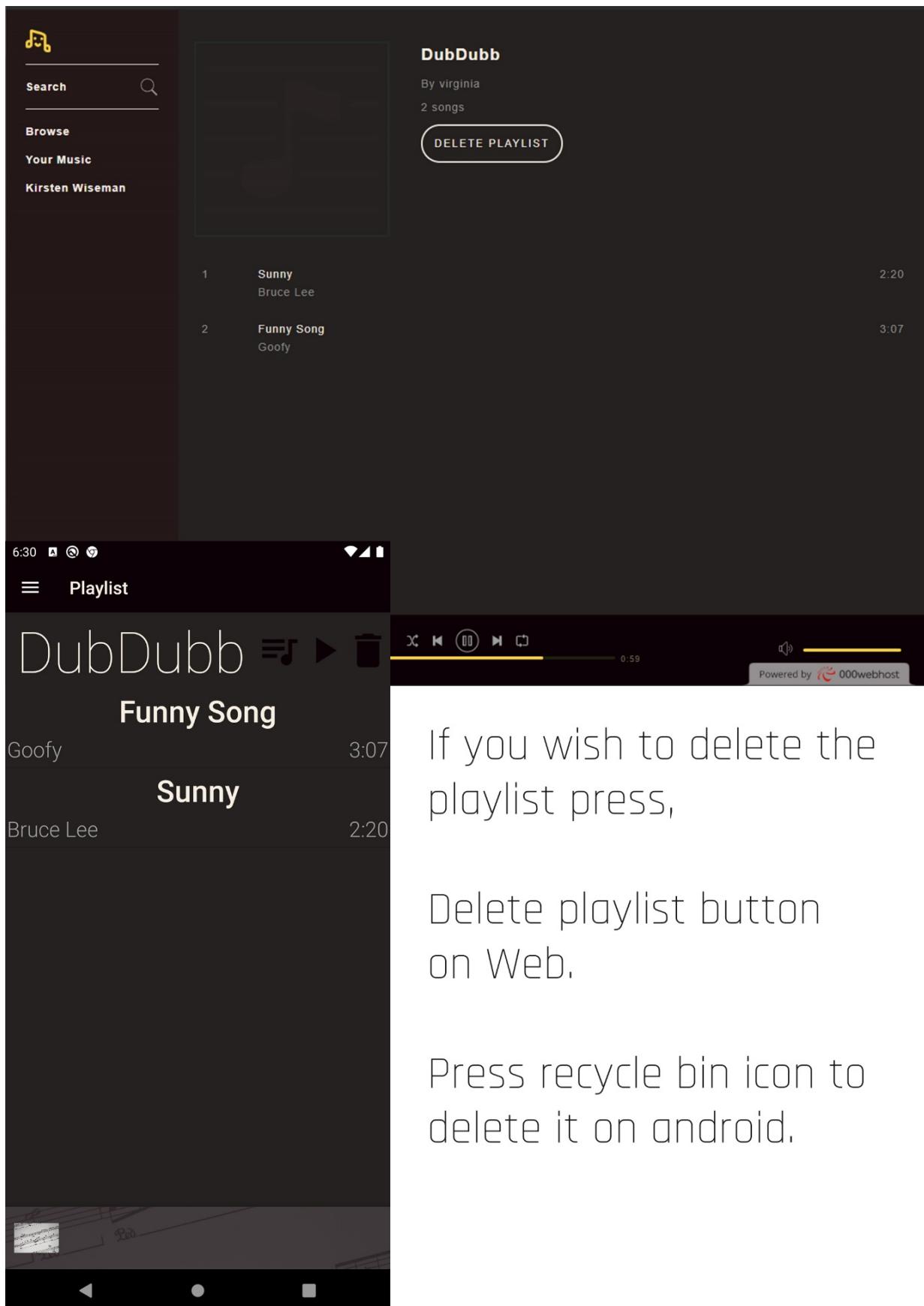
Press any song to play.

