A

Major Project On

COLOUR DEPICTION FOR VARIOUS SOUNDS

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

P. Sharanya(187R1A0541)

R. Keerthana Yadav (187R1A0536)

A. Keerth Koumudh(187R1A0506)

Under the Guidance of

Dr. G. SOMA SEKHAR

(Associate Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE NewDelhi)
Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,
Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**COLOUR DEPICTION FOR VAROIOS SOUNDS**" being submitted by **P.SHARANYA (187R1A0541), R.KEERTHANA YADAV (187R1A0536) & A.KEERTH KOUMUDH (187R1A0506)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2018-22.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma

G. Soma Sekhar
Associate Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HOD

EXTERNAL EXAMINER

Submitted for viva voice Examination held on_____

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

Dr. G. Soma Sekhar, Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC)

Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Dr. Suwarna Gothane, Mr. A. Uday Kiran, Mr. A. Kiran Kumar, Mrs. G. Latha for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

P. Sharanya (187R1A0541)

R. Keerthana Yadav (187R1A0536)

A. Keerth Koumudh (187R1A0506)

ABSTRACT

Human perception of surrounding events is strongly dependent on audio cues. Thus, acoustic insulation can seriously impact situational awareness. We present an exploratory study in the domain of assistive computing, eliciting requirements and presenting solutions to problems found in the development of an environmental sound recognition system, which aims to assist deaf and hard of hearing people in the perception of sounds. To take advantage of smartphones computational ubiquity, we propose a system that executes all processing on the device itself, from audio features extraction to recognition and visual presentation of results. Our application also presents the confidence level of the classification to the user. A test of the system conducted with deaf users provided important and inspiring feedback from participants.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

Consciousness of what happens in the surrounding environment is strongly dependent on an individual's capacity to perceive sounds and accurately identify events related to them. Acoustic insulation can seriously compromise the ability of a person to acquire situational awareness, which is important for the execution of daily tasks, social interaction, and even personal safety. Hearing individuals may not realize how much they depend on auditory ability to perceive what is happening around them. Environmental sound awareness is necessary in a much higher number of situations than commonly imagined. Examples of this problem are provided by Matthews et al, who propose an environmental sound recognition (ESR) system for deaf users. Among other cases presented in their study, a participant reported that once had forgotten the vacuum cleaner on all night, since the device did not provide any visual cues that it was in operation. Based on previous research, the present document describes the design, development, and test of a mobile ESR system that aims to expand deaf individuals' situational awareness. We present an exploratory study in the field of assistive computing, describing solutions for problems encountered during the development of an ESR system specifically designed for deaf users. Currently there are still few studies on this topic, despite its importance. To foster future works, we formalize the requirements that guided the development of our system and provide details about the implementation of our solution.

In general terms, existing studies on sound recognition are divided into three categories: speech, music, and environmental sound. In the latter case, the most common approach is the use of predefined environmental sound classes, which can then be applied to the indexing/retrieval of audio/video documents and in surveillance systems, for instance. In our study, due to the diverse and ever-changing nature of sounds that the system is supposed to cover, we had to address the open-set problem regarding ESR as well as the representation of uncertainty. During recognition tasks, with the system being executed in an environment where there is no control over the occurrence of sounds, results can be quite inconsistent. One alternative to minimize this problem is to provide the user with information on the confidence level for the classification results.

1.1 OBJECTVE:

This application helps the deaf people to understand what is happening in the surrounding. This application will detect noise from surrounding and display message of detected noise. So that user can know, what is happening in the surrounding. We get signals from our surrounding and our brain process those signals. We hear those signals via our ear. Keeping these important words in mind we present this project to mainly focus on aiding the speech impaired and paralyzed patients.

2. LITERATURE SURVEY

LITERATURE SURVEY

What is an Android Application -?

Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear).

The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touchscreen input, it also has been used in game consoles, digital cameras, and other electronics. Android is the most popular mobile OS. As of 2013, Android devices sell more than Windows, iOS, and Mac OS devices combined, with sales in 2012, 2013 and 2014 close to the installed base of all PCs. As of July 2013, the Google Play-store has had over 1 million Android apps published, and over 50 billion apps downloaded.

A developer survey conducted in April–May 2013 found that 71% of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over 1 billion active monthly Android users (that have been active for 30 days), up from 538 million in June 2013.

3. SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

Android Studio and Firebase are used for developing our project which are available everywhere. It provides the technical guarantee of accuracy, reliable and security. The current system develop is technically feasible with all the resources need for development of the apps as well as the maintenance of the same is easy.

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.2, which was released in September 2018.

