ST. TERESA'S COLLEGE (AUTONOMOUS)

AFFILIATED TO MAHATMA GANDHI UNIVERSITY, KOTTAYAM

PITCH PERFECT

PROJECT REPORT

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN

COMPUTER APPLICATIONS

[TRIPLE MAIN]

Submitted By

Anna Teresa Abraham

III B.Sc. Computer Applications [Triple Main]

Register No: SB21CA007

Under the guidance of

Assistant Professor

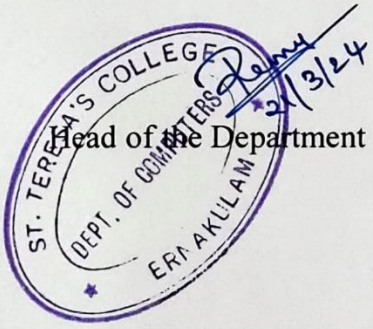
Ms. Elizabeth Paul

DEPARTMENT OF COMPUTER APPLICATIONS

2021-2024

CERTIFICATE

This is to certify that the project report entitled "**Pitch Perfect**" is a bona-fide record of the work done by **ANNA TERESA ABRAHAM (SB21CA007)** during the year 2021 – 2024 and submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Applications (Triple Main) under Mahatma Gandhi University, Kottayam.



Head of the Department

21/03/24

Internal Examiner

22.03.24

External Examiner

Date: 21/03/2024

DECLARATION

I, **ANNA TERESA ABRAHAM (Register no: SB21CA007),** B.Sc. Computer Applications [Triple Main] final year student of St. Teresa's College (Autonomous), Ernakulam, hereby declare that the project submitted named "**Pitch Perfect**" for the Bachelor's Degree in Computer Applications [Triple Main] is my original work. I further declare that the said work has not previously been submitted to any other university or academic body.

Date: 21/03/2024

Place: ERNAKULAM

ANNA TERESA ABRAHAM

ACKNOWLEDGEMENT

I would like to convey my heartfelt gratitude to **Rev. Dr. Sr. Vinitha (CSST) Manager, Director Rev. Sr. Emeline (CSST)** and **Principal Dr. Alphonsa Vijaya Joseph** for providing me with this wonderful opportunity to work on a project with the topic **Pitch Perfect**.

I would like to express my profound gratitude to the Head of the Department of Computer Applications **Ms. Remya C.J**, my project guide **Ms. Elizabeth Paul** and all other faculty of the department for their contributions to the completion of my project. The completion of the project would not have been possible without their help and insights.

I would also like to Thank my friend Ahsan Salim for training me well to develop this project.

Finally, I take this opportunity to Thank all them who has directly or indirectly helped me with my project.

ANNA TERESA ABRAHAM

SYNOPSIS

Pitch Perfect is a mobile application designed to empower users in expressing their musical creativity through vocal manipulation. Serving as a versatile tool for aspiring musicians and content creators, Pitch Perfect offers a user-friendly platform for recording and processing vocals with a focus on auto-tune functionality. The app provides an array of pre-set vocal effects, ranging from reverb to harmonization, enabling users to effortlessly enhance their recordings with a professional touch.

With real-time processing capabilities, users can receive instant feedback and guidance to improve their pitch accuracy, allowing them to sing with precision and confidence. Pitch Perfect facilitates a seamless music-making experience, allowing users to export their creations in various formats and share them on social media or other music-sharing platforms. Additionally, the app may offer in-app purchases for premium content in the future, expanding the range of creative possibilities for users seeking to elevate their vocal productions. Overall, Pitch Perfect stands as a user-centric application that democratizes vocal processing, making music production accessible to a broader audience.

CONTENTS

1.INTRODUCTION 1

1.1 About Project 2

1.2 Objectives of the Project 2

2. SYSTEM ANALYSIS 3

2.1 Introduction 4

2.2 Existing System 4

2.3 Proposed System 4

2.4 Justification of the proposed system 5

2.5 Benefits of the proposed system 6

2.6 System Specification 7

2.7 Operating System 7

2.8 Languages or Software Packages 8

2.9 Hardware and Software Specifications 9

3. SYSTEM DESIGN 10

3.1 Introduction 11

.3.2 Data Flow Diagram 11

3.3 E-R Diagram 12

3.4 Data Dictionary 14

3.5 Data Design 14

4. SYSTEM DEVELOPMENT 16

4.1 Introduction 17

4.2 Process Description 17

4.3 Code Design 18

5. SYSTEM TESTING AND IMPLEMENTATION 23

5.1 Introduction 24

5.2 System Implementation 24

5.3 Debugging 24

5.4 System Security 25

5.5 Scope for future enhancement 26

5.6 Conclusion 26

7. APPENDIX 27

8. BIBLIOGRAPHY 34

8.1 References 35

1.INTRODUCTION

1.1 About Project

Pitch Perfect is a specialized mobile application designed to assist singers in refining their pitch accuracy through real-time pitch correction. With its intuitive interface and precise feedback mechanism, Pitch Perfect provides singers with a dedicated platform to practice their vocal skills effectively. While its primary focus is on pitch correction, Pitch Perfect offers a user-friendly experience for singers of all levels, helping them to improve their vocal performance and enhance their overall musical abilities. Whether you're a beginner looking to develop your pitch accuracy or an experienced vocalist aiming for perfection, Pitch Perfect is here to support your journey towards musical excellence.

1.2 Objectives of the Project

Pitch Perfect aims to achieve two primary objectives: first, to provide singers with a dedicated platform for refining their pitch accuracy through real-time pitch correction; and second, to offer a user-friendly experience that caters to singers of all skill levels, from beginners to professionals. Through its intuitive interface and precise feedback mechanism, Pitch Perfect strives to empower singers to enhance their vocal performance and overall musical abilities. By focusing on these objectives, Pitch Perfect seeks to become the go-to tool for singers looking to improve their pitch accuracy and achieve their musical aspirations, thereby contributing to the growth and development of the global singing community.

2. SYSTEM ANALYSIS

2.1 Introduction

Pitch Perfect is a cutting-edge mobile application designed to revolutionize the way singers refine their vocal skills. With its primary focus on real-time pitch correction, Pitch Perfect provides users with an intuitive platform to enhance their pitch accuracy and overall singing performance. Whether you're a seasoned vocalist or just starting your musical journey, Pitch Perfect offers a user-friendly interface and precise feedback mechanism to support singers of all levels in achieving their musical goals. By combining advanced technology with user-centric design, Pitch Perfect aims to empower singers worldwide to unlock their full potential and elevate their vocal abilities to new heights.

2.2 Existing System

Existing vocal training platforms like Yousician, Vanido, Smule, Singing Machine, and Karaoke typically lack the real-time pitch correction feature that Pitch Perfect offers. Without this functionality, users may struggle to identify and correct pitch inaccuracies in their singing, hindering their progress in vocal improvement. Pitch Perfect's emphasis on pitch correction provides users with immediate feedback on their pitch accuracy, enabling them to address errors in real-time and make tangible improvements in their singing abilities. By offering this essential feature, Pitch Perfect distinguishes itself as a valuable tool for singers looking to refine their pitch and enhance their overall vocal performance.