Tap and Hold (android)
or
Press 3 dots (web)
To get options

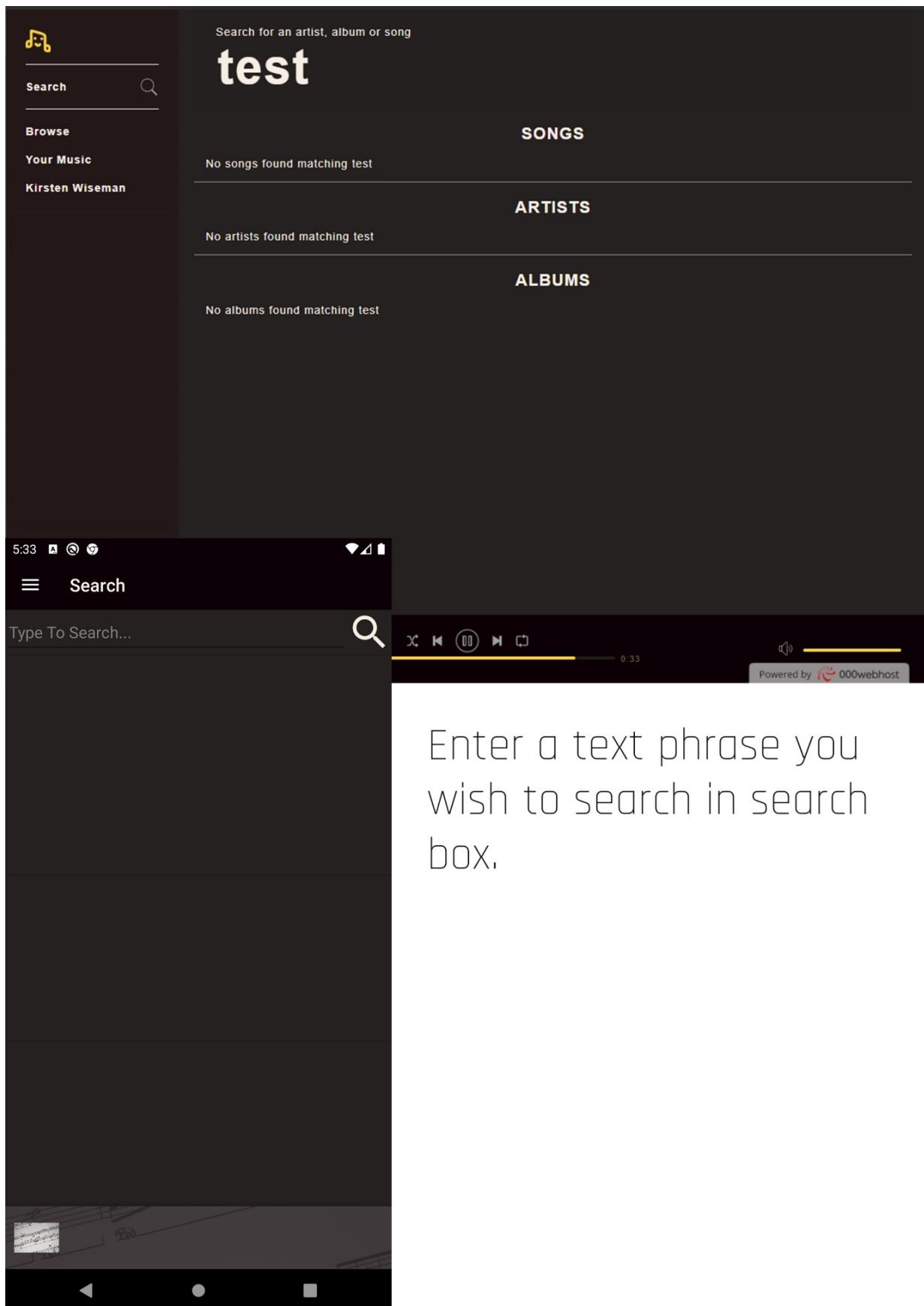
Music streaming from a web-server to an Android app and Website.



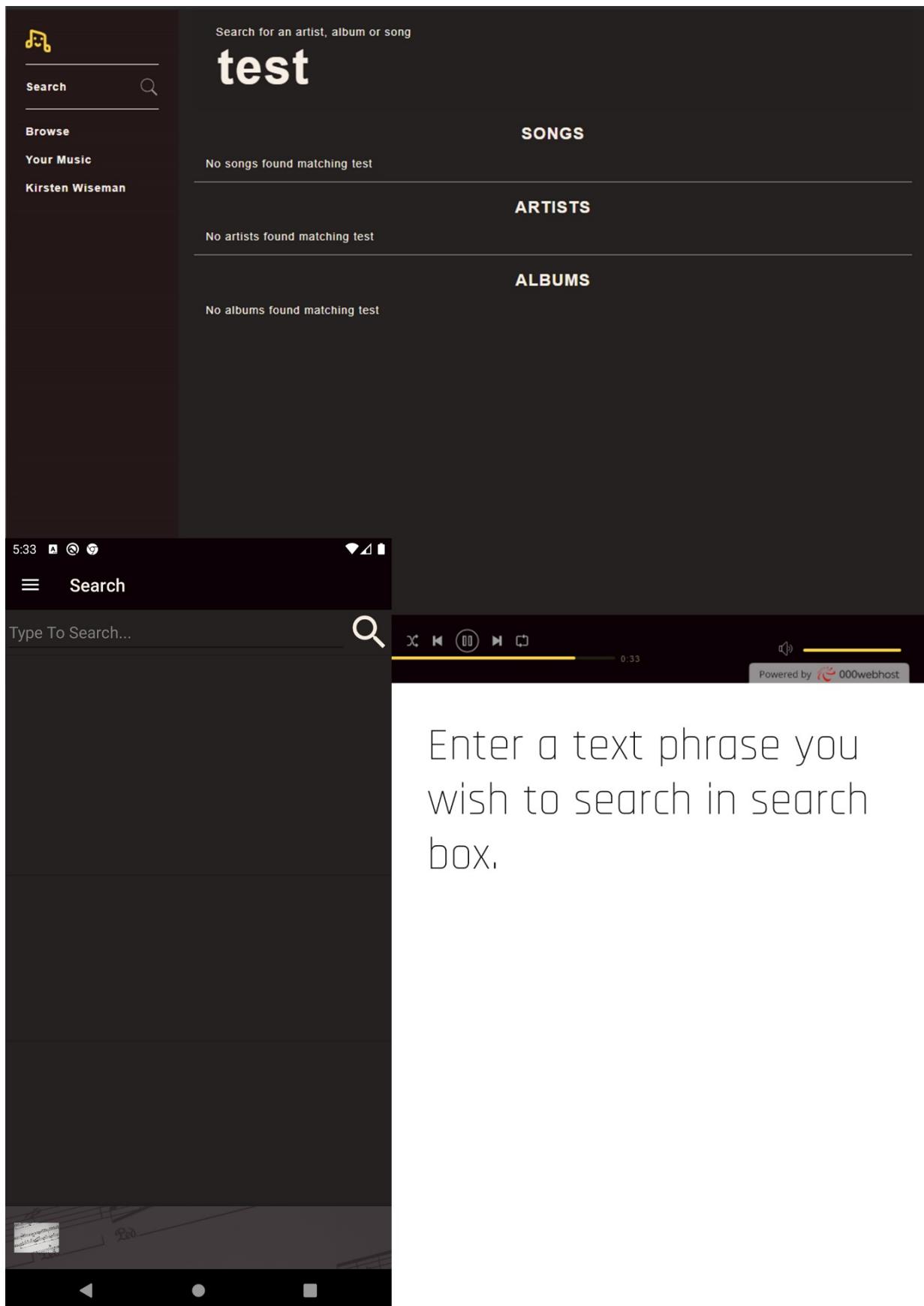
Music streaming from a web-server to an Android app and Website.



