

**A
Project Report
On
ONDEMAND MUSIC (SELF-DJ) SYSTEM
BTECH-SEM VII**

**PREPARED BY
PATEL PARTH N. (IT-080)
PATEL JAY P. (IT-076)**



**DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF TECHNOLOGY,
DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001**

December, 2019

**A
Project Report
On**

ONDEMAND MUSIC (SELF-DJ) SYSTEM

**In partial fulfillment of requirements for
System Design Practice**

BTECH, SEM VII

Submitted By:

PATEL PARTH N.

PATEL JAY P.

Under the Guidance of

Prof. (Dr.) M. M. Goswami.



DEPARTMENT OF INFORMATION TECHNOLOGY

**FACULTY OF TECHNOLOGY, DHARMSINH DESAI UNIVERSITY
COLLEGE ROAD, NADIAD- 387001**

CANDIDATE'S DECLARATION

We declare that pre-final semester report entitled “Ondemand Music (Self-DJ) System” is our work conducted under the supervision of the guide Prof. (Dr.) M. M. Goswami.

We further declare that to the best of our knowledge the report for B.Tech VII semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate's Signature

Candidate's Name: Patel Parth N.

Student ID: 16ITUOS092

Candidate's Signature

Candidate's Name: Patel Jay P.

Student ID: 14ITUTS020

Submitted To:

Prof. (Dr.) M. M. Goswami.

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Gujarat.

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT



CERTIFICATE

This is to certify that the project carried out in the subject of Software Design Project, entitled “Ondemand Music (Self-DJ) System” and recorded in this report is a bonafide report of work of

- | | | |
|---------------------------------|-------------------------------|---------------------------------|
| 1) <u>PATEL PARTH N.</u> | Roll No. <u>IT-080</u> | ID No: <u>16ITUOS092</u> |
| 2) <u>PATEL JAY P.</u> | Roll No. <u>IT-076</u> | ID No: <u>14ITUTS020</u> |

of Department of Information Technology, semester VII. They were involved in Project work during academic year 2019 -2020.

Prof. (Dr.) M. M. Goswami
(Project Guide),
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date: 19th OCT. 2019

Prof. (Dr.) V. K. Dabhi,
Head, Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad.
Date: 19th OCT. 2019

ACKNOWLEDGMENT

We would like to express our profound gratitude to our project guide, Prof. (Dr.) M. M. Goswami, as well as our head of department, Prof. (Dr.) Vipul Dabhi, who gave us this resplendent opportunity to work on this project. This project being the first one undertaken by us was an unforgettable and educative experience. We take this opportunity to thank all those who have generously helped us to give a proper shape to our work. Our sincerest appreciation must be extended to Faculty of Technology, DDU, Nadiad. We also want to thanks faculties of the college; they have been very kind & helpful to us.

WITH REGARDS

PATEL PARTH N.

PATEL JAY P.

Table of Contents

Abstract.....	i
List of Figures	ii
List of Tables	iii
1.0 INTRODUCTION	1
1.1 Project Details	1
1.2 Purpose	1
1.3 Scope	1
1.4 Objectives	1
1.5 Technology and Literature Review	2
2.0 PROJECT MANAGEMENT	3
2.1 Feasibility Study	3
2.1.1 Technical Feasibility	3
2.1.2 Time Schedule Feasibility	3
2.1.3 Operational Feasibility	3
2.1.4 Implementation Feasibility	3
2.2 Project Planning	4
2.2.1 Project Development Approach and Justification	4
2.2.2 Project Plan	5
2.2.3 Milestones and Deliverables	6
2.2.4 Roles and Responsibilities	7
2.3 Project Scheduling chart	7
3.0 SYSTEM REQUIRMENTT STUDY	8
3.1 Study of Current System	8
3.2 Problems and Weaknesses of Current System	8
3.3 User Characteristics (Type of users who is dealing with the system)	8
3.4 Hardware and Software Requirements	8
3.4.1 Hardware Requirement	8
3.4.2 Software Requirement	8
3.5 Constraints	9
3.5.1 Hardware Limitations	9

3.5.2 Interfaces to Other Applications	9
3.5.3 Reliability Requirements	9
3.5.4 Criticality of the Application	9
3.5.5 Safety and Security Consideration	9
4.0 SYSTEM ANALYSIS	10
4.1 Requirements of New System	10
4.2 Features of New System	10
4.3 Class Diagram	11
4.4 Use case Diagram	12
5.0 SYSTEM DESIGN	14
5.1 System Architecture Design.....	14
5.1.1 Customer Activity	14
5.1.2 Supplier Activity	15
5.1.3 Sequence Diagram(Supplier side).....	16
5.1.4 Sequence Diagram(Customer side)	17
5.1.5 Deployment Diagram	18
5.2 Input/output and Interface Design	18
6.0 IMPLEMENTATION ENVIRONMENT	20
6.1 Implementation Environment.....	20
6.2 Program/Modules Specification	22
6.3 Coding Standards	22
7.0 TESTING	23
7.1 Testing Plan.....	23
7.2 Testing Strategy	23
7.3 Testing Methods	23
7.4 Test Cases	24
8.0 USER MANUAL	25
9.0 LIMITATION AND FUTURE ENHANCEMENT	29
9.1 Limitations	29
9.2 Future Enhancement	29

10.0 CONCLUSION AND DISCUSSION	30
10.1 Conclusion.....	30
10.2 Discussion	30
10.2.1 Self Analysis of Project Viabilities	30
10.2.2 Problems Encountered and Possible Solutions	30
10.2.3 Summary of Project Work	30
 REFERENCES	 31

ABSTRACT

The purpose of Music app is to play, upload and up-vote Music songs according to one's choice efficiently. Simply our app consist data of both user & admin franchise who provide this type of service. So we can say that in our app main two modules one is end-user & second is for the franchise. We had performed this in android studio for front end & controller and firebase for database or backend part. The first phase is designing which shows the way application will work.

Next phase is implementation. Application contains login and logout functionality, registration for user side, forgot password functionality, Admin side uploading songs and creating album, contains information of both, etc. To Store data about songs ad new registered user, we are going to store them in real-time database which is provided by firebase. Our app is very easy to learn & use.

After completion of implementation we have to check some test cases. According to that if modification required then we have to modify in our app. It has built-in database connection modules which makes it easy to connect to the database.

LIST OF FIGURES

Figure	Page No.
Fig 2.1 Iterative Waterfall Model	7
Fig 4.3 Class Diagram	18
Fig 4.4 Use case Diagram	19
Fig 5.1 Client Activity	20
Fig 5.2 Admin Activity	21
Fig 5.3 Admin side Sequence Diagram	22
Fig 5.4 Client side Sequence Diagram	23
Fig 5.5 Deployment Diagram	24
Fig 5.6 Three Tier Architecture	24
Fig 8.1 Sign in Page	31
Fig 8.2 Sign-Up Page	32
Fig 8.3 Client GUI	33
Fig 8.4 Upload song(Admin side)	34
Fig 8.5 List of Songs(Admin side)	35
Fig 8.6 List of Admin	36
Fig 8.7 List of Album	37
Fig 8.8 Home Page(Supplier side)	38
Fig 8.9 Other options(Supplier side)	39
Fig 8.10 Requested order in customer side	39
Fig 8.11 Data analysis	39

LIST OF TABLES

Table	Page No.
Table 2.1 Milestones and Deliverables	9
Table 2.2 Project Scheduling	11
Table 7.1 Test Cases	29

1.0 INTRODUCTION

1.1 Project Details

Nowadays, the demand for music is also increasing and people always want something more something new. So, we have implemented ondemand music system app. There is also some application available which is providing Music but all of them are paid such as saavan, gaana, Wink music, etc. But now if you want to hear music online and freely. So, you have to search for our app and download it. You will have unlimited music fun at no cost. There are two types of apps that are available. First one is, for the usage of the end-user who can vote songs dynamically and the second one is, for the admin-franchise that will be able to play songs, create album.

We also provide one functionality in admin-franchise app that is to show the current playing album to the end-user. Based on the user likes of a particular song we will display that particular song on top in admin-franchise and all of these backend work is stored into the database which we are using it as firebase. So, basically all the required functionality is provided by our music app.

1.2 Purpose

The Main purpose of making this app is when we have to listen our favorite song at crowded place then sometimes we may face difficulties in listening music or our favorite music isn't playing after requesting a lot. With the help of our app we had introduced a concept with which we can listen our favorite music at any public place by voting to particular song and according to the likes of the song, it will be displayed on top in to admin app. Finally admin will play the song.

1.3 Scope

One of the key feature is also that end-user can login into our app anytime and also we can create as many number of users as we want to create. We will store all these details into our firebase i.e. backend. We can implement one more functionality that end-user should get the notification upon entering into any new franchise so that user may be aware of our apps facilities and he/she should know how to use it.

There are many functionalities that are to be added to our app. We will be looking further to implement all these functionalities to our app. Also, we will be trying to look forward that our apps should work properly under any condition i.e. apps should be maintained.

1.4 Objectives

The Main objective behind building this app is to provide Online Music to each and every individual registered or logged-in to our app. Also, it should provide these basic facilities not only to end-users but also to different franchise owner also. So, according to us, this is a very good concept into the market. With the help of this concept each and every user can up-vote his or her own song. The songs are to be displayed in albums and currently playing album will be displayed to end-user

automatically as he/she enters the franchise. User also has lot of friends as well who can use this system and can listen to their favorite music any time and in any franchise.

1.5 Technology and Literature Reviews:

Off-course, without the help of new technologies we would not be able to develop such kind of project. For such kind of project Android studio is required and also to maintain the lots of data we will be using firebase as backend technology.

Front-end Technology: Android Studio

Most of the front-end part of our project is done in android such as login page, logout page, main page, albums page, songs page, list of franchise page, etc. all of these pages are developed in android both xml file and .java file for each page.

We came across many errors in android studio but hopefully with the help of stack-overflow we are successfully be able to resolve these errors such as android manifestation merger failed, android resource linking failed, etc.

Back-end Technology: Firebase (Provided by google)

To Store the data used by the application we use firebase as back-end technology. Firebase provides Authentication of the application once the application is connected to the firebase. Also, the main feature it provide is of Real-time database to store the user email and all other details of user which is taken during registration, Also, it will store songs which are the most important element of our project under the storage section of firebase.

2.0 PROJECT MANAGEMENT

2.1 Feasibility Study

2.1.1 Technical Feasibility

Technical analysis evaluates technical merits of the system at the same time collects additional information about performance, reliability, maintainability and productivity. The technical feasibility means that the project can be done with the current equipment, existing software technology and the current knowledge. We use Android Studio for front-end purpose and Google's firebase for back-end purpose as technical tools.