2.3 Proposed System

Pitch Perfect offers a streamlined user experience focused on two core features: real-time pitch correction and diverse recording effects. Through its intuitive interface, users can access these functionalities seamlessly, whether they're beginners looking to refine their pitch accuracy or seasoned vocalists seeking to add creative flair to their recordings. The real-time pitch correction feature provides users with immediate feedback on their pitch accuracy while singing, allowing for precise adjustments and improvements.

This feature is complemented by a variety of recording effects, ranging from reverb and echo to distortion and modulation, enabling users to experiment with different vocal styles and enhance their recordings with professional-grade effects.

User: The user base of Pitch Perfect comprises individuals with varying levels of singing experience and diverse musical interests. Beginners may utilize the app to develop their pitch accuracy and gain confidence in their singing abilities, while experienced vocalists can leverage the recording effects to create unique and polished vocal performances. Additionally, music enthusiasts may explore Pitch Perfect's recording capabilities to express their creativity and share their musical talents with others. Whether users are aspiring singers, seasoned performers, or casual music enthusiasts, Pitch Perfect offers a versatile platform for vocal expression and artistic exploration. With its user-friendly interface and powerful features, Pitch Perfect empowers users to unleash their creativity and achieve professional-quality vocal recordings, making it an essential tool for singers and music lovers alike.

2.4 Justification of the proposed system

1. **Accessibility**: Pitch Perfect makes Indian classical music education accessible to a global audience. It removes geographical barriers by providing lessons and practice sessions through a mobile app, allowing anyone with a smartphone to learn and practice at their convenience.

2. **Structured Learning**: The app offers a structured curriculum designed by experienced music educators. Users can progress through various levels of difficulty, ensuring a systematic and comprehensive learning experience.

3. **Personalized Feedback**: Pitch Perfect employs advanced technologies such as machine learning and artificial intelligence to provide personalized feedback to users. This feature helps learners understand their strengths and weaknesses, enabling them to focus on areas that need improvement.

4. **Practice Tools**: The app offers a range of practice tools, including vocal exercises, rhythm exercises, and song tutorials. These tools help users develop essential skills such as pitch accuracy, rhythm sense, and tonal quality.

5. **Gamification**: Pitch Perfect incorporates gamification elements to make the learning process engaging and enjoyable. Users earn rewards and achievements as they progress through the lessons, motivating them to continue practicing and improving.
6. **Community Engagement**: The app fosters a sense of community among users by providing forums, discussion boards, and live sessions with expert instructors. This sense of community encourages collaboration, peer learning, and support among learners.
7. **Cost-Effective**: Traditional music education can be expensive, requiring private lessons or enrollment in music schools. Pitch Perfect offers an affordable alternative, providing high-quality instruction at a fraction of the cost.
8. **Continuous Improvement**: The developers of Pitch Perfect regularly update the app with new features, lessons, and improvements based on user feedback and technological advancements. This commitment to continuous improvement ensures that users have access to the latest resources and tools for their musical journey.

2.5 Benefits of the proposed system

1. **Convenience**: Pitch Perfect allows users to learn and practice Indian classical music at their own pace and schedule, eliminating the need for fixed lesson times or locations. This flexibility enables users to fit music practice into their busy lives.
2. **Structured Learning**: The app offers a structured curriculum designed by experienced music educators, guiding users through a progressive series of lessons that cover fundamental concepts to advanced techniques. This structured approach ensures a comprehensive and effective learning experience.
3. **Personalized Feedback**: Pitch Perfect provides users with personalized feedback on their performance, helping them identify areas for improvement and track their progress over time. This individualized guidance enhances the learning process and facilitates skill development.

4. **Variety of Practice Tools**: The app offers a variety of practice tools, including vocal exercises, rhythm exercises, and song tutorials. These tools cater to different learning styles and help users develop essential skills such as pitch accuracy, rhythm proficiency, and tonal quality.
5. **Community Support**: Pitch Perfect fosters a supportive community of learners and instructors, where users can connect, share experiences, and seek advice. This sense of community encourages collaboration, motivation, and mutual learning among users.

6. **Accessibility**: As a mobile app, Pitch Perfect provides easy access to music education for users around the world, regardless of their geographical location or access to traditional music schools. This accessibility democratizes music learning and promotes inclusivity.
7. **Affordability**: Compared to traditional music lessons or workshops, Pitch Perfect offers an affordable alternative, making high-quality music education accessible to a broader audience. This affordability removes financial barriers to learning and allows users to pursue their passion for music without breaking the bank.
8. **Continuous Improvement**: The developers of Pitch Perfect are committed to regularly updating the app with new features, content, and improvements based on user feedback and technological advancements. This commitment to continuous improvement ensures that users have access to the latest resources and tools to support their musical journey.

2.6 System specifications

A system specification is a formal document that outlines the functional and non-functional requirements for a proposed system. It provides a comprehensive description of the system's behaviour, features, and limitations. A system specification is used as a roadmap for the design, development, and implementation of the system.

Pitch Perfect, an innovative mobile app, operates efficiently across Android platforms, utilizing modern development frameworks for optimal performance. With its intuitive interface, users can seamlessly navigate through various features, including real-time pitch correction and vocal effects. The app supports external audio interfaces and Bluetooth devices, providing flexibility in audio setup. Security measures are implemented to protect user data and privacy.

2.7 Operating System

An operating system (OS) is a software program that manages computer hardware resources and provides common services for computer programs. It acts as an intermediary between computer hardware and software applications, providing an interface for software developers to interact with the hardware. It simply provides an environment within which other programs can do useful work. Operating system required for Pitch Perfect is Android 5.0 or later.

2.8 Languages or Software Packages

- **Visual Studio:**
  Visual Studio Code is a lightweight yet powerful source code editor developed by Microsoft. It is widely used by developers for building and debugging various types of applications across different platforms, including web development, cloud services, and mobile apps. Visual Studio Code offers a rich set of features such as syntax highlighting, IntelliSense for code completion, debugging capabilities, Git integration, and support for extensions. Its customizable and user-friendly interface, along with its cross-platform compatibility, makes it a popular choice among developers seeking a fast and efficient code editing experience.

- **Java:** Java is a general-purpose computer-programming language that is concurrent, class- based, object-oriented, and specifically designed to have as few implementation dependencies as possible. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. The language derives much of its original features from Smalltalk, with a syntax like C and C++, but it has fewer low-level facilities than either of them. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.

- **MySQL**: MySQL is an open-source relational database management system (RDBMS) that is widely used for managing and storing structured data. It provides a robust, scalable, and reliable platform for building database-driven applications. MySQL supports SQL (Structured Query Language) for querying and manipulating data, making it easy to interact with databases. It offers features such as transactions, indexing, replication, and security mechanisms to ensure data integrity and performance. MySQL is highly customizable and can be used in various environments, from small-scale applications to large-scale enterprise systems. Overall, MySQL is a popular choice for developers seeking a powerful and flexible database solution for their applications.