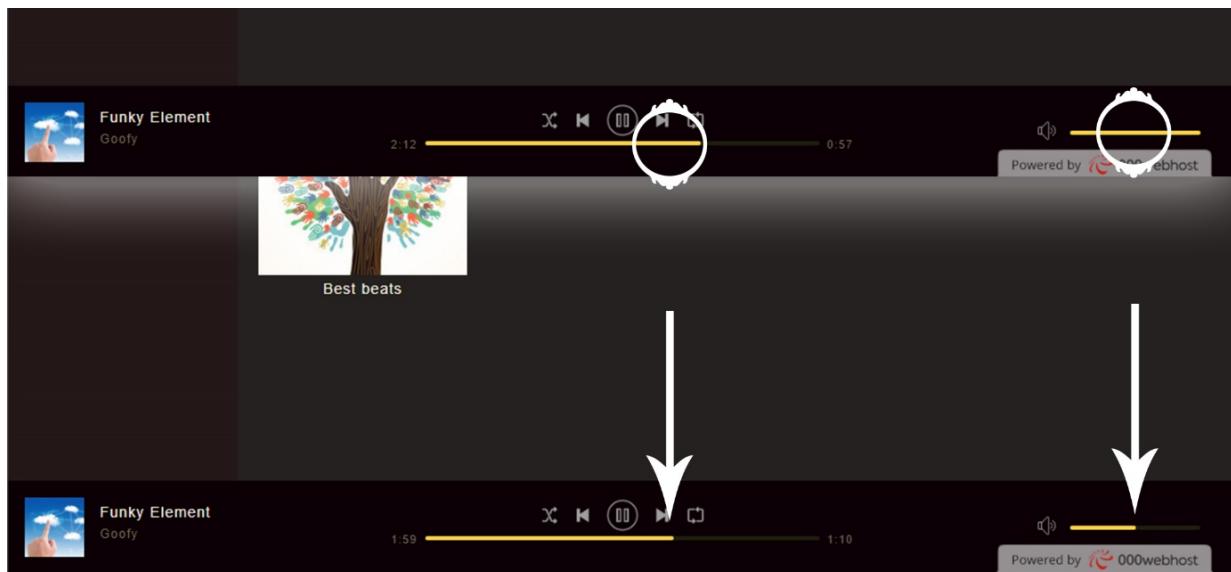
Music streaming from a web-server to an Android app and Website.



Music streaming from a web-server to an Android app and Website.