2.1.2 Time Schedule Feasibility

The Application has simple working and the basic requirement like storing songs, registration new user, displaying list of franchise to user and displaying list of albums and in each album displaying list of songs, feedback from user side can be satisfied within the allotted time period. All of these things can be completed in a particular time-period. So, the project is feasible can be completed before the deadline.

2.1.3 Operational Feasibility

Two kinds of operation will be performed by our application. 1] It will be performed on client side. Here the main operation is when the user registers or log-ins into our app it should provide like song functionality. 2] It will be on admin side. When admin login into his app it should provide list of album created by that admin only and in that particular album only some songs are to be displayed. Also it should provide upload functionality in each and every Album that admin creates. Finally songs should be able to play is the main operation of our app.

2.1.4 Implementation Feasibility

The Application is easily implemented by use of Android studio which is very easy to use and secure. The Backend firebase is also very easy to understand and works very well with android studio. The connection of Android Studio and Firebase is also very secure and can be easily done by some simple steps. This easily connect our apps. Firebase is connected to client app as well as admin app. Firebase is open source database & connectivity is very simple compare to other databases. Basically Implementation is easy.

2.2 Project Planning

2.2.1 Project Development Approach and Justification:

For Project Development Iterative-Waterfall Model is used.

Iterative waterfall model:

The Iterative water fall model approach overcomes the problems associated with the waterfall model approach. If any difficulty or problem encounter in any phase may require going back to the previous phase and performing the required modifications and proceeds sequentially. This backtracking allows modifying any corrections or modifications required in the previous phase.

As illustrated in Fig2.1, this model divides the cycle into the phases mentioned below:

1. Feasibility Study.
2. Requirements analysis and specification.
3. Design.
4. Coding and Unit Testing.
5. Integration and System Testing.
6. Maintenance.

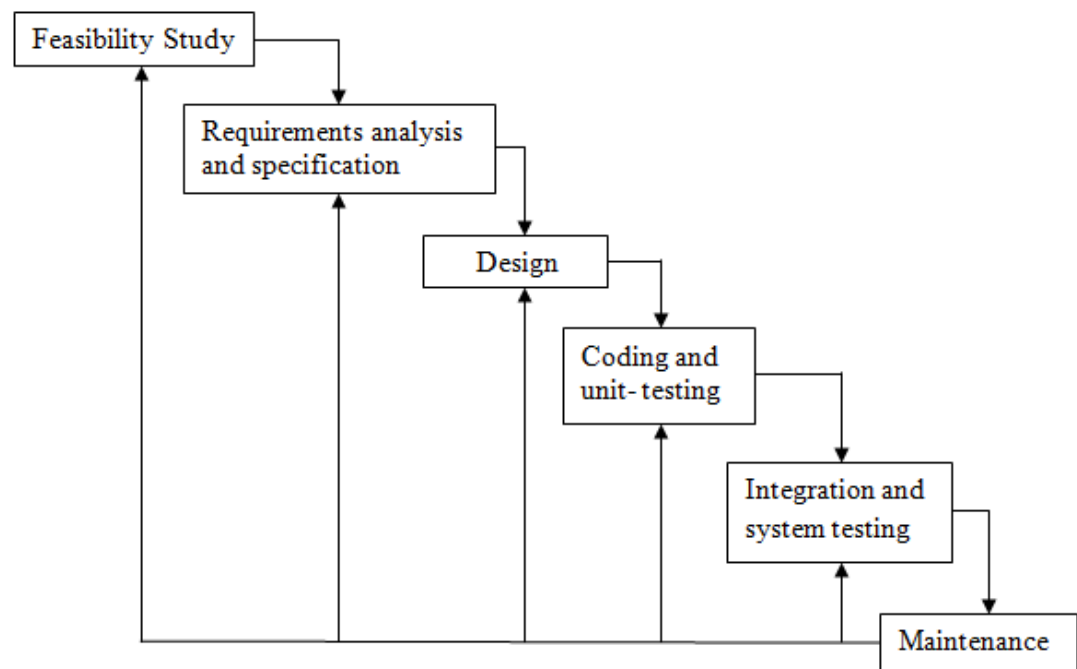


Fig2.1 Iterative Waterfall Model

Advantages of using Iterative Waterfall Model:

- You are provided the chance to see the potential outcomes of every stage and make changes to areas of concern if necessary. This is one of the reasons that make the iterative model useful.
- Iterative development is more adjustable to changes as it considers each stage like a vital portion of the development cycle.
- The time spent on each successive interval maybe lessened depending on how the last stage went and what knowledge was gained from past stages. The system therefore grows through adding new functionalities in the development part of all iterations.

Disadvantages of using Iterative Waterfall Model:

- When using the iterative model people working on the project can get stuck in a loop. Always finding problems than having to go back and design a fix, implement it, than test the system again and finding another problem can mean that the project can run over time and budget.
- Informal requests for improvement after each phase may lead to confusion and may also create scope creep, since user feedback following each phase may lead to increased customer demands. As users see the system develop, they may realize the potential of other system capabilities which would enhance their work, this can be an advantage as much as it can be a disadvantage.

Justification:

After feasibility study as the functional requirements are almost clear, but in some cases like implementing, designing as well as in testing time error can occur. Here we have to decompose the system in to modules. That is why we decide to use iterative-waterfall model which is most suitable model here i.e. if we find any difficulty in coding and testing a modification in design can be done easily.

2.2.2 Project Plan

After feasibility study as the functional requirements are almost clear which are decided by our project lead. After analyzing and thoroughly understanding the requirements of the application we planned the project.

2-tier architecture is used for this System. Here we have to decompose the system in to modules. Also the internals of the individual modules are designed other details. Coding and Unit Testing phase is required to translate the software design into source code. Integration and System Testing phase consists of the integration of the modules in a planned manner. Here during each integration step we have to test the partially integrated system. Finally, when all the modules are successfully integrated and tested, system testing is carried out successfully.

2.2.3 Milestones and Deliverables

Timely directions are always required to run a project successfully. Milestones tell the developers how far he has reached and also tell him what things are still left and how to full-fill them. Milestones may be the short report of achievement n project activity that are used by the project manager to check project progress but which are not delivered to the Clients. The deliverables are the project results that are provided to the customer. Itis usually delivered at the end of some major project phases.

MILESTONES	DELIVERABLES	PURPOSE
Software Installation and Understanding of Technology.	Had complete knowledge of Firebase and Android Studio with its features.	To be familiar with Android Studio and Firebase.
System feasibility study, Requirement and Analysis.	Functional Specifications. Non Functional Specifications.	It gives exact understanding of the User's requirements.
System Design.	Class diagram Sequence Diagram Use Case Diagram Activity Diagram	It gives the logical Structure that describes the system.
Coding and Unit Testing and corrections if any.	Individually Tested and Functional Modules.	It gives the required Module.
Integration and System Testing.	The output obtained for the required functionality after implementing and doing various types of testing.	Integrated System is Ready.

Table 2.1 Milestones and Deliverables

2.2.4 Roles and Responsibilities

As only two members are involve in the team both have to perform all the tasks as the project proceed through its different phases.one member has done designing and functionality of client side while the other has done for Admin side. This helps each one to develop all kinds of skills in all the phases (phases like Designing, Implementation, Testing, Documentation etc.)

2.3 Project Scheduling

Scheduling the project tasks is an important project planning activity. It involves deciding which tasks should be taken up and when. In order to schedule the project activities:

Client app:

- 1] Login/Sign-up page
- 2] Displaying list of franchise
- 3] Displaying list of albums
- 4] Displaying songs
- 5] Implementing likes of Songs

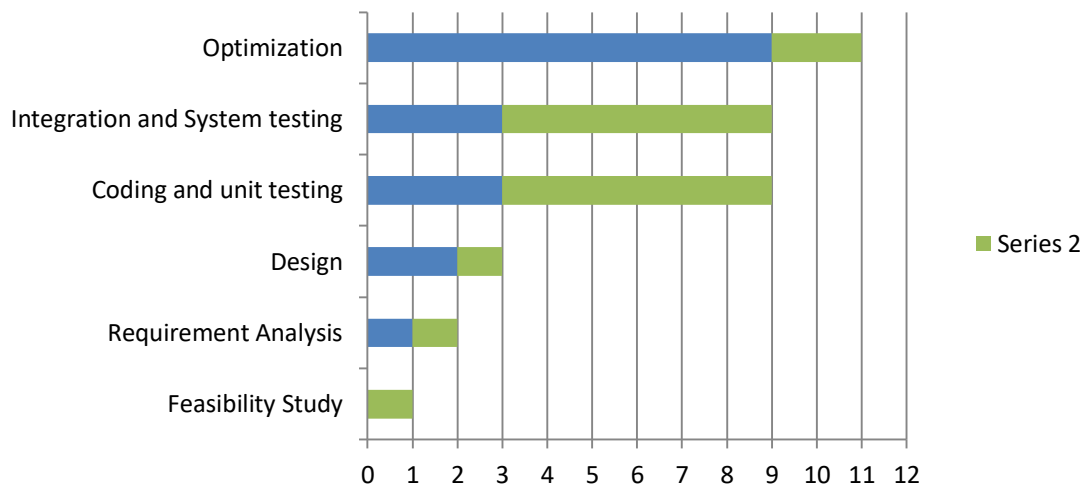


Fig 2.2 Project plan

Week	Task Description
1	Definition Selection
2	Gathering Information and Topic Discussion
3	SRS
4	Installing Android Studio Application
5	Android Authentication and Login Sign-up
6	Display Songs in Client App.
7	Like/Unlike System implementation
8	Playing Songs in Admin app
9	Uploading songs From Admin
10	Displaying Current Playing Album in Client app
11	Displaying List of Admins in Client app

Table 2.2 Project Scheduling

3.0 SYSTEM REQUIREMENTS STUDY

3.1 Study of Current System

Currently, there is no any system that provides Online Music to all the End-users.

3.2 Problems and weakness of current system

Currently, there is no such concept for listening songs in this manner. Apparently, we are facing problems with the Testing face. Application is crashing due to lots of data in our program. Also, application is facing problem with executing and displaying warning that is needed to be resolved. Client app is facing issues with displaying likes of the song in the app. Both client app and admin app, UI needs to be improved. Albums should be displayed in list in both the apps. Client app is taking lot of time after login/registration of user has been done. Admin app also faces the same problem. Also, we have not still used AndroidX in our apps which must be implemented.

3.3 User Characteristics

The major User classes in the System would be:

End-users:

They are essential part of our application. As soon as the, User enters the franchise shop/store he/she gets to likes songs in whichever album it is playing by the franchise store because the franchise store has the Admin app.

Franchise-user (Admin-users):

On the other hand, Admin users will have access to upload song functionality, creating album functionality and playing songs which is displayed on top of particular album. Every album has its own songs and each album contains number of songs.