- **JavaScript**: JavaScript, the language behind the buttons you click and the forms you submit, breathes life into static web pages. It's like the hidden conductor orchestrating the dynamic elements we take for granted. This lightweight, beginner-friendly language empowers you to create interactive experiences, from simple calculations to dazzling animations. Whether you're building a basic website or a complex application, JavaScript has the tools you need.

2.9 Hardware and Software Specifications

Hardware Requirements:

Android:
Operating system: Android 5.0 or later
RAM: 2 GB or more
Storage: 500 MB or more
Processor: 1.2 GHz or faster dual-core processor

Software Requirements:

Internet connection: Pitch Perfect likely requires a stable internet connection for downloading app content, streaming lessons, and interacting with other users.
Microphone access: If you want to record your voice for singing practice or participating in challenges, you'll need to grant the app microphone access.
Camera access: Some features may require camera access, such as recording video lessons or using augmented reality features.
Additional apps: Certain features might rely on other apps being installed on your device, like a music player or social media app.

3. SYSTEM DESIGN

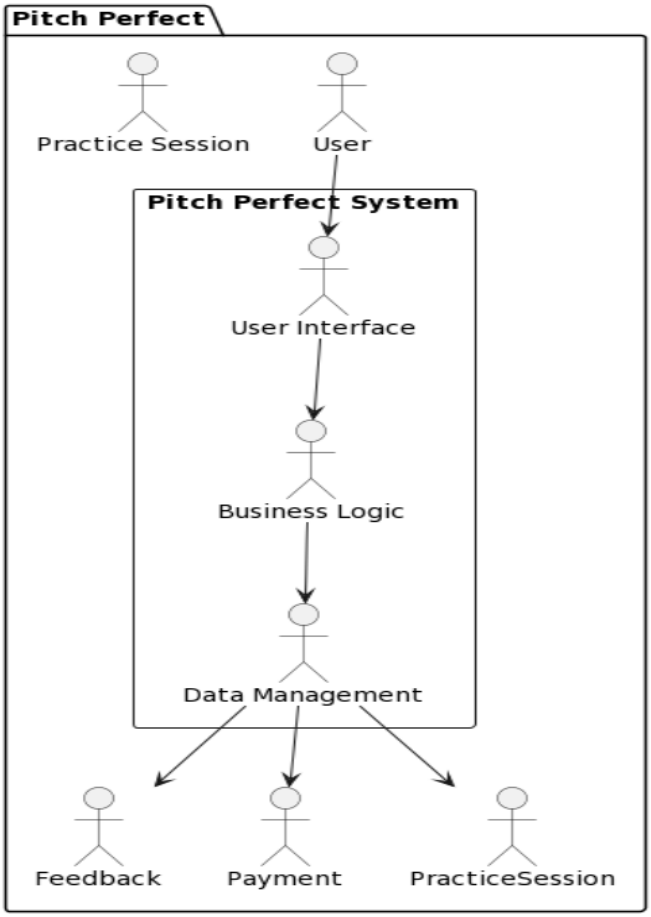
3.1 Introduction

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It is a critical step in the software development lifecycle that bridges the gap between the requirements analysis and implementation phases.

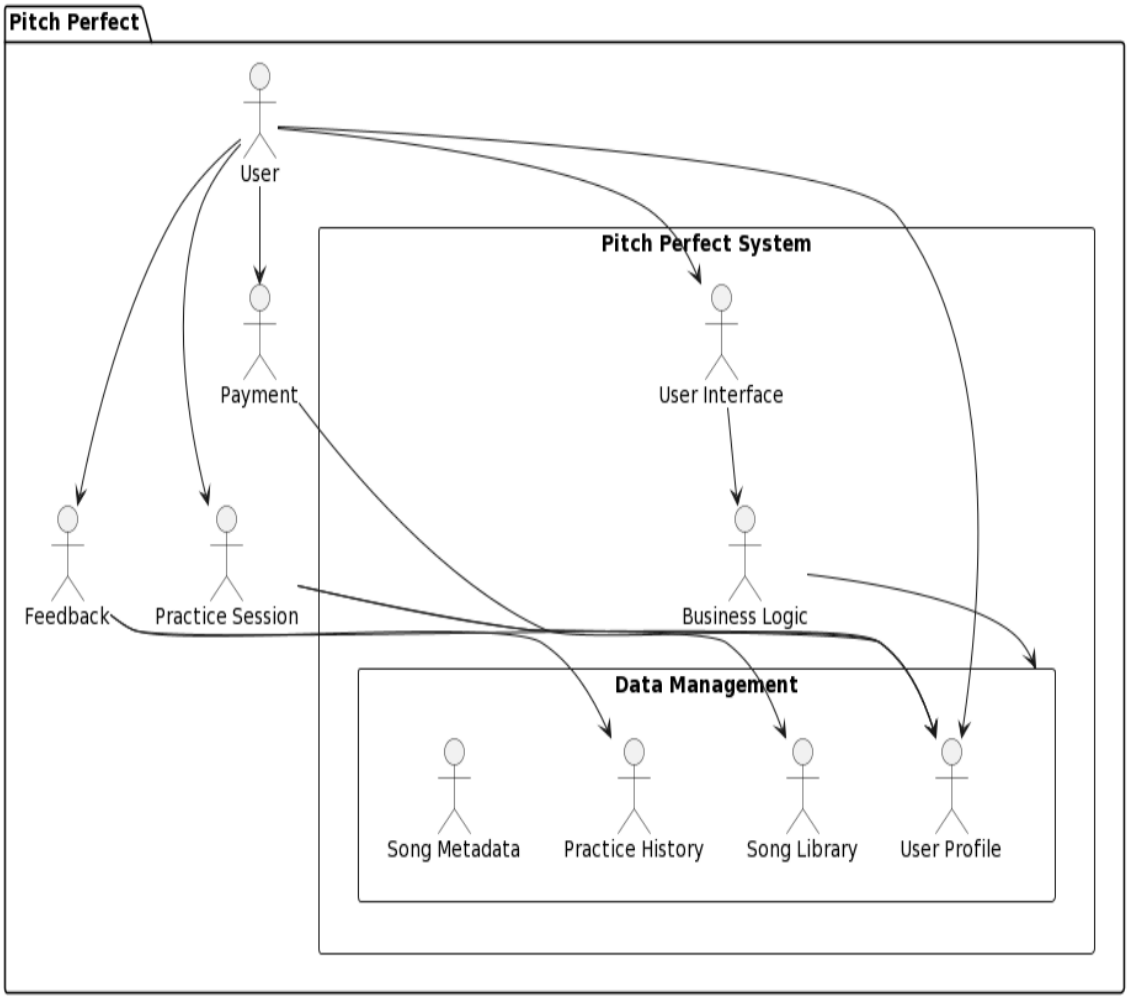
3.2 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the flow of data through an information system. A DFD is often used as a primary step to create an overview of the system, which can later be elaborated. A DFD shows what will be the input of the system as well as the output. It clearly represents where the data will come from and go to, and where the data will be stored.