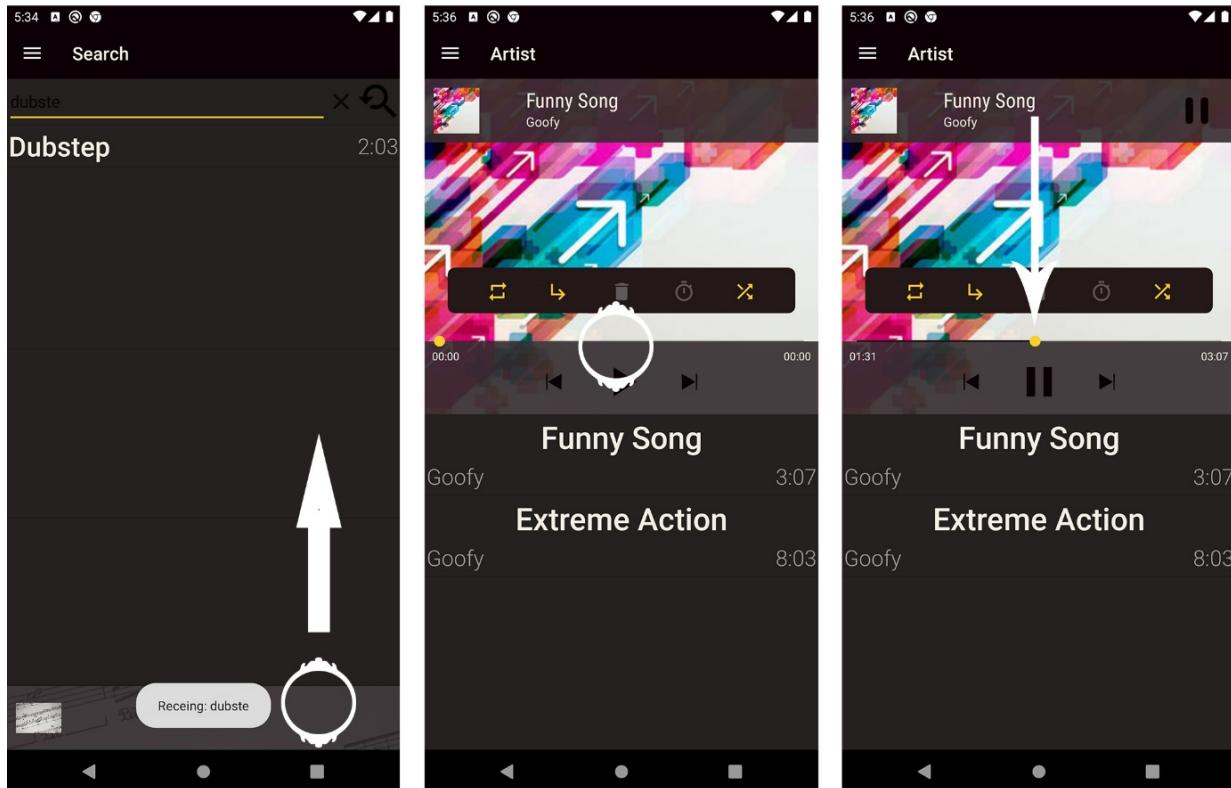


Music streaming from a web-server to an Android app and Website.