3.4 Hardware and Software Requirement

3.4.1 Hardware Requirement

- At least 3GB RAM(if 8GB then it is better)
- 1280 x 800 minimum screen resolution
- Intel core i3 processor(if i5 or i7 then it is better)
- Android Mobile(Testing module)
- Microsoft® Windows® 7/8/10 (32- or 64-bit)

3.4.2 Software Requirements

- Android studio IDE
- Java
- Firebase Database
- The user's browser should support HTML5 and JavaScript and cookies must be enabled for a satisfactory user experience.

3.5 Constraints

3.5.1 Hardware limitations

There are no hardware limitations for this system because once the complete System is developed care would be taken while deploying system so necessary pre-requisites are met.

3.5.2 Interface to other applications

There are no other systems that use this application as an interface.

3.5.3 Reliability Requirements

The application does demand much reliability and it is fully assured that the particular information about the user should be secured and flow is maintained and accessed according to the rights.

3.5.4 Criticality of the Application

The application deals with the user's personal tasks so the task and respective details should be highly confidential and in proper flow.

3.5.5 Security and Safety considerations

The system provides a tight security to each and every end-user. Also application generate OTP for verification.

3.6 Assumptions and Dependencies

Assumptions are described as follows:

1. User has sufficient privileges to access internet.
2. Server is running smoothly.
3. Database updates are giving expected and accurate results.

4.0 SYSTEM ANALYSIS

4.1 Requirements of New System (SRS)

R4.1: Client Side Application

Description: Login or Registration (if new user) in to our app and also includes authentication of user and providing user the like functionality for each and every song.

R4.1.1: Login

Description: For Already registered User/Client.

Input: Insert Email Id and Password.

Output: successfully login if it's valid and redirects to home page else display error and redirect to that page only.

Description: By login into our app user can easily managing his Account.

R4.1.2: Forgot password

Input: Enter new Password

Description: New password will be replaced over old password.

Output: successful message on replaced else display error.

R4.1.3: Sign-Up:

Input: Enter details of the name, last name, age, DOB, gender, etc.

Output: Successful message on sign-in else display error and redirect to sign-up page.

Description: it will take some above mentioned details and store it in firebase database.

R4.1.4: List of Admin/Franchise:

Output: List of Admin will be displayed to the client.

Description: For each and every user or client it will display list of admin so that client can select the admin first.

R4.1.5: List of Albums:

Output: Display list of albums.

Description: Now if user has selected admin he/she will select a particular album or current playing album.

R4.1.6: Liking the song:

Output: Song will be liked.

Description: User can be able to like/unlike a particular song.

R4.1.7: Logout:

Description: To log out from panel.

Output: Login Page.

R4.2: Admin Side Application:

Description: performing functionalities like Uploading song, Creating album and playing song.

R4.2.1: Login

Description: For Already registered User/Client.

Input: Insert Email Id and Password.

Output: successfully login if it's valid and redirects to home page else display error and redirect to that page only.

Description: By login into our app user can easily managing his Account.

R4.2.2: Forgot password

Input: Enter new Password

Description: New password will be replaced over old password.

Output: successful message on replaced else display error.

R4.2.3: Sign-Up:

Input: Enter details of the name, last name, age, DOB, gender, etc.

Output: Successful message on sign-in else display error and redirect to sign-up page.

Description: it will take some above mentioned details and store it in firebase database.

R4.2.4: List of Album:

Output: Display list of album for particular Admin.

Description: It will display list of album for particular logged in admin.

R4.2.5: Upload song:

Input: Admin will select song from its internal storage.

Output: Selected song will be Uploaded successfully else display error.

Description: Admin will select song and upload his /her choice songs to particular album.

R4.2.6: Play Song:

Description: Admin will play songs in admin application according to the likes song will be displayed on top list of particular album's particular admin.

Input: Admin will select song.

Output: Selected song will be played.

R4.2.7: Logout:

Description: To log out from panel.

Output: Login Page

4.2 Feature of New System

The application provides tiffin service but as well as some suppliers who provides lunch/dinner at home they can also get it. Application contains day, week and month wise order details.