DFD Level 0



DFD Level 1



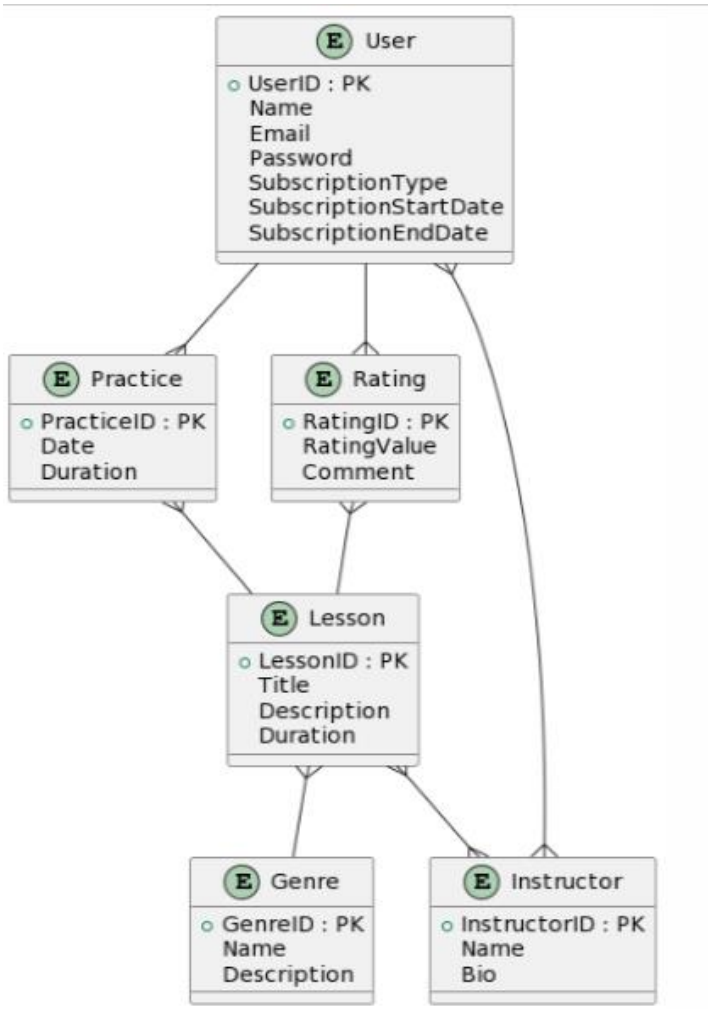
3.3 E-R Diagram

An E-R (Entity-Relationship) diagram is a visual representation used in database design to illustrate the relationships between different entities in a system. It depicts the logical structure of a database by showing the entities, their attributes, and the relationships between them.

In an E-R diagram:

1. Entities: Entities represent real-world objects or concepts that can be distinguished from one another. Each entity is depicted as a rectangle in the diagram and typically corresponds to a table in the database.

2. Attributes: Attributes are the properties or characteristics of entities. They are depicted as ovals connected to the corresponding entity rectangle. Attributes describe the features of an entity and help define its properties.
3. Relationships: Relationships illustrate the associations or connections between entities. They describe how entities interact with each other in the database. Relationships are represented by lines connecting the relevant entities, with labels indicating the nature of the relationship (such as "belongs to," "has," or "is related to")
.



3.4 Data Dictionary

A data dictionary is a collection of metadata that provides information about the data used in a database or information system. It serves as a reference guide for data elements and their definitions, as well as their relationships with other data elements in the system. A data dictionary is useful for ensuring data accuracy, consistency, and completeness. It can also help facilitate communication and collaboration among stakeholders who are involved in designing, developing, and maintaining a database or information system.

3.5 Data Design

Database design is the process of creating a well-organized and structured database that efficiently stores and retrieves data. A well-designed database ensures data accuracy, consistency, and completeness and provides an intuitive and user-friendly interface for interacting with the data. Data design is the first design activity, which results in less complex, modular, and efficient program structure. The information domain model developed during analysis phase is transformed into data structures needed for implementing the software. The data objects, attributes, and relationships depicted in entity relationship diagrams and the information stored in data dictionary provide a base for data design activity. During the data design process, datatypes are specified along with the integrity rules required for the data. For specifying and designing efficient data structures, some principles should be followed.

login

Field	Data Type	Description
User ID	String	Unique identifier for the user
Username	String	Unique username chosen by the user
Email	String	Email address provided by the user
Password	String	Securely stored password
Full Name	String	User's full name
Date of Birth	Date	User's date of birth
Gender	String	User's gender
Phone Number	String	Phone number provided by the user
Country	String	User's country of residence
Account Type	String	Indicates whether the account is premium/free
Profile Image	Image	's profile picture
Registration Date	DateTime	Timestamp of user registration

user reg

Field	Data Type	Description
Result ID	String	Unique identifier for the result
User ID	String	Identifier for the user associated with the result
Practice Session ID	String	Identifier for the practice session associated with the result
Song ID	String	Identifier for the song associated with the result
Date	DateTime	Timestamp of the result
Duration	Duration	Duration of the practice session
Accuracy	Float	Accuracy score achieved by the user during practice
Progress	Float	Progress made by the user during practice
Feedback	String	Feedback provided by the system or instructor

result

Field	Data Type	Description
Username	String	Unique identifier for the user
Password	String	Securely stored password
Email	String	Email address associated with the user
Phone Number	String	Phone number associated with the user
Last Login	DateTime	Timestamp of the user's last login
Account Type	String	Indicates whether the account is premium/free
Profile Image	Image	User's profile picture

4. SYSTEM DEVELOPMENT

4.1 Introduction

Software Development is the process of analysing, designing, testing, implementation,

and maintenance. It is called Software Development Life Cycle (SDLC). Different SDLC include waterfall, prototyping, iterative, incremental, spiral development, rapid application development and agile methodology.

4.2 Process Description

Different processes of each module are given below:

User Registration: During registration, users may be required to provide basic information such as their name, email address, and preferred username.

Skill Level Assessment: After registration, users are prompted to complete a skill level assessment to determine their current proficiency in singing. This assessment may involve singing exercises or answering questions about their vocal range, tone, and musical background.

Personalized Practice Plan: Based on the results of the skill level assessment, Pitch Perfect generates a personalized practice plan for each user. This plan outlines specific vocal exercises and practice routines tailored to the user's skill level and areas for improvement.

Vocal Exercises: Pitch Perfect offers a variety of vocal exercises targeting different aspects of singing technique, such as breath control, pitch accuracy, vocal range, and articulation. Users can access these exercises through the app and practice them at their own pace.

Practice Sessions: Users can engage in structured practice sessions within the Pitch Perfect app, following the guidance provided in their personalized practice plan. During practice sessions, users receive real-time feedback on their performance, including pitch accuracy and tone quality.

Feedback and Progress Tracking: Pitch Perfect provides users with feedback on their singing performance, highlighting areas for improvement and offering suggestions for enhancement. Users can track their progress over time and monitor their improvement through performance metrics and statistics.

Community Engagement: Pitch Perfect fosters a sense of community among users by providing features for interaction and collaboration. Users can participate in group challenges, share their progress with others, and receive support and encouragement from fellow singers.

Additional Features: In addition to the core features mentioned above, Pitch Perfect may offer additional functionalities such as vocal warm-up routines, song tutorials, and integration with social media platforms for sharing singing performances.