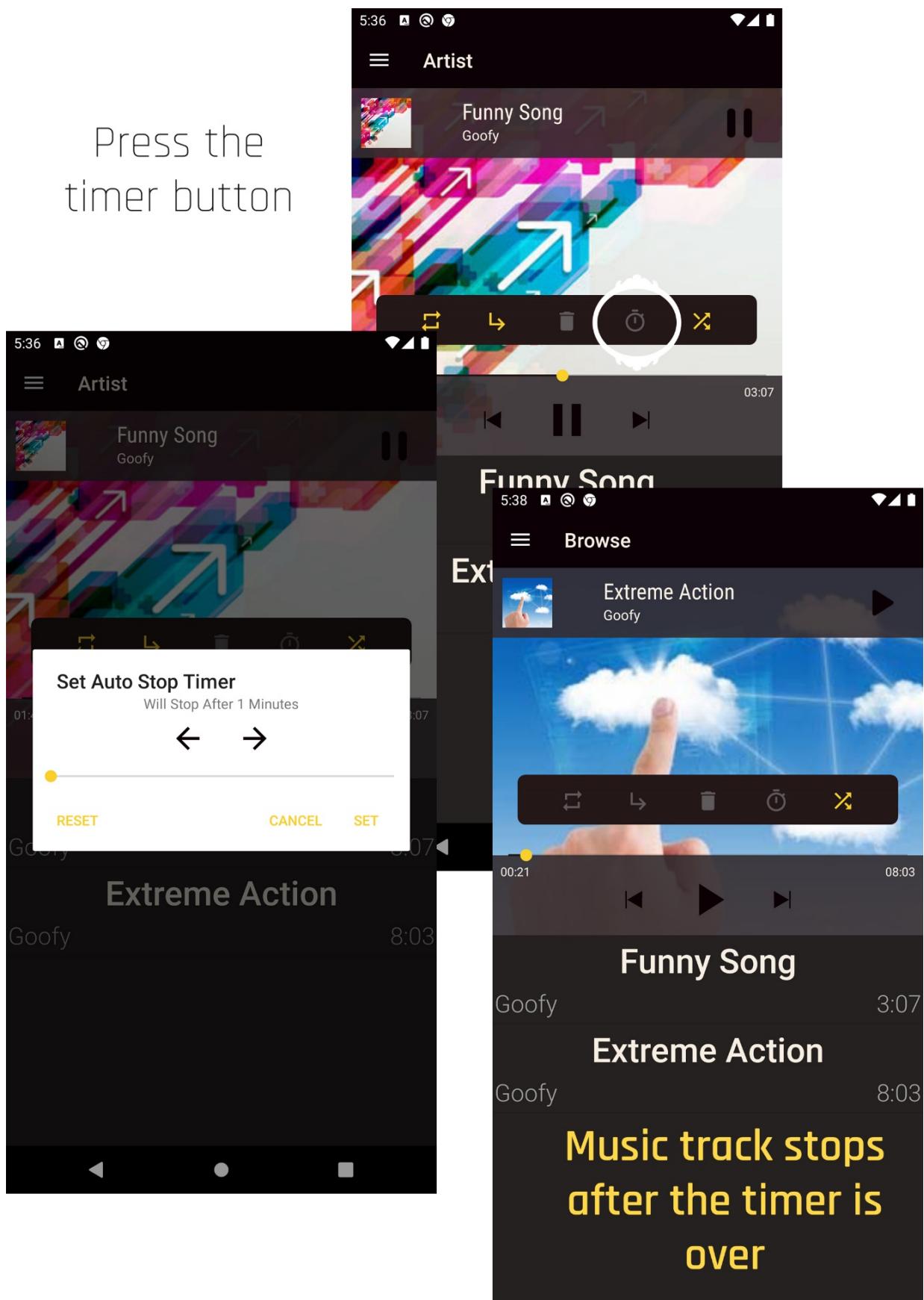


For web, Press on seek bar or volume bar where ever you wish

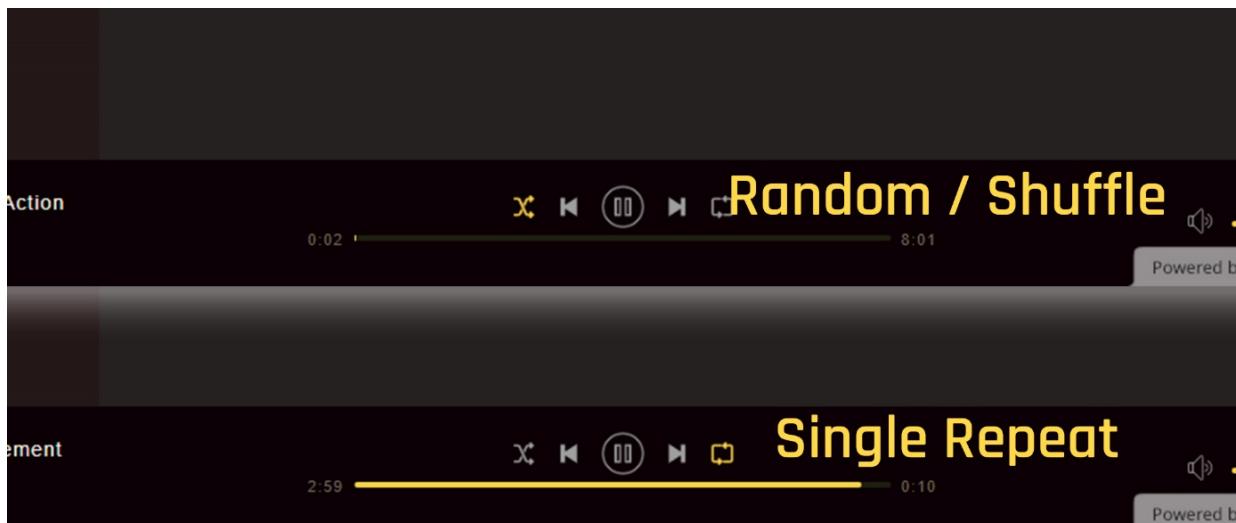
For android , drag and open the Music Player. Then press on seek bar, where ever you wish



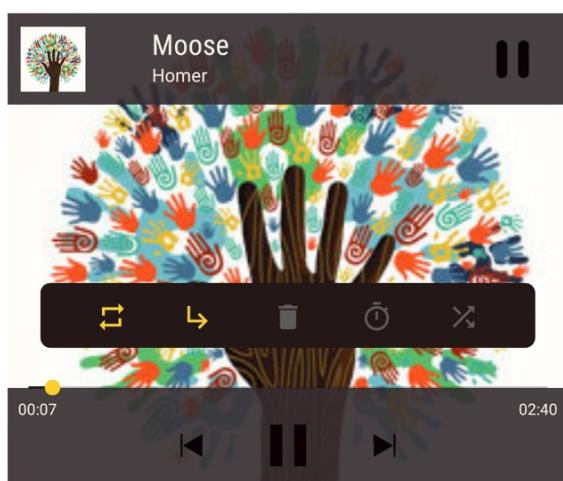
Music streaming from a web-server to an Android app and Website.