4.3 Class Diagram

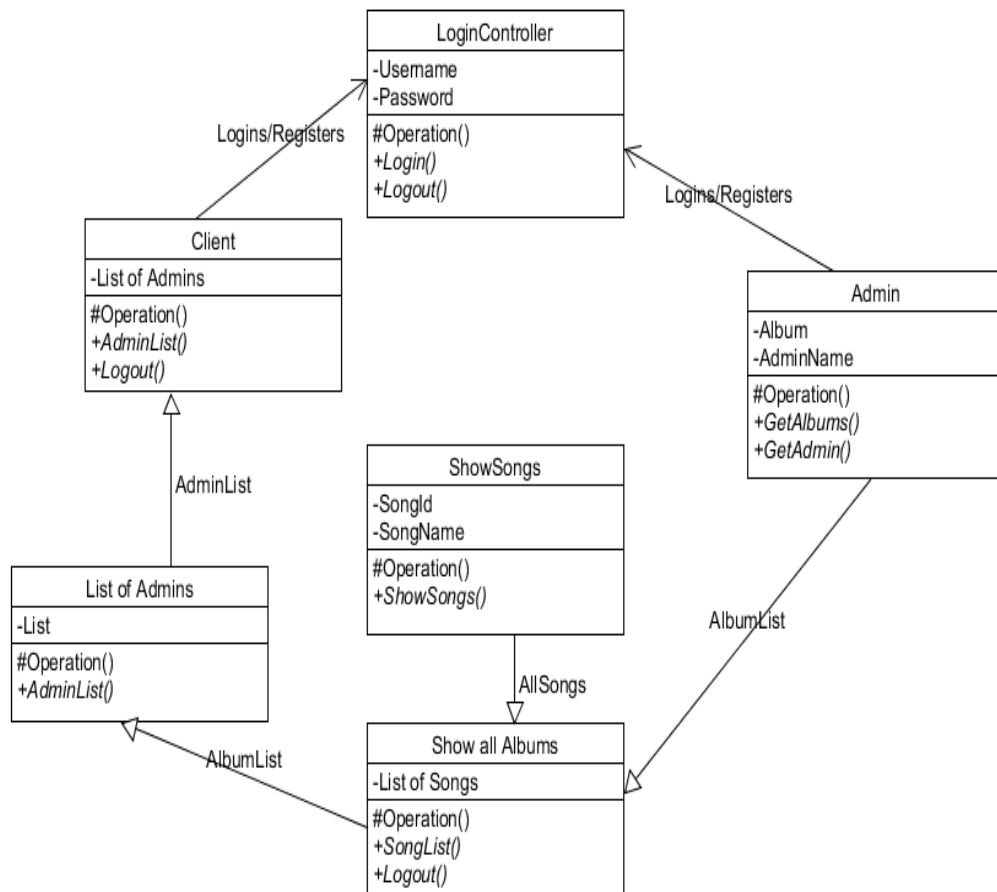


Fig 4.1 Class Diagram

4.4 Use case Diagram

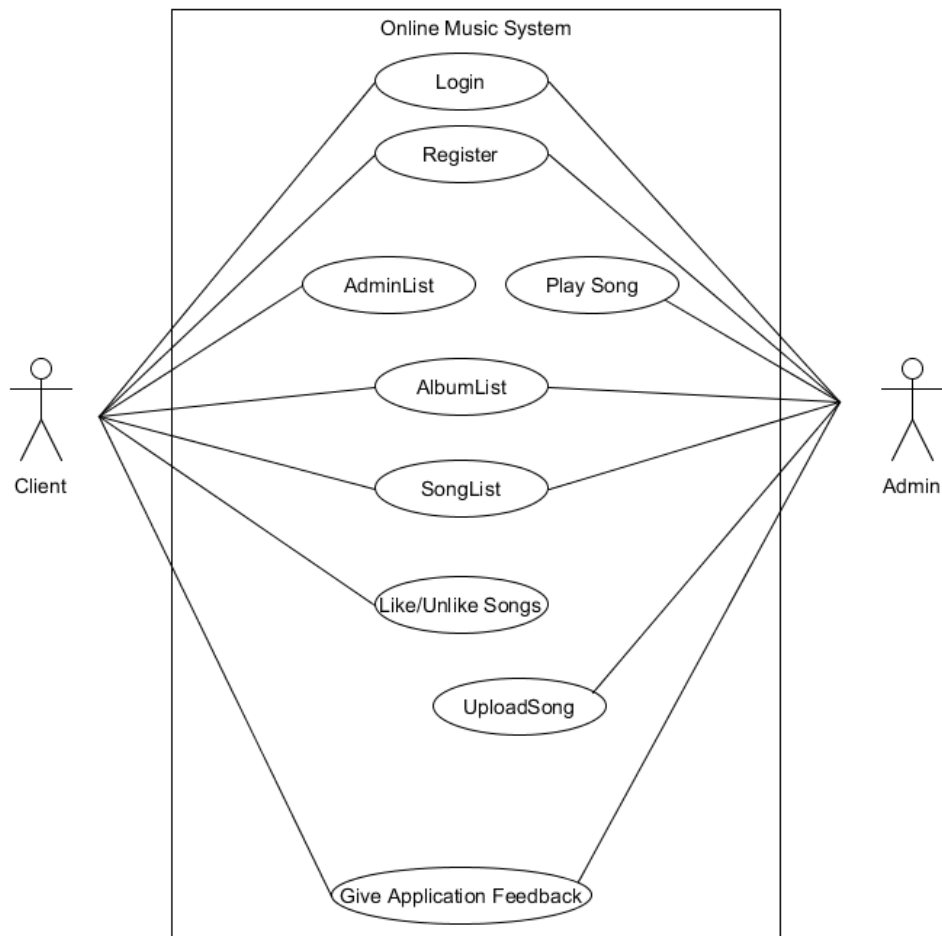


Fig 4.2 Use case Diagram

4.5 E-R Diagram

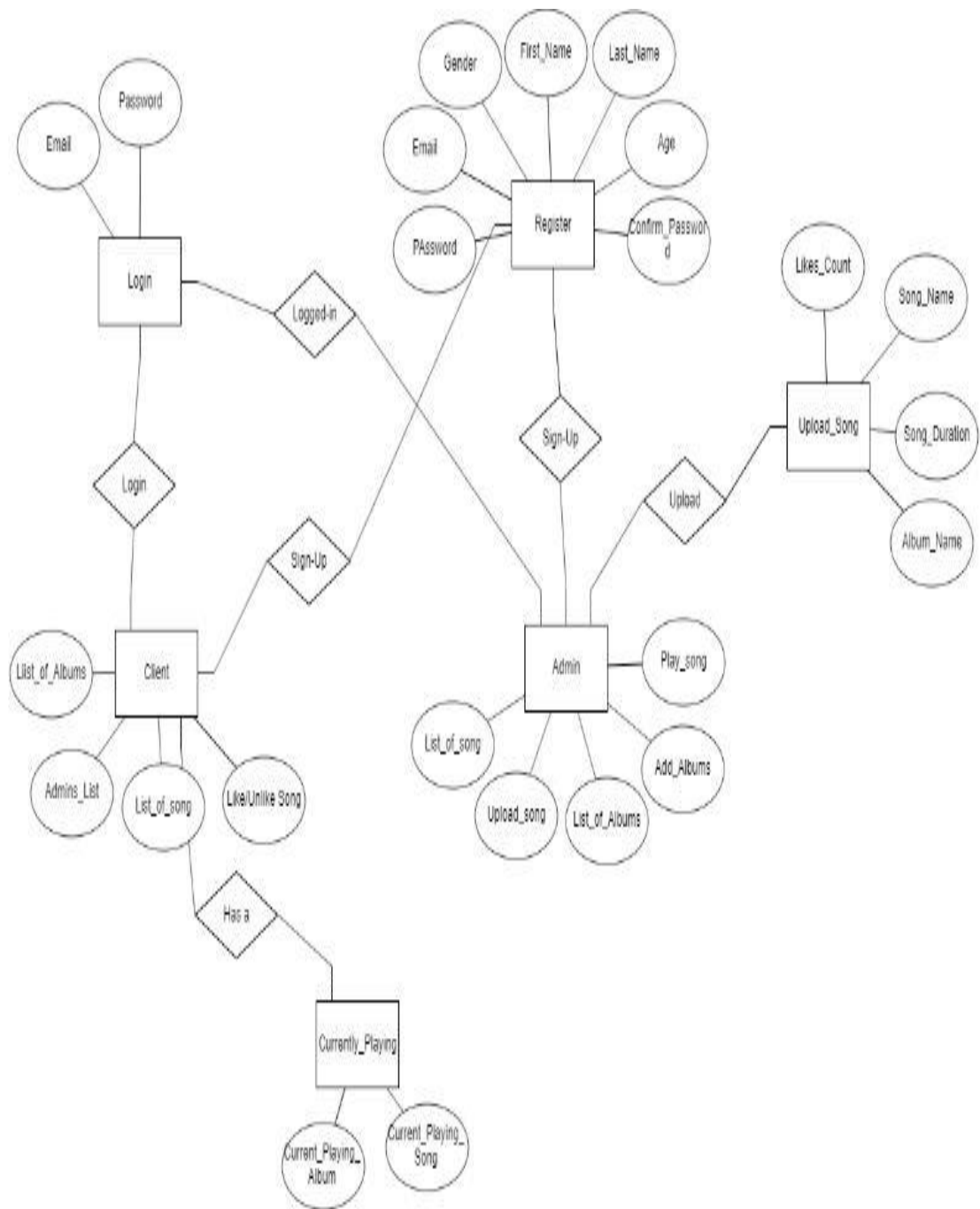


Fig 4.5 E-R Diagram

5.0 SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE DESIGN

5.1.1 CLIENT ACTIVITY

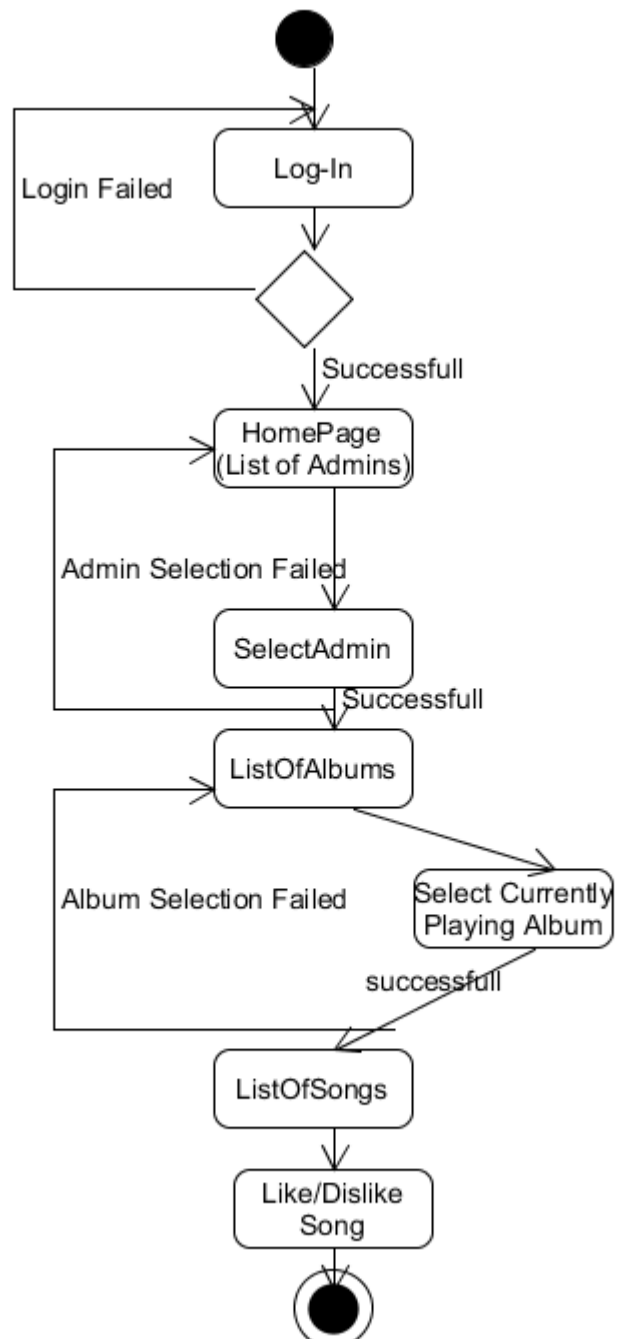


Fig 5.1 Client Activity Diagram

5.1.2 ADMIN ACTIVITY

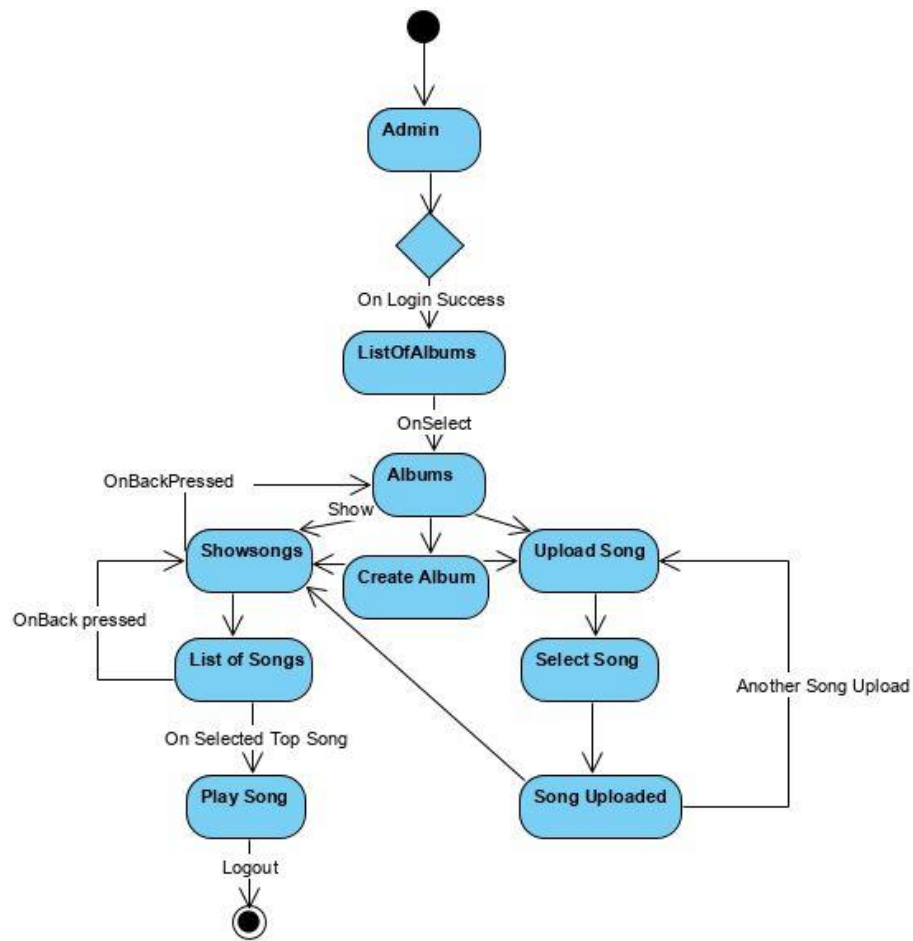


Fig 5.2 Admin Activity Diagram

5.1.3 SEQUENCE DIAGRAM (Admin side)

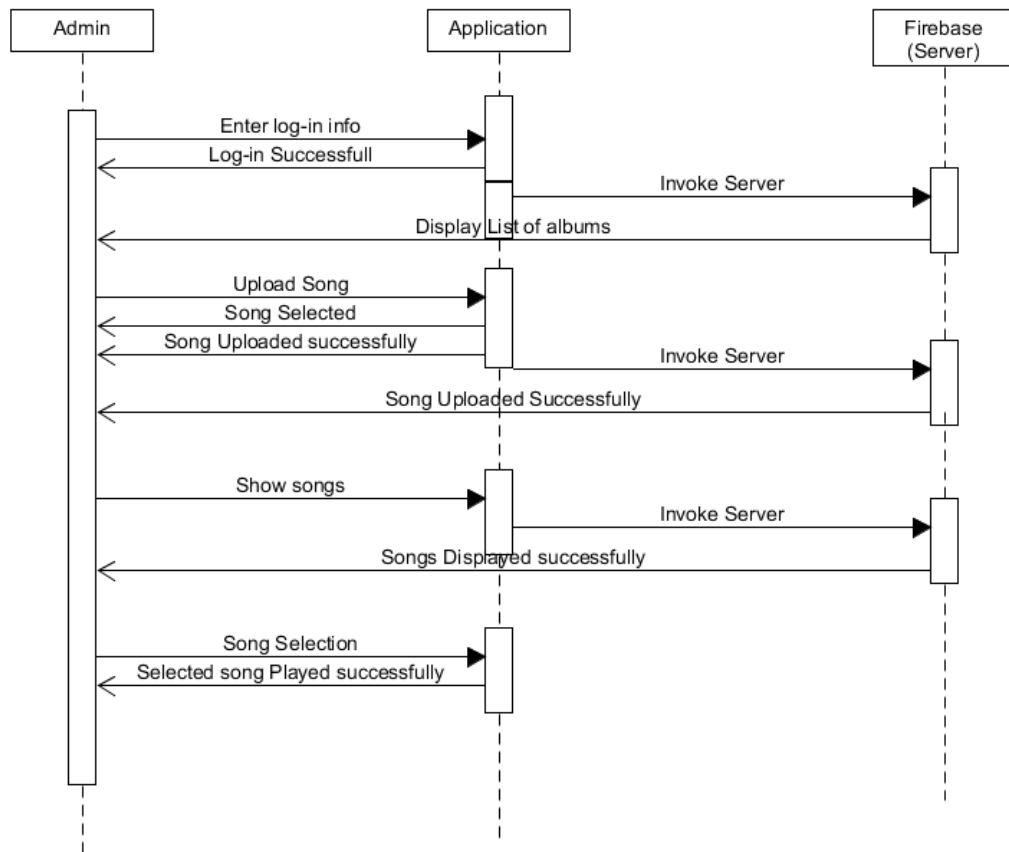


Fig 5.3 Admin Sequence Diagram

5.1.4 SEQUENCE DIAGRAM (Client side)

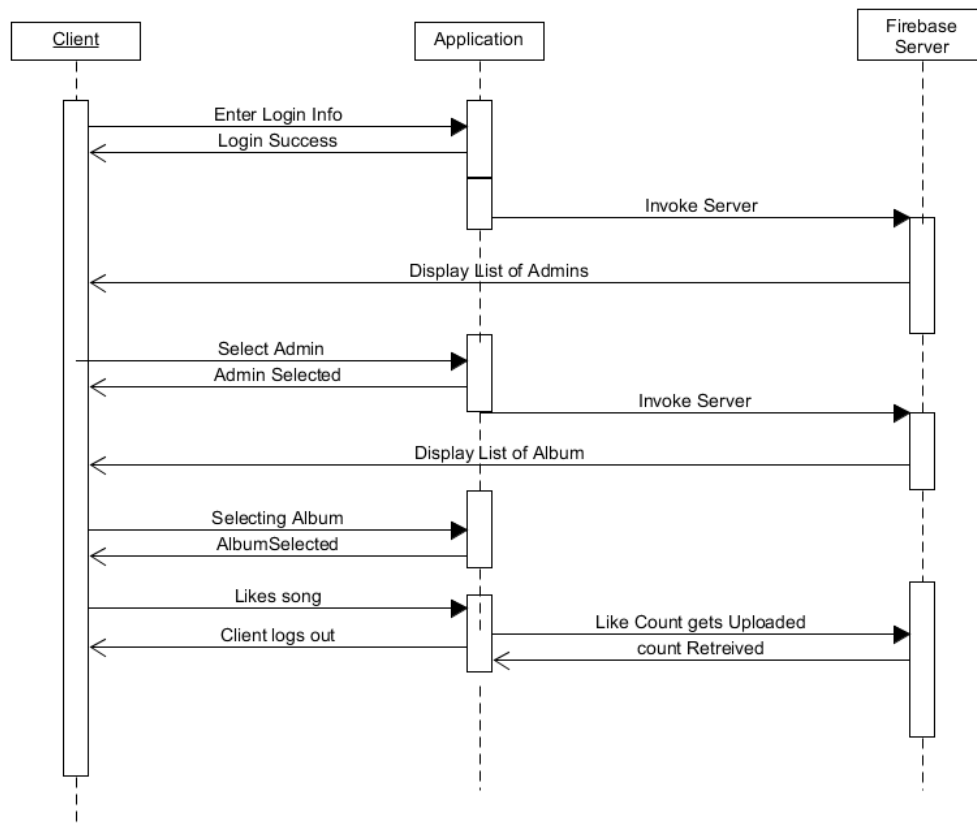


Fig 5.4 Client Sequence Diagram

5.1.5 DEPLOYMENT DIAGRAM

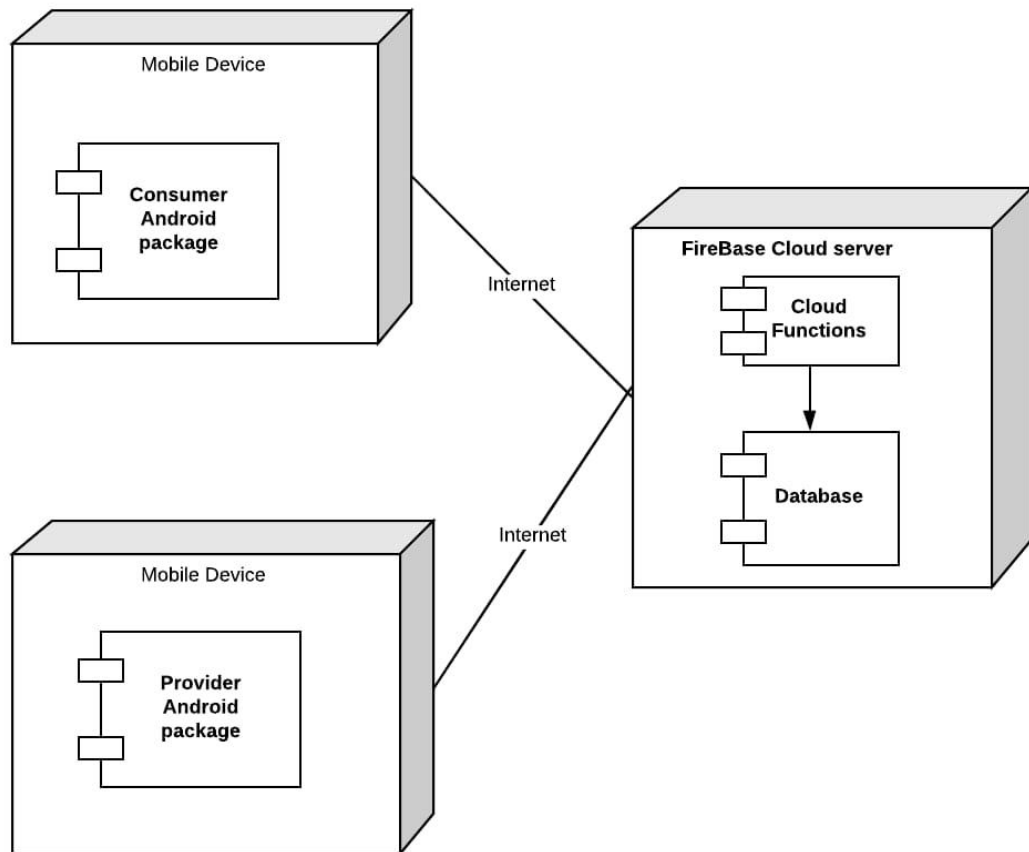


Fig 5.5 Deployment Diagram

5.2 Input/output and Interface Design

The application will follow two-tier architecture. In two-tier architecture, application will run the traditional client/server model but from the web server. The client only displays the GUI and data but has no part in producing results.

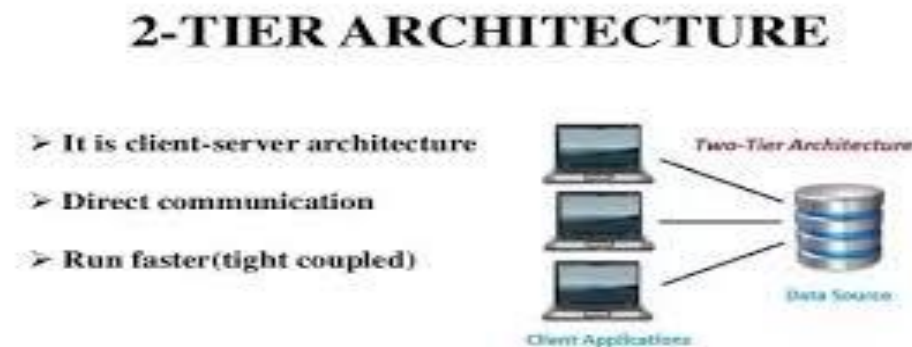


Fig 5.6 Two-tier Architecture

Two-tier architecture will contain the following tiers:

Client/Presentation Tier:

This tier includes all the components present at the client side/presentation side. It is the form which provides the user interface to end user. Programmer uses this tier to get or set the data back and forth. It includes all the application side GUI.

Server/Data Tier:

Data Tier contains methods and classes that deal with passing and storing data to the data Storage Layer. Queries or stored procedures are used to access the data from the database or to perform any operation to the database. It stores the data passed by the presentation. It also includes data stored at firebase (in this case) back-end data.

6.0 IMPLEMENTATION PLANNING

6.1 Implementation Environment

For implementation we have used:

1. Android Studio as Front-End
2. Firebase as Back-End

GUI based vs. non GUI based

In GUI based application it is very easy for user to understand the overall content and the previous and next step of the action. Also GUI based application does not make issues like navigation, ambiguities in data selection and entry.

GUIs have become an established alternative to traditional forms-based user interfaces. GUIs are the assumed user interface for virtually all system development using modern technologies. There are several reasons why we have used GUI based application:

1. GUIs provide the standard look and feel of an application.
2. GUIs are so flexible to use in application areas.
3. The GUI provides seamless integration of custom and package applications.