4.3 Code Design

```
package coil.compose;

import coil.size.ScaleResolver; import coil.size.SizeResolver; import
android.content.Context; import coil.size.Dimension;
import coil.size.Dimension$Original; import coil.size.Dimension$Pixels;
import androidx.compose.ui.semantics.SemanticsModifierKt; import
androidx.compose.ui.unit.Constraints;
import coil.size; import coil.size.Scale;
import androidx.compose.ui.node.ComposeUiNode$Companion; import
androidx.compose.runtime.Updater;
import kotlin.jvm.functions.Function0;
import androidx.compose.runtime.ComposablesKt; import
androidx.compose.runtime.Applier;
import androidx.compose.ui.node.ComposeUiNode; import
androidx.compose.ui.ComposedModifierKt;  import
androidx.compose.ui.platform.ViewConfiguration; import
androidx.compose.ui.unit.LayoutDirection;
import androidx.compose.runtime.CompositionLocal;  import
androidx.compose.ui.platform.CompositionLocalsKt; import
androidx.compose.ui.unit.Density;
import androidx.compose.ui.draw.ClipKt;
```

```
import androidx.compose.runtime.ScopeUpdateScope; import
coil.request.ImageRequest

import kotlin.jvm.functions.Function2;
import androidx.compose.ui.graphics.painter.Painter;
import androidx.compose.ui.graphics.drawscope.DrawScope; import
androidx.compose.runtime.Composer;
import androidx.compose.ui.graphics.ColorFilter; import
androidx.compose.ui.layout.ContentScale; import
androidx.compose.ui.Alignment;
import kotlin.Unit;
import kotlin.jvm.functions.Function1; import androidx.compose.ui.Modifier;
import coil.ImageLoader;

public final class AsyncImageKt {
public static final void a(final Object o, final String s, final ImageLoader
imageLoader, Modifier r, Function1<? super AsyncImagePainter.State, ? extends
AsyncImagePainter.State> a, Function1<? super AsyncImagePainter.State, Unit>
function1, Alignment d, ContentScale b, float n, ColorFilter colorFilter, int b2,
final Composer composer, final int n2, final int n3, final int n4) {
        final Composer h = composer.h(-1423043099); if ((n4 & 0x8) != 0x0) {
r = (Modifier)Modifier.R; }
if ((n4 & 0x10) != 0x0) {
a = AsyncImagePainter.v.a(); }
if ((n4 & 0x20) != 0x0) { function1 = null;
}
if ((n4 & 0x40) != 0x0) { d = Alignment.a.d();
}
if ((n4 & 0x80) != 0x0) { b = ContentScale.a.b();
}
if ((n4 & 0x100) != 0x0) { n = 1.0f;
}
if ((n4 & 0x200) != 0x0) { colorFilter = null;
}
int n5;
if ((n4 & 0x400) != 0x0) { b2 = DrawScope.T.b();
n5 = (n3 & 0xFFFFFFF1); }


else {
n5 = n3; }
final ImageRequest h2 = h(UtilsKt.e(o, h, 8), b, h, 0x8 | (n2 >> 18 & 0x70)); final
int n6 = n2 >> 6;
final int n7 = n2 >> 9;
final int n8 = 0xE000 & n7;
final AsyncImagePainter d2 = AsyncImagePainterKt.d(h2, imageLoader, a,
function1, b , b2, h, (n5 << 15 & 0x70000) | ((n6 & 0x1C00) | ((n6 & 0x380) |
0x48) | n8), 0);
```

```
final ConstraintsResolver c = UtilsKt.c(h2); Modifier t0;
if (c != null) {
t0 = r.t0((Modifier)c); }
else {
t0 = r; }
b(t0, (Painter)d2, s, d, b, n, colorFilter, h, (n7 & 0x1C00) | (n2 << 3 & 0x380) | n8
| (n7 & 0x70000) | (0x380000 & n7));
final ScopeUpdateScope k = h.k(); if (k != null) {
k.a((Function2)new AsyncImageKt$AsyncImage$2(o, s, imageLoader, r,
(Function1) a, (Function1)function1, d, b, n, colorFilter, b2, n2, n3, n4));
} }

public static final void b(final Modifier modifier, final Painter painter, final String
s, final Alignment alignment, final ContentScale contentScale, final float n, final
ColorFilter colorFilter, final Composer composer, final int n2) {
final Composer h = composer.h(-341424986);
final Modifier t0 = ClipKt.b(f(modifier, s)).t0((Modifier)new
ContentPainterModifier (painter, alignment, contentScale, n, colorFilter));
final AsyncImageKt$Content$1 a = AsyncImageKt$Content$1.a; h.x(1376091099);
final Density density = (Density)h.n((CompositionLocal)CompositionLocalsKt.d());
final LayoutDirection layoutDirection = (LayoutDirection)h.n((CompositionLocal)
CompositionLocalsKt.i());
final ViewConfiguration viewConfiguration = (ViewConfiguration)h.n
((CompositionLocal)
final Modifier c = ComposedModifierKt.c(h, t0);
final ComposeUiNode$Companion u = ComposeUiNode.U; final Function0 a2 =
u.a();
h.x(1546164872);
if (!(h.j() instanceof Applier)) { ComposablesKt.c();
} h.D();
if (h.f()) {
h.G((Function0)new AsyncImageKt$Content$$inlined$Layout$1(a2)); }

else {
h.p(); }
h.E();
final Composer a3 = Updater.a(h); Updater.c(a3, (Object)a, u.d()); Updater.c(a3,
(Object)density, u.b());
Updater.c(a3, (Object)layoutDirection, u.c()); Updater.c(a3,
(Object)viewConfiguration, u.f()); Updater.c(a3, (Object)c, u.e());
h.c(); h.r(); h.O(); h.O();
final ScopeUpdateScope k = h.k(); if (k != null) {
k.a((Function2)new AsyncImageKt$Content$2(modifier, painter, s, alignment,
contentScale, n, colorFilter, n2));
} }

private static final Scale e(final long n, final Scale scale) { if (Constraints.j(n) &&
Constraints.i(n)) {
```

```
return scale; }
return Scale.FIT; }
private static final Modifier f(final Modifier modifier, final String s) { Modifier c =
modifier;
if (s != null) {
c = SemanticsModifierKt.c(modifier, false, (Function1)new
AsyncImageKt$contentDescription$1(s), 1, (Object)null);
}
return c; }

private static final Size g(final long n) { Size size;
if (Constraints.r(n)) { size = null;
}
else { Object a;
if (Constraints.j(n)) {
a = new Dimension$Pixels(Constraints.n(n));

else {
a = Dimension$Original.a; }
Object a2;
if (Constraints.i(n)) {
a2 = new Dimension$Pixels(Constraints.m(n)); }
else {
a2 = Dimension$Original.a; }
size = new Size((Dimension)a, (Dimension)a2); }
return size; }

public static final ImageRequest h(ImageRequest a, final ContentScale
contentScale, final Composer composer, final int n) {
composer.x(-1553384547);
final ImageRequest.Builder r = ImageRequest.R(a, null, 1, null); if (a.q().m() ==
null) {
composer.x(-3687241); Object y;
if ((y = composer.y()) == Composer.a.a()) { y = new ConstraintsResolver();
composer.q(y);
} composer.O();
final ConstraintsResolver constraintsResolver = (ConstraintsResolver)y ;
r.k((SizeResolver)constraintsResolver); if (a.q().l() == null) {
constraintsResolver.h(UtilsKt.g(contentScale));
r.j((ScaleResolver)constraintsResolver);
} }
a = r.a(); composer.O(); return a;
} }
```

5. SYSTEM TESTING AND IMPLEMENTATION

5.1 Introduction

Software testing is a crucial process in the software development life cycle that involves evaluating the quality and functionality of software applications.