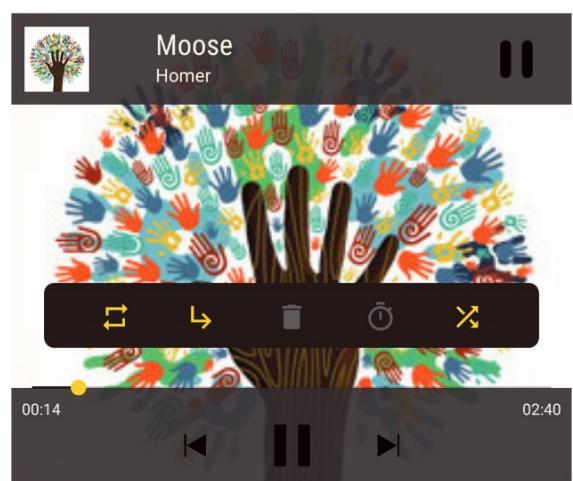
Music streaming from a web-server to an Android app and Website.



Single Repeat

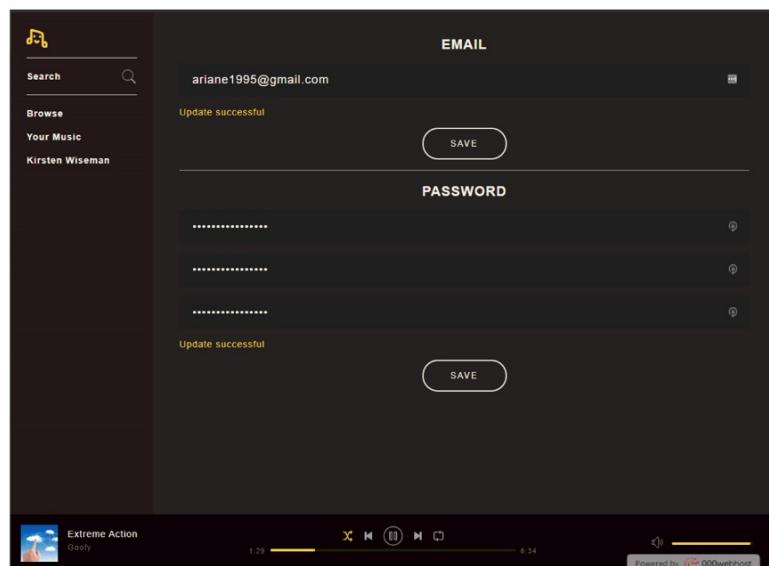
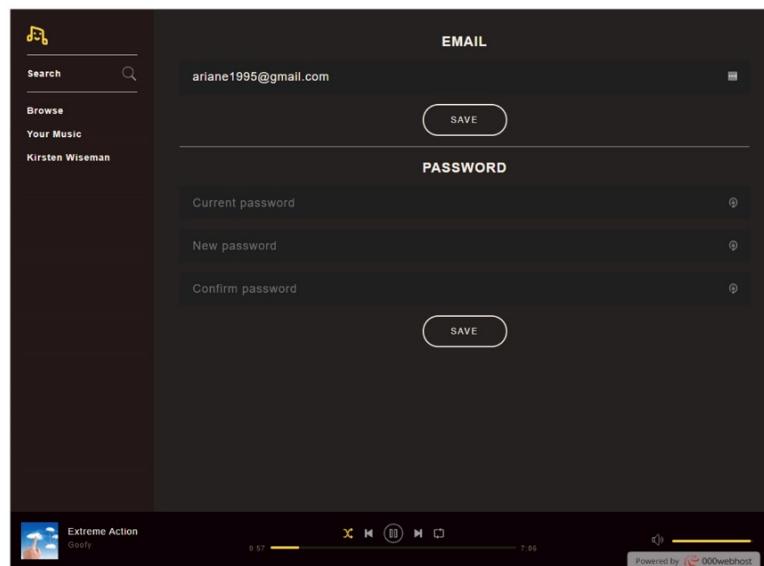
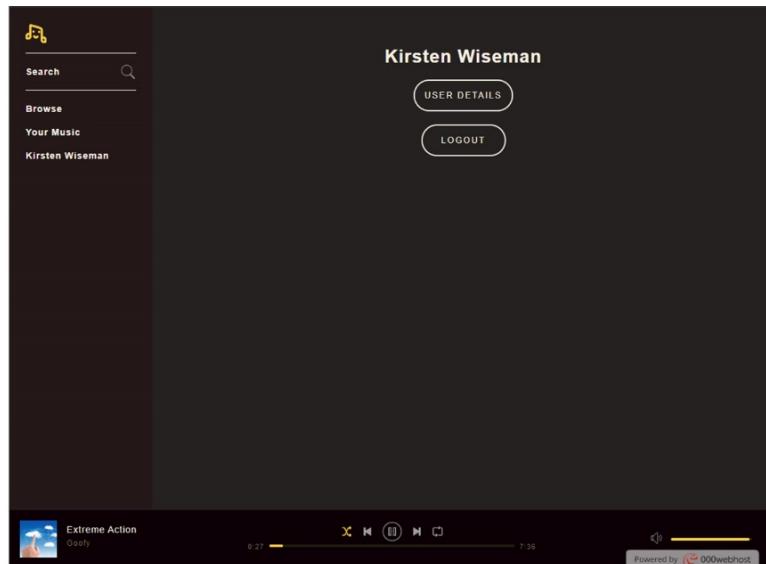