4. The user has a more natural interface to applications: user understanding is improved.

Our Application is GUI based and multi-user system so multiple users can access the Application in parallel.

In our system all connections with database is not done by the process of designing. There is proper and efficient coding for connection with database. For all functionality of our Application, We have used Data source which help to do connection with database and to retrieve data from the database.

6.2 Program/Modules specification

This system is based on such an architecture that we have created different repository which provides different methods to access the data. Controller handles the request and uses repository classes accordingly to generate the model and pass it to the view which renders and update the user's view. In this way model, view and controller classes together form a system which separates all three of them.

6.3 CODING STANDARDS

To make the system coding easy, easy to remember and reducing the chances of errors some techniques are used at the time of coding of the application which is called coding standard. The coding standard which we adopted during the coding is explained as follows:

Method names are written with words starting in upper case.

Example: "GetAddressByUser()".

Some other Examples are:

- Each nested block should be properly indented and spaced.
- The code should be properly commented for understanding easily. Comments regarding the statements increase the understandability of the code.
- Better to avoid use of digits in variable names.
- The names of the function should be written in camel case starting with small letters.
- The name of the function must describe the reason of using the function clearly and briefly.

7.0 TESTING

7.1 Testing Plan

The testing technique that is going to be used in the project is white box testing. It is a way of testing the software in which the tester has knowledge about the internal structure the code or the program of the software. If third party check our application & testing it then it is known as black box testing because internal details are hidden.

7.2 Testing Strategy

The development process repeats this testing sub-process a number of times for the following phases:

- a) Unit Testing.
- b) Integration Testing.

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. This phase is after the coding phase only.

Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

7.3 Testing Methods

Black box and White box testing:

In black-box testing a software item is viewed as a black box, without knowledge of its internal structure or behavior. Possible input conditions, based on the specifications (and possible sequences of input conditions), are presented as test cases.