The main goal of software testing is to identify defects or errors in the application before it is released to the end-users. The software testing process typically involves several steps, including test planning, test design, test execution, and test reporting.

5.2 System Implementation

Implementation is the action that must follow any preliminary thinking for something

to happen. Software/hardware implementations should always be designed with the end user in mind and the implementation process usually benefits from user involvement and support from managers and other top executives in the company. If users participate in the design and implementation of the system, ideally it will serve their business objectives more accurately and reflect their priorities and the ways in which they prefer to work.

5.3 Debugging

Debugging is the process of identifying and resolving issues or errors in software applications. It is a critical step in the software development life cycle and is aimed at improving the quality and functionality of the application. During the development process, errors or bugs can occur in the code that can cause the application to behave in unexpected ways or not work at all. Debugging involves identifying and diagnosing these errors, tracing their root cause, and then making the necessary changes to fix them.

Different types of debugging methods used in this system are:

• **Unit Testing:** The application was divided into smaller components and tested individually. Each code was executed separately to ensure accuracy.

• **Integration Testing:** Each small component was integrated or combined into a module to ensure that each module works properly when put together. This was done to check connectivity between modules.

• **System Testing:** The system was tested by combining every module. This was to ensure that each process have a particular order. This was to ensure that

the system does not crash while using.

5.4 System Security

Password encryption is used to protect each user’s details.

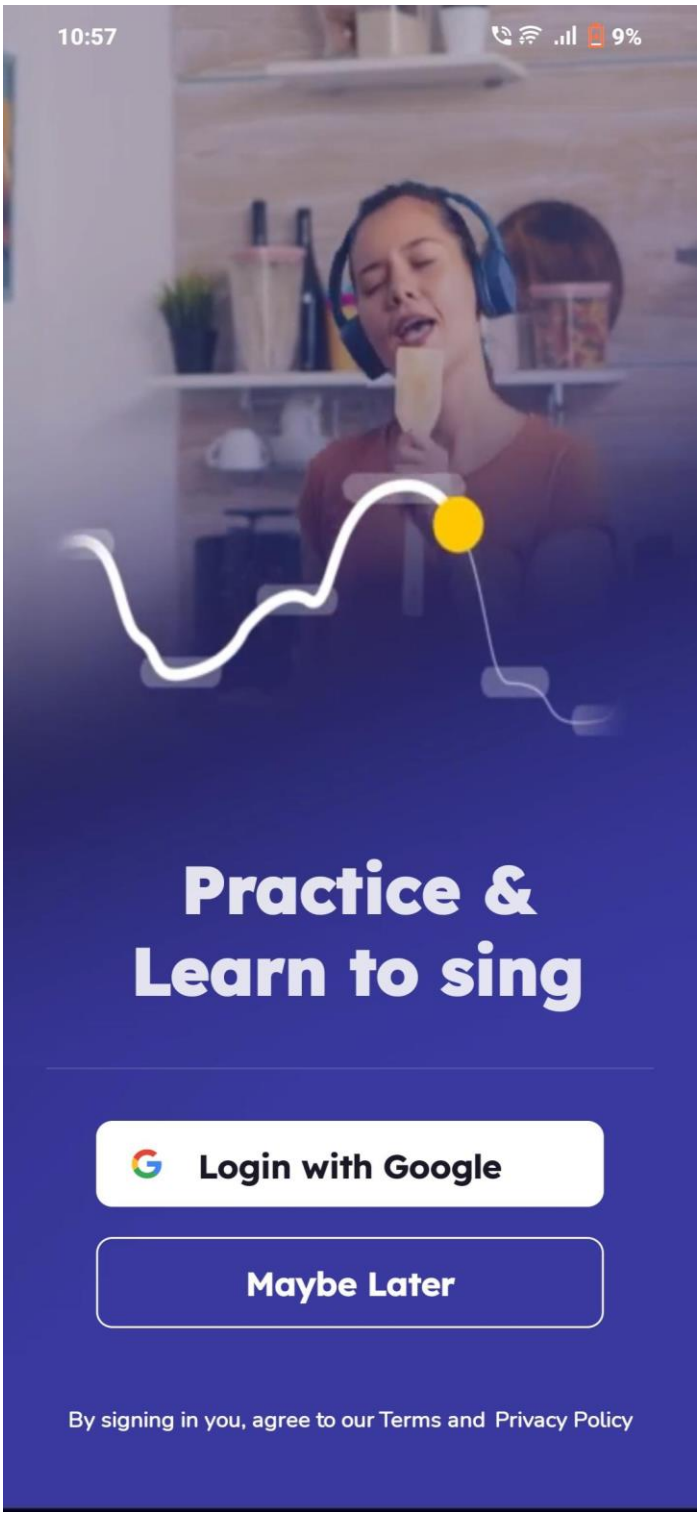
5.5 Scope for future enhancement

In the future, Pitch Perfect aims to expand its vocal training exercises, enrich song libraries, and implement advanced feedback mechanisms. Additionally, enhancing personalization through AI, fostering community interaction, and ensuring multi-platform compatibility are key objectives. These enhancements will provide users with a more comprehensive learning experience, encourage engagement within the community, and cater to diverse skill levels and preferences, positioning Pitch Perfect as a leading app for vocal practice and improvement.