Repeat All

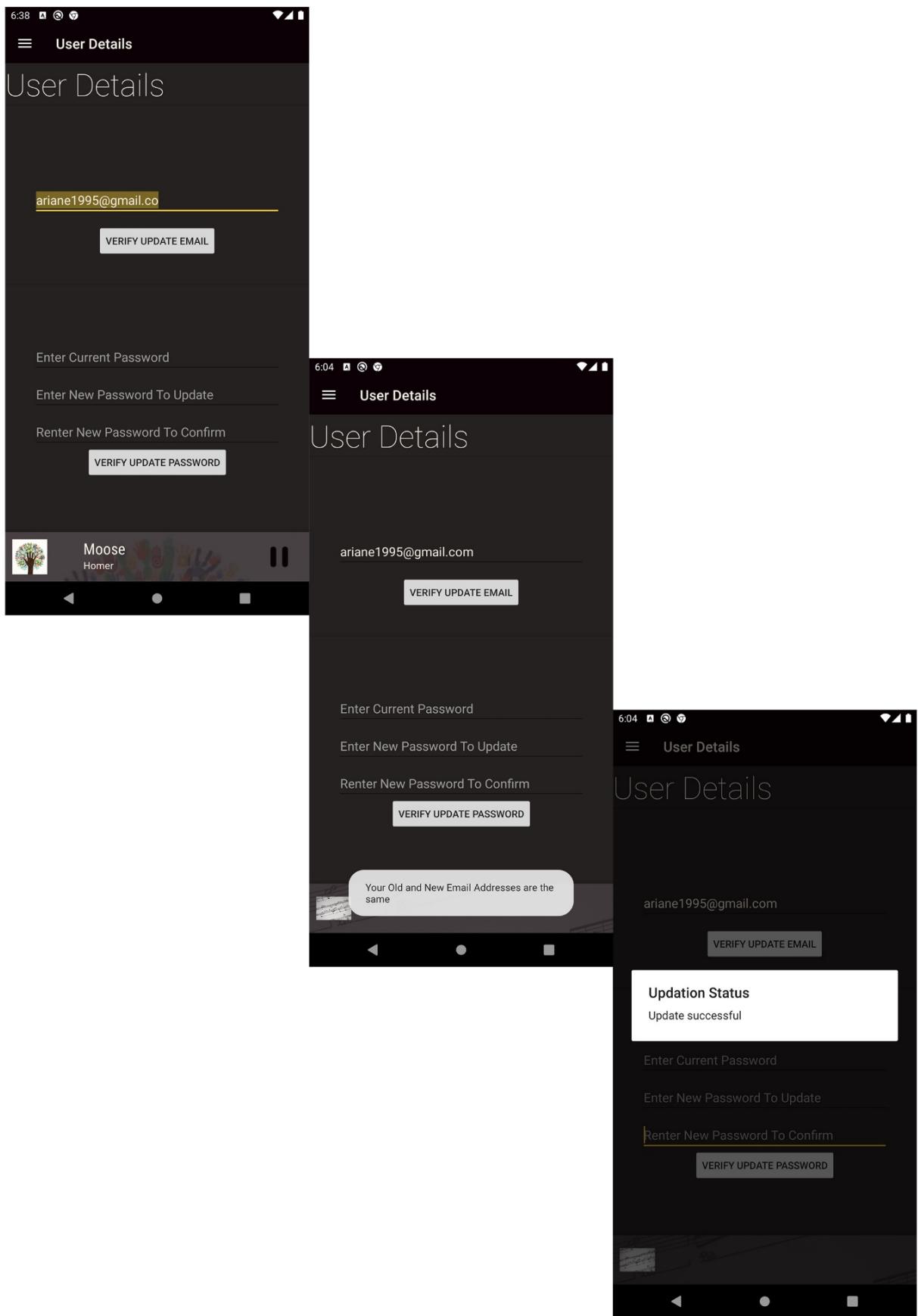


Random / Shuffle

Music streaming from a web-server to an Android app and Website.



Music streaming from a web-server to an Android app and Website.



CHAPTER 11

CONCLUSION

Limitations

- Our server isn't considered as a Streaming quality server.
- We aren't using streaming dedicated solutions.
- Some bugs may be present, the project lacks a few advanced functions.
- Software can be optimised on both the web (At least the front end) and Android platform.

Application and Future Scope

- Goal of the project was to create a music streaming Android app, website, with server.
- This was demonstrational project, and hence the goal was achieved.
- This can be either applied to stream music which is open source or is declared CC0.
- We plan to advance this project, and add audiobooks from services like, LibriVox, who produce audio narration for literature which is copyright free, or copyright expired, or is available in public domain. They largely focus on AudioBooks.
- Because of democratisation of technology, any one with interest, internet, and a Computer can produce music. There are many new and upcoming artist who rather prefer to keep their music compositions and tracks without any limitation of royalty.