In white-box testing knowledge of internal structure and logic is exploited. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing are likely to be found.

7.4 Test Cases

Entity	Test case	Expected output	Actual output	Result
Login	Authentication	Login process successful	Login process successful	PASS
Registration	Authentication	Registration successful	Registration successful	PASS
Like songs from Client app	Like song	Song has been Liked Successfully	Song has been Liked Successfully	PASS
Add Album	Add Album to the Database	Album added successfully	Album added successfully	PASS
Logout	Exit operation	Logout process successful	Logout process successful	PASS
Upload Song	Uploading Song to particular Album	Song Uploaded successfully	Song Uploaded successfully	PASS
Add Admin	Add Admin to the Database	Admin added successfully	Admin added successfully	PASS
Analysis	Data analysis of users, emails, passwords, etc.	Check user info successfully	Check user info successfully	PASS
History	Past records Management	Show past records in display window	Show past records in display window	PASS

Table 7.1 Test Cases

8.0 USER MANUAL

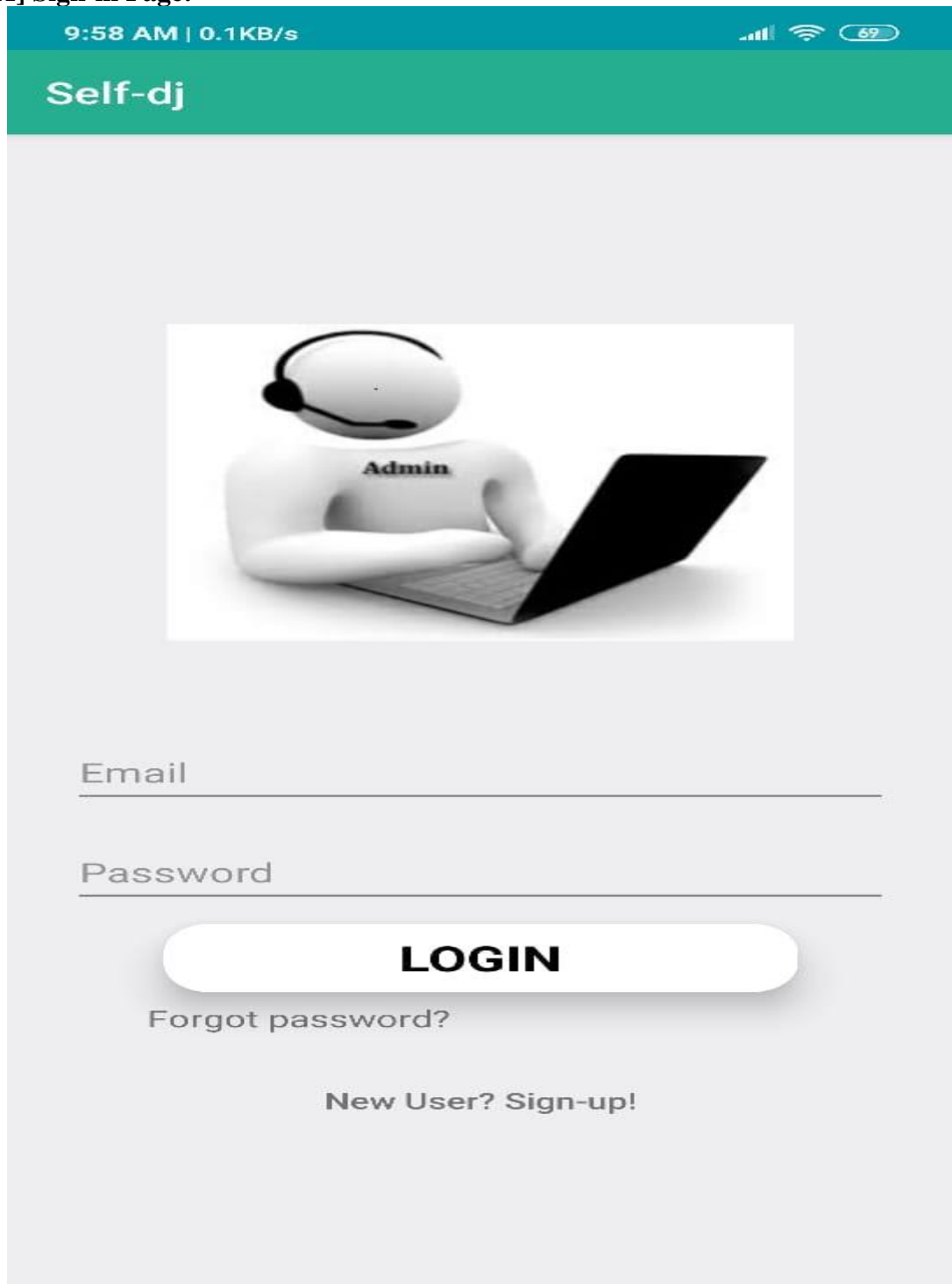
A user guide or user's guide, also commonly known as a manual, is a technical communication document intended to give assistance to people using a particular system.

It is usually written by a technical writer, although user guides are written by programmers, product or project managers, or other technical staff, particularly in smaller companies.

User guides are most commonly associated with electronic goods, computer hardware and software.

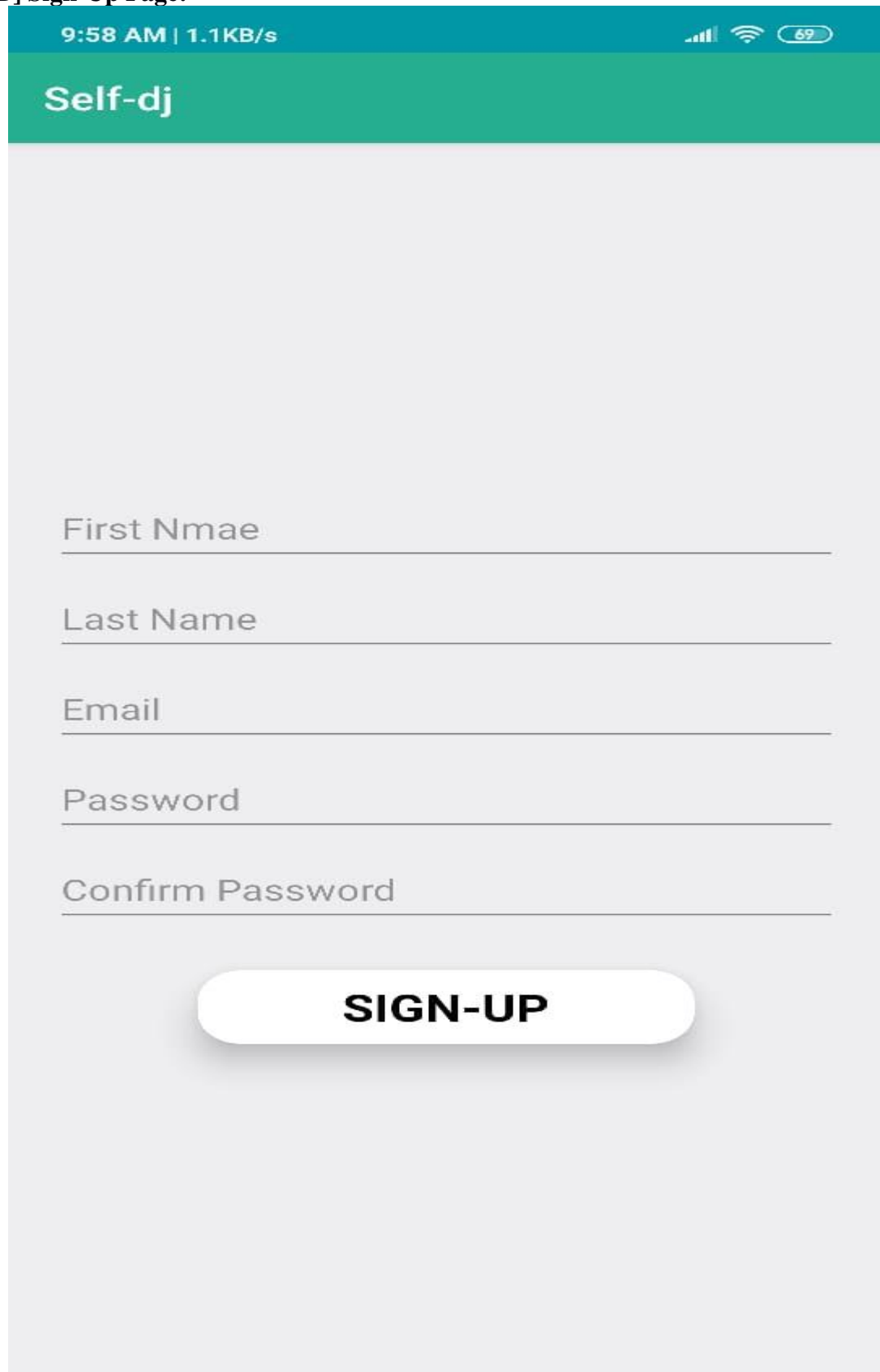
Our user guides contain both a written guide and the associated images. In the case of our application, it is usual to include screenshots of how the program should look. The language used is matched to the intended audience.

A] Sign-in Page:



The screenshot shows a mobile application interface for 'Self-dj'. At the top, a teal status bar displays '9:58 AM | 0.1KB/s' and signal icons. Below this is a green header with the text 'Self-dj'. The main content area is light gray and features a 3D illustration of a white humanoid figure wearing a headset and sitting at a laptop; the figure's chest is labeled 'Admin'. Below the illustration are two input fields: 'Email' and 'Password', each with a horizontal line underneath. A prominent white button with rounded corners and a shadow contains the word 'LOGIN' in bold black capital letters. Underneath the button is the text 'Forgot password?'. At the bottom of the form area is the text 'New User? Sign-up!'.

Fig 8.1 Sign-in page

B) Sign-Up Page:

The screenshot shows a mobile application interface for a sign-up page. At the top, a teal status bar displays '9:58 AM | 1.1KB/s' and signal icons. Below this is a green header bar with the text 'Self-dj'. The main content area is light gray and contains five text input fields with labels: 'First Nmae', 'Last Name', 'Email', 'Password', and 'Confirm Password'. Each label is positioned above its corresponding input line. At the bottom of the form is a white, rounded rectangular button with a shadow, containing the text 'SIGN-UP' in bold black capital letters.