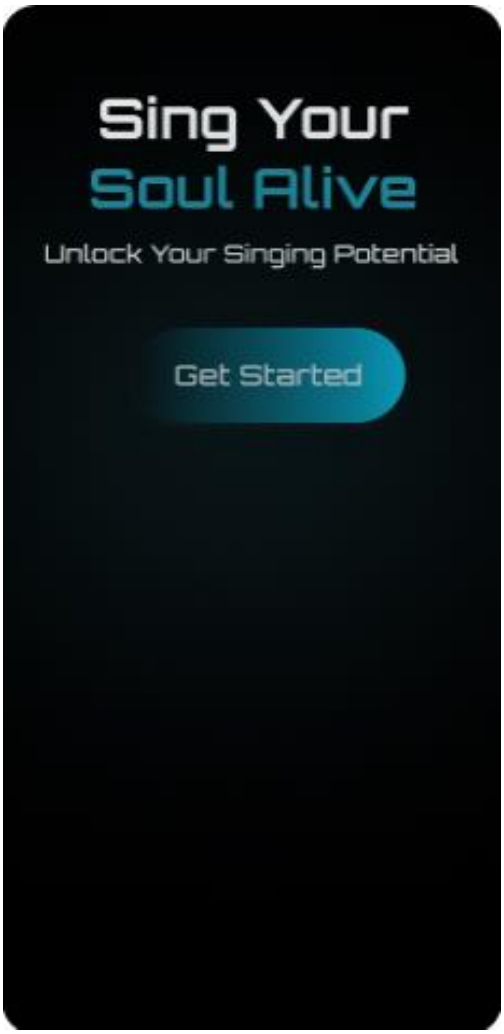
5.6 Conclusion

Pitch Perfect is a comprehensive app designed to revolutionize music learning. With its user-friendly interface and extensive library of songs across various genres and languages, Pitch Perfect offers a dynamic platform for aspiring singers to hone their skills. Its innovative features, including real-time feedback and personalized practice routines, empower users to improve their vocal abilities at their own pace. Moreover, Pitch Perfect's integration of technology, such as AI-driven assessments and interactive tutorials, enhances the learning experience, making it accessible to enthusiasts of all levels. Overall, Pitch Perfect stands out as a pioneering tool that combines traditional music education with modern technology, making it an indispensable companion for anyone passionate about refining their singing talents.

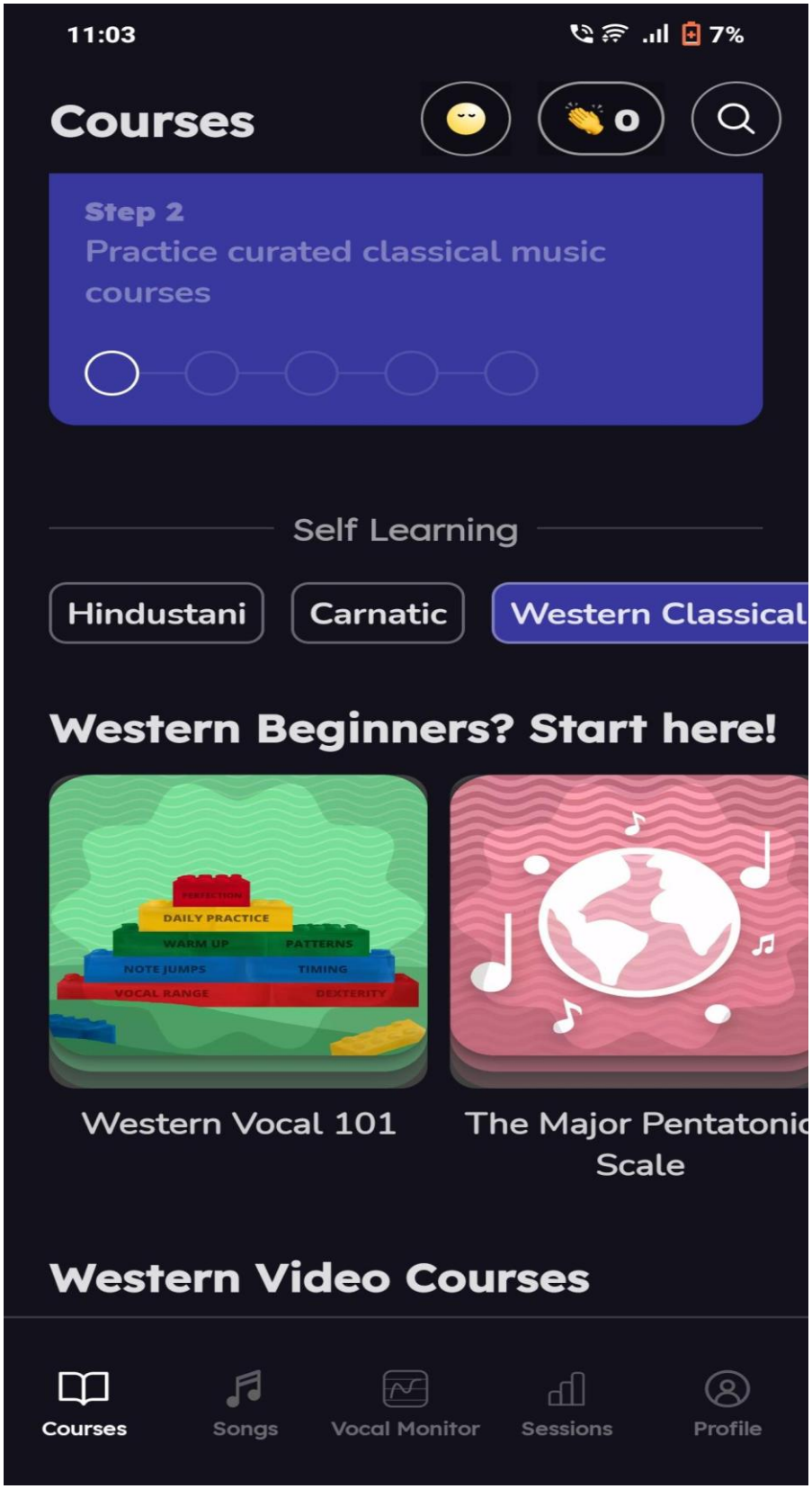
Login page


10:57
9%
Practice &
Learn to sing
Login with Google
Maybe Later
By signing in you, agree to our Terms and Privacy Policy