Conclusion

- While we haven't made our project, in such a way that it can be considered as a final production grade software.
- This project can be considered as an early stage under development project. And can be used by users, although there is possibility of unstable performance and efficiency loss.

CHAPTER 12

REFERENCE AND BIBLIOGRAPHY

1. <https://blog.hubspot.com/marketing/history-of-internet-radio>
2. <https://cdn.statcdn.com/Infographic/images/normal/5152.jpeg>
3. <https://comingsoonwp.com/000webhost-review/>
4. <https://en.wikipedia.org/w/index.php?cirrusUserTesting=control&search=Timeline+of+music+streaming&title=Special%3ASearch&go=Go&ns0=1>
5. https://en.wikipedia.org/w/index.php?title=Music_streaming_service&redirect=no
6. https://en.wikipedia.org/wiki/Comparison_of_on-demand_music_streaming_services
7. https://en.wikipedia.org/wiki/Streaming_media
8. <https://idapgroup.com/blog/much-cost-build-app-like-spotify/>
9. <https://in.000webhost.com/features>
10. <https://inc42.com/resources/hottest-music-streaming-apps-in-india-2020/>
11. <https://leftronic.com/android-vs-ios-market-share/#:~:text=Google%20Android%20and%20Apple%20iOS,share%20in%202019%20is%2022.17%25.&text=Between%202012%20and%202019%2C%20the,smartphone%20users%20grew%20by%202020%25.>
12. <https://musically.com/2019/12/09/report-spotify-has-36-market-share-of-music-streaming-subs/>
13. <https://musically.com/2020/02/19/spotify-apple-how-many-users-big-music-streaming-services/>
14. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2149853
15. <https://qz.com/india/1573236/youtube-music-spotify-jiosaavn-spice-up-indian-music-streaming/>
16. <https://reports.fipp.com/partner/fipp/global-digital-subscriptions-snapshot-november-2019-update-535?t=1>
17. <https://soundcharts.com/blog/how-music-streaming-works-trends>
18. <https://soundcharts.com/blog/how-music-streaming-works-trends#:~:text=Music%20streaming%20works%20in%20a,seconds%20before%20playing%20a%20song.>
19. <https://theappsolutions.com/blog/development/music-app-development/>

Music streaming from a web-server to an Android app and Website.

20. <https://thenextweb.com/in/2020/02/07/indias-streaming-music-market-is-now-200-million-strong-but-hardly-anyones-paying-for-it/>
21. <https://www.cleveroad.com/blog/how-to-create-a-music-streaming-app>
22. <https://www.cleveroad.com/blog/how-to-make-a-spotify-app--look-inside-a-spotify-app-and-find-out-the-cost-of-app-development>
23. <https://www.dynaudio.com/dynaudio-academy/2018/may/a-history-of-music-streaming>
24. <https://www.gizbot.com/apps/features/8-best-music-streaming-app-in-india-that-offers-the-best-music-listening-experience-065853.html>
25. <https://www.google.com/search?q=india+music+streaming+services&oq=india+music+streaming&aqs=chrome.3.69i57j0l7.4651j0j7&sourceid=chrome&ie=UTF-8>
26. <https://www.innofied.com/create-a-music-streaming-app-like-spotify/>
27. <https://www.investopedia.com/terms/n/network-effect.asp>
28. <https://www.pcmag.com/encyclopedia/term/music-streaming-service>
29. https://www.researchgate.net/publication/309394769_Psychological_ownership_and_music_streaming_consumption
30. https://www.researchgate.net/publication/321111230_ANALYSING_THE_STREAMING_SERVICES_IN_THE_MUSIC_INDUSTRY
31. <https://www.statista.com/chart/20826/music-streaming-services-with-most-subscribers-global-fipp/>
32. <https://www.statista.com/chart/4713/global-recorded-music-industry-revenues/>
33. <https://www.statista.com/chart/5152/music-streaming-subscribers/>
34. <https://www.statista.com/outlook/209/100/music-streaming/worldwide>
35. <https://www.sutori.com/story/history-of-music-streaming--zqSr2qQSuWhuDsXbmAUxfi6Y>
36. <https://www.tandfonline.com/doi/full/10.1080/15405702.2019.1701675>
37. <https://www.technavio.com/report/music-streaming-service-market-industry-analysis?tnplus>
 - a. Music Streaming Service Market by End-users, Streaming Service, and Geography - Global Forecast and Analysis 2019-2023
 - b. Published: Aug 2019
 - c. Pages: 128
 - d. SKU: IRTNTR31904
38. <https://www.tutorialspoint.com/sdlc>

Music streaming from a web-server to an Android app and Website.

39. https://www.tutorialspoint.com/sdlc/sdlc_overview.htm
40. <https://www.unifiedinfotech.net/blog/create-a-music-streaming-app-like-spotify/>
41. <https://yalantis.com/blog/music-apps-market-experience-music-apps-development/>