Fig 8.2 Sign-Up page

C] Client GUI:

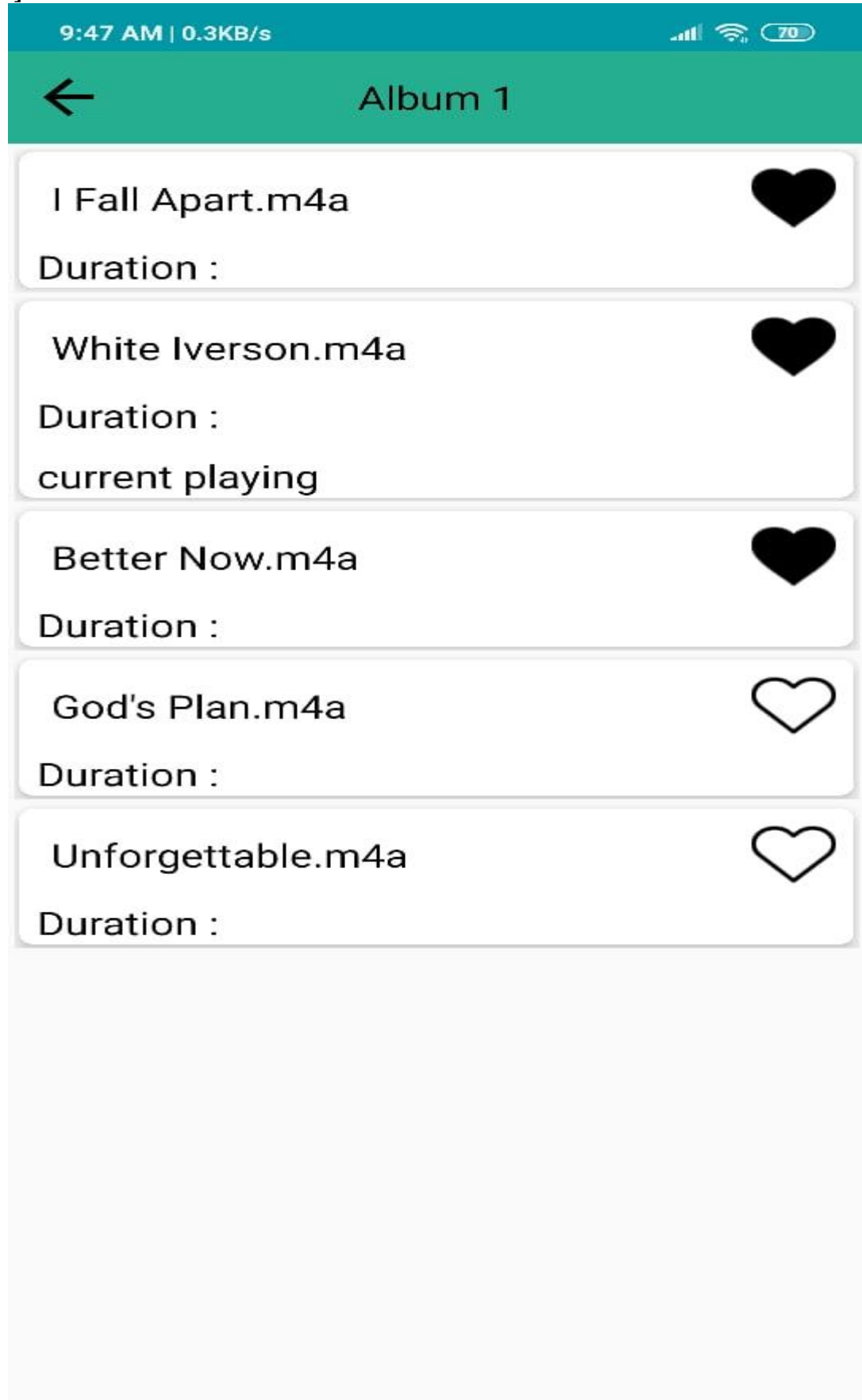
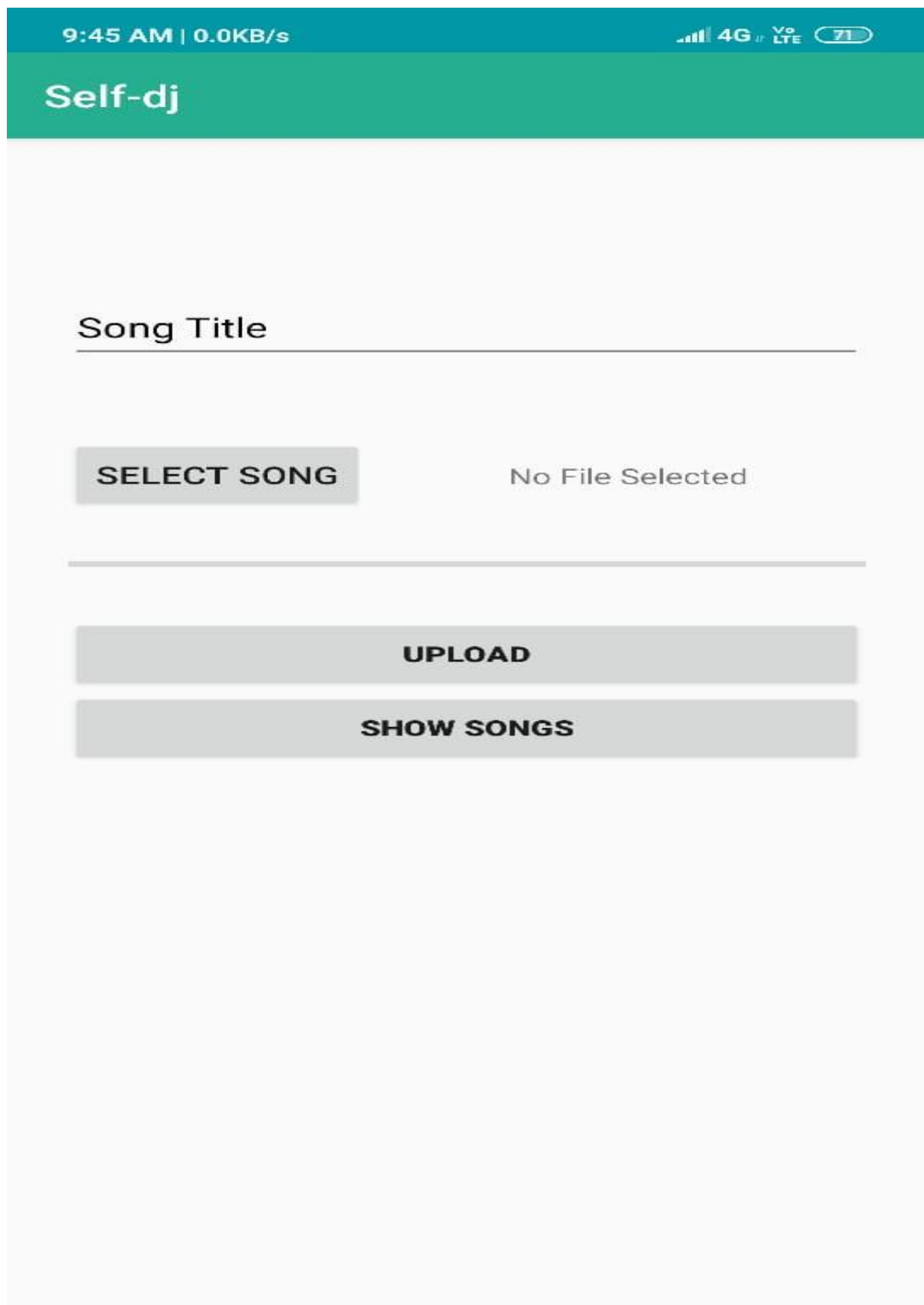


Fig 8.3 Client GUI

DJ Uploading Song:

The screenshot shows a mobile application interface titled "Self-dj". At the top, a status bar displays "9:45 AM | 0.0KB/s", signal strength, "4G LTE", and a battery level of "71". Below the title bar, there is a text input field labeled "Song Title". Underneath this field is a button labeled "SELECT SONG" and the text "No File Selected". A horizontal separator line follows. Below the line are two large, light gray buttons: "UPLOAD" and "SHOW SONGS".

Fig 8.4 Uploading Song

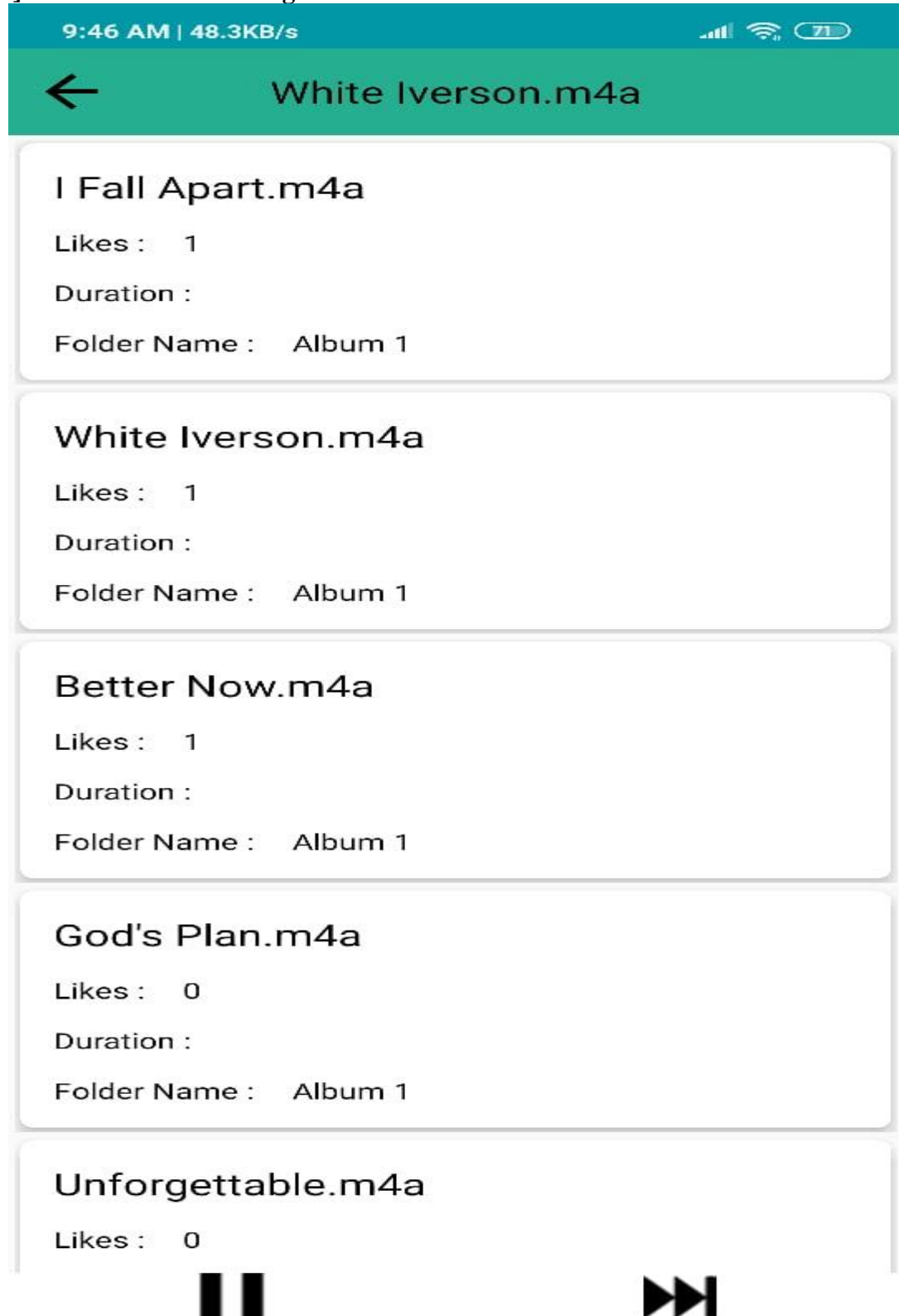
E] Admin side List of songs:

Fig 8.5 List of Songs Admin side

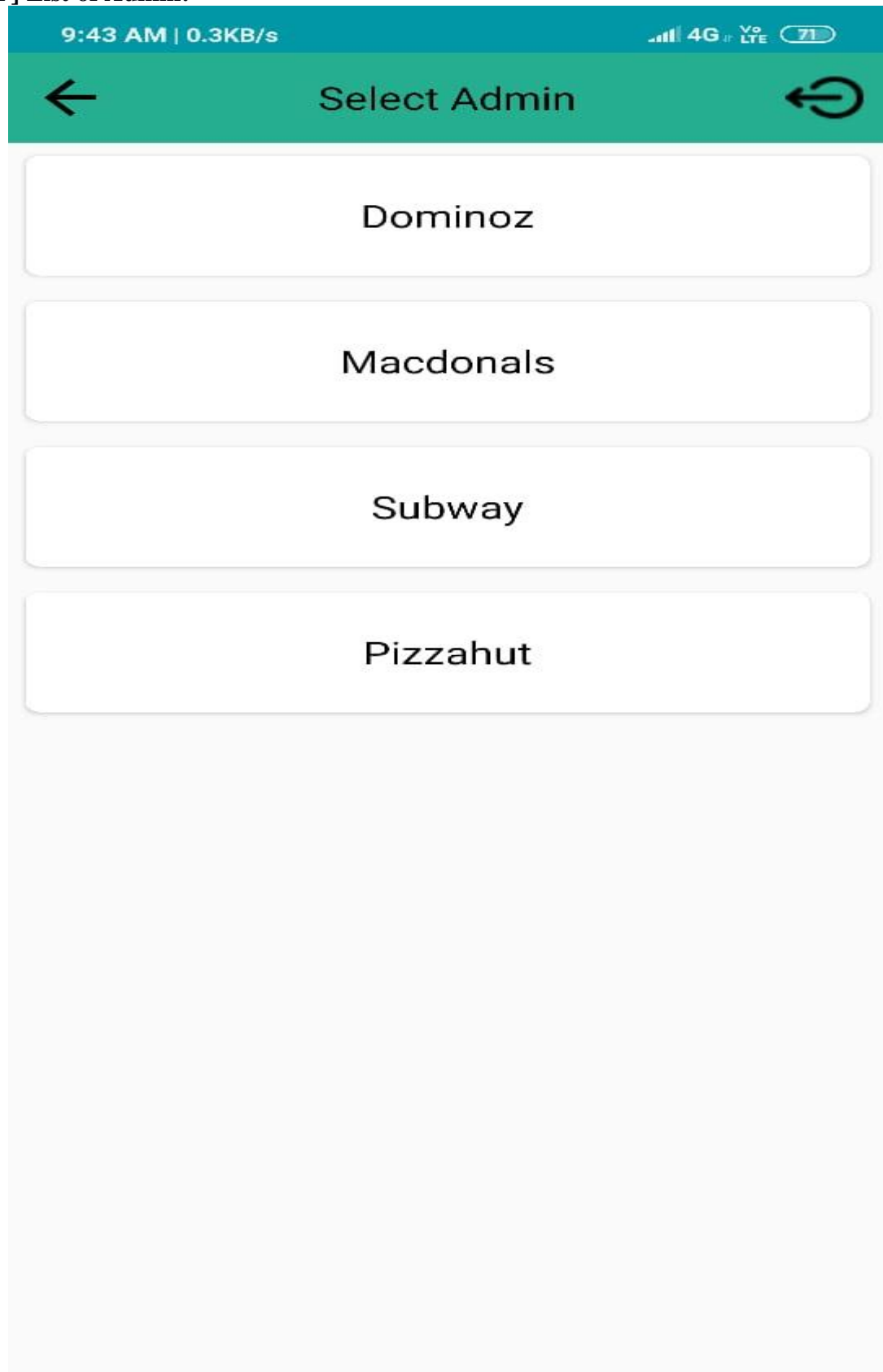
F] List of Admin:

Fig 8.6 List of Admins

G] List of Albums:

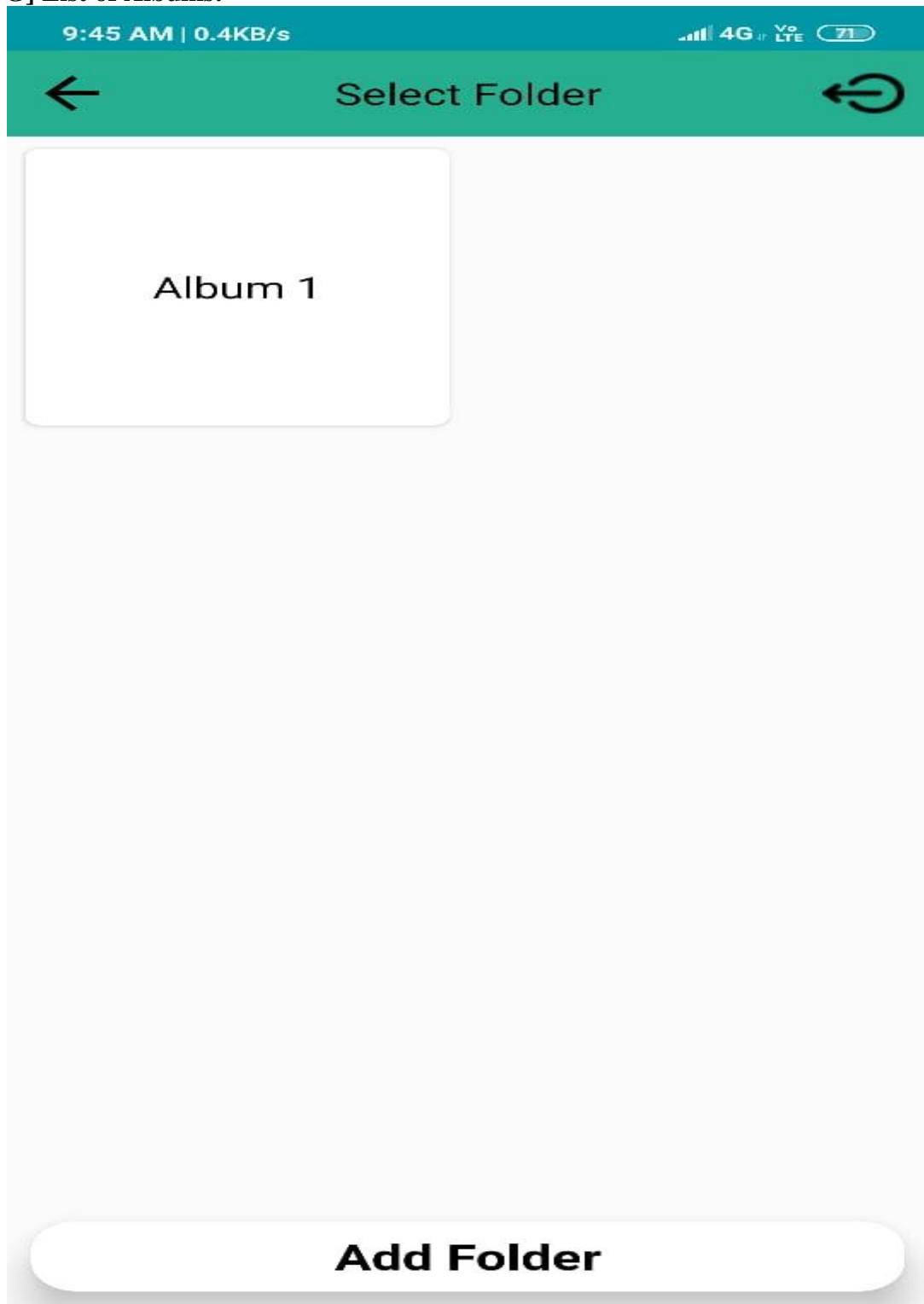


Fig 8.7 List of Albums

9.0 LIMITATION AND FUTURE ENHANCEMENT

9.1 Limitation

Biggest problem ever facing by Ondemand Music (Self-DJ) app is to stream Music Online over the internet, Also internet has been the major issue. The System is not able to send notifications to its end-users. Also, Search functionality is also not there to search albums, songs, etc.

9.2 Future Enhancement

This Ondemand Music (Self-DJ) system aimed at developing an Ondemand Music (Self-DJ) system which can be used in small places and medium cities firstly and then large scale. It is developed to introduce a new concept to hear songs in most subtle way. Most of the people quarrel among each other for playing their own favorite song. With the help of the music system they will be able to manage very easily and efficiently. Also, this Music System can work on Small scale environment. It can be changed from small scale to large scale. In large scale purpose, each and every user of this system can be able to use this system properly.

10.0 CONCLUSION AND DISCUSSION

10.1 Conclusion

Every project work, doesn't matter software project or any other project, could not be the result of sole effort. We think that the success of any project doesn't depend only on better software development skills but also, zeal to listen and help the users. We experienced that developing this software helped enhance our technical skills of programming.

10.2 Discussion

10.2.1 Self-Analysis of Project Viabilities

According to us, this project is absolutely a good start for gaining hands-on experience on projects. It is useful if it is managed according to the goal for which it is made.

10.2.2 Problems Encountered and Possible Solutions

There are so many problem encountered during this project.

1. Problem to maintain databases.
2. Need to change some functionality fully which lead to do the whole work again

10.2.3 Summary of Project Work

It is a great achievement to successfully complete the project. The prior knowledge of software engineering has helped immensely in overcoming the various roadblocks. We have done work with pre-planned scheduling related with time constraints and result oriented progress in project development.

11.0 REFERENCES

1. https://www.youtube.com/results?search_query=arsl+tech
2. www.stackoverflow.com
3. www.arsltechvideos.com
4. www.firebase.com
5. <https://codelabs.developers.google.com>
6. www.youtube.com/AndroidStudioFirebaseTutorials
7. www.firebaseTutorials.com
8. www.stackoverflow.com