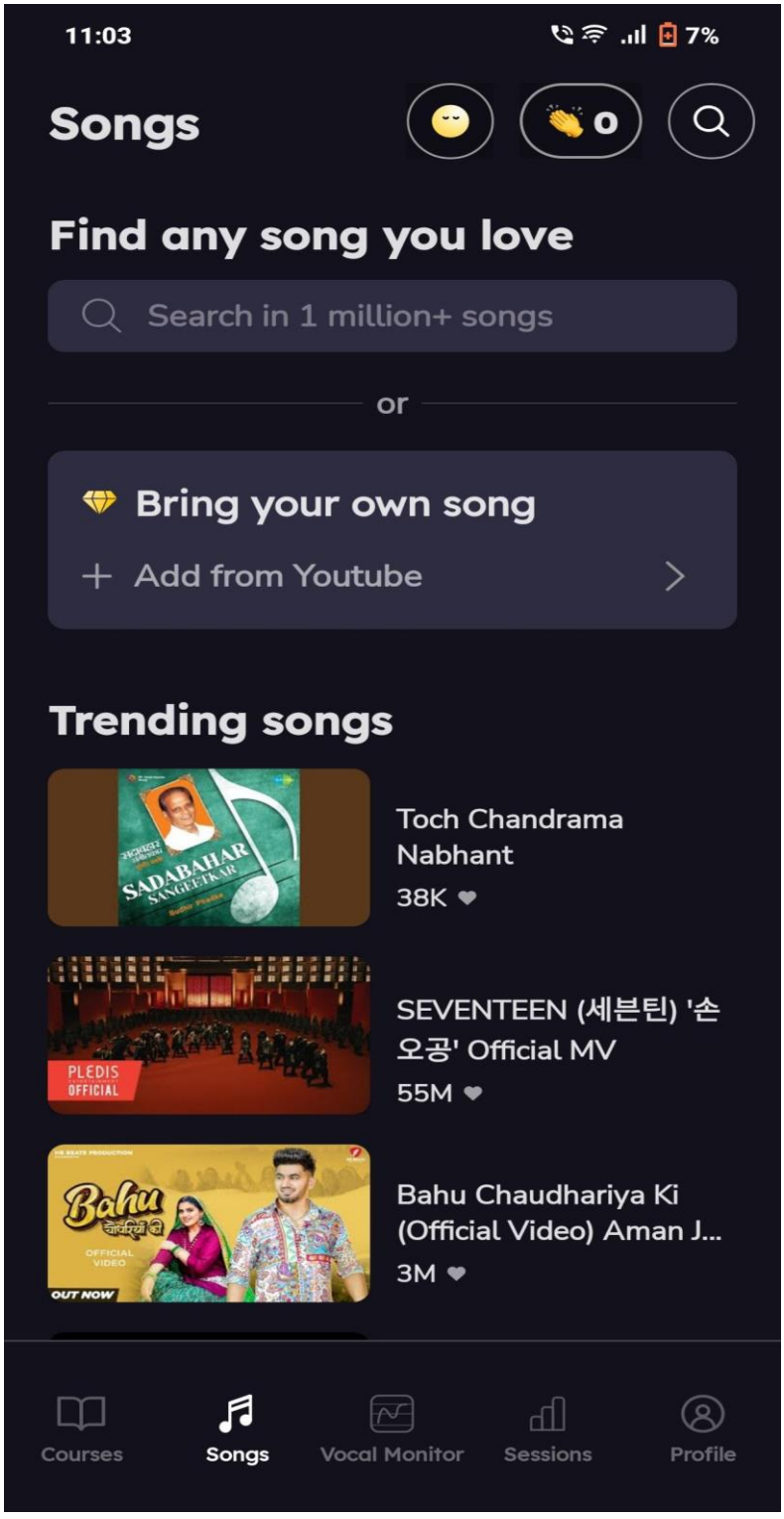
Homepage


Sing Your
Soul Alive
Unlock Your Singing Potential
Get Started

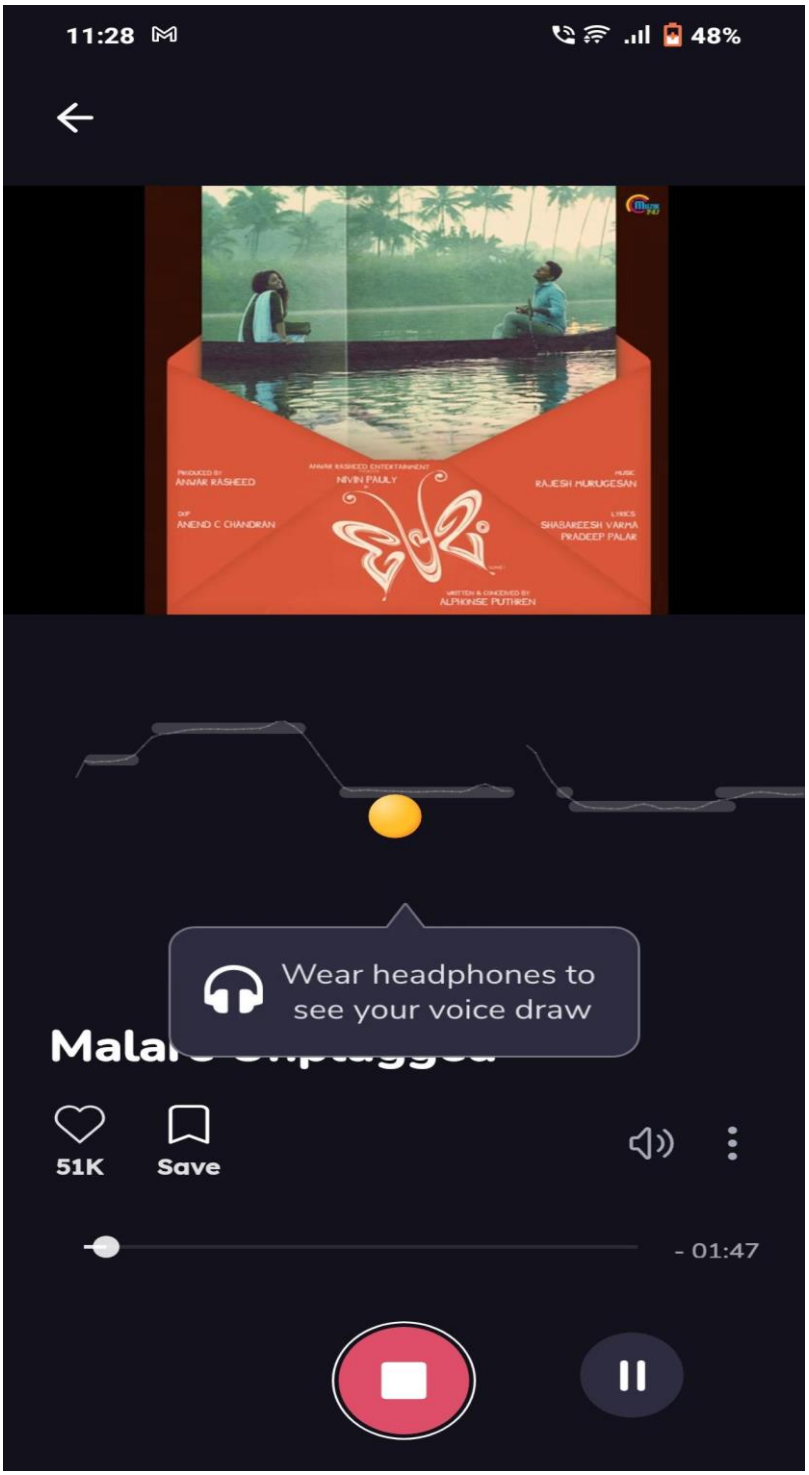
Courses


11:03
7%
Courses
0
Step 2
Practice curated classical music courses
Self Learning
Hindustani
Carnatic
Western Classical
Western Beginners? Start here!
DAILY PRACTICE
WARM UP
PATTERNS
NOTE JUMPS
TIMING
VOCAL RANGE
DEXTERITY
Western Vocal 101
The Major Pentatonic Scale
Western Video Courses
Courses
Songs
Vocal Monitor
Sessions
Profile

Songs


11:03
7%
Songs
0
Find any song you love
Search in 1 million+ songs
or
Bring your own song
Add from Youtube
Trending songs
Toch Chandrama Nabhant
38K
SEVENTEEN (세븐틴) '손 오공' Official MV
55M
Bahu Chaudhariya Ki (Official Video) Aman J...
3M
Courses
Songs
Vocal Monitor
Sessions
Profile

Sing Along



Profile


11:03
7%
Aakino Furaito
Profile
Library
Voice Résumé
More you sing, more accurate this gets.
Achievements
0
Lifetime claps
0
Joy Days
Tools
Vocal Range Detector
2 Minutes Warmup
Breath Monitor
How can we help?
Courses
Songs
Vocal Monitor
Sessions
Profile

8. BIBLIOGRAPHY

8.1 References

Websites referred:

- https://www.purrweb.com/blog/
- https://yellow.systems/blog/
- https://www.figma.com/