3.1 FEATURES

The following features are provided in the current stable version:

- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version compatibility and other problems.
- Pro-Guard integration and app-signing capabilities.
- Template-based wizards to create common Android designs and components.
- A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
- Support for building Android Wear apps.
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.
- Gradle-based build support.

- Android Studio supports all the same programming languages of IntelliJ, and PyCharm e.g., Python, and Kotlin and Android Studio 3.0 supports "Java 7 language features and a subset of Java 8 language features that vary by platform version. External projects backport some Java 9 features.

JAVA LANGUAGE

Java is the name of a programming language created by Sun Microsystems in 1995. This company was bought out by Oracle Corporation, which continues to keep it up to date. The latest version is Java SE 9, which came out in 2017.

Java, which was called Oak when it was still being developed, is object oriented, meaning it is based on objects that work together to make programs do their jobs. Java code looks like C, C++, or C#, but code written in those languages will not work in Java in most cases without being changed.

Java runs on many different operating systems, including Android, the world's most popular mobile operating system. This makes Java platform independent. It does this by making the Java compiler turn code into Java bytecode instead of machine code. This means that when the program is executed, the Java Virtual Machine interprets the bytecode and translates it into machine code.

Java was developed to achieve 5 main goals. These are:

- It should be simple, object-oriented, distributed and easy to learn.
- It should be robust and secure.
- It should be independent of a given computer architecture or platform.
- It should be very performant.
- It should be possible to write an interpreter for the language. The language should also support parallelism and use dynamic typing.

XML LANGUAGE

In computing, Extensible Mark-up Language (XML) is a mark-up language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The W3C's XML 1.0 Specification and several other related specifications all of them free open standards define XML.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many applications programming interfaces (APIs) to aid the processing of XML data.

FIREBASE

- Firebase is a Backend-as-a-Service that started as a YC11 start up and grew up into a next-generation app-development platform on Google Cloud Platform.
- Firebase frees developers to focus fantastic crafting user experiences. You don’t need to manage servers. You don’t need to write APIs. Firebase is your server, your API, and your data store, all written so generically that you can modify it to suit most needs. Yeah, you’ll occasionally need to use other bits of the Google Cloud for your advanced applications. Firebase can’t be everything to everybody. But it gets close.
- Real-time data is the way of the future. Nothing compares to it.
- Most databases require you to make HTTP calls to get and sync your data. Most databases give you data only when you ask for it.
- When you connect your app to Firebase, you’re not connecting through normal HTTP. You’re connecting through a Web-Socket. Web-Sockets are much, much faster than HTTP. You don’t have to make individual Web-Socket calls, because one socket connection is plenty. All your data syncs automatically through that single Web-Socket as fast as your client’s network can carry-it.
- Firebase sends you knew data as soon as it’s updated. When your client saves a change to the

data, all connected clients receive the updated data almost instantly.

- Firebase Storage provides a simple way to save binary files most often images, but it could be anything to Google Cloud Storage directly from the client!!!
- Firebase Storage has its own system of security rules to protect your G-Cloud bucket from the masses, while granting detailed write privileges to your authenticated clients.

3.2 PROPOSED SYSTEM

We are going to develop an application to overcome the limitations of the current system. The deaf people face many problems in their day-to-day life. As they cannot hear any kind of voice, they are not able to know what is happening in their surrounding without seeing. So, we decided why don't, we build an application, which will help deaf people to know surrounding noise. Firstly, you need to register yourself in this application with your name, email, phone number and password. After successful registration, login yourself. After login you can see six cards in front of you, named as garden, road traffic, station, environment, home door, explosion. As per place you need to activate card and keep application run in the background. When some noise will be detected, app will automatically show some messages.

3.3 SYSTEM SPECIFICATION:

HARDWARE REQUIREMENTS:

- **Processor** : Windows 10 Core
- **Input Devices** : Android Mobile
- **Ram** : 8GB.

SOFTWARE REQUIREMENTS:

- **Tool** : Android Studio.
- **Coding Language** : JavaScript
- **Back End** : Google Colab

4.SYSYTEM DESIGN

4. SYSTEM DESIGN

3.1 FRONT END:

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud
- Messaging and App Engine

3.2 BACK END:

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform, and it is now their flagship offering for app development.

Firebase evolved from Envolve, a prior startup founded by James Templin and Andrew Le in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Templin and Lee found that it was being used to pass application data that were not chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Templin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in September 2011[4] and it launched to the public in April 2012.

Firebase's first product was the Firebase Realtime Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase's cloud. The product assists software developers in building real-time, collaborative applications.

4. DATAFLOW DIAGRAMS

DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. The development of DFD'S is done in several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The Top-level diagram is often called context diagram. It consists of a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

5.1 DFD LEVEL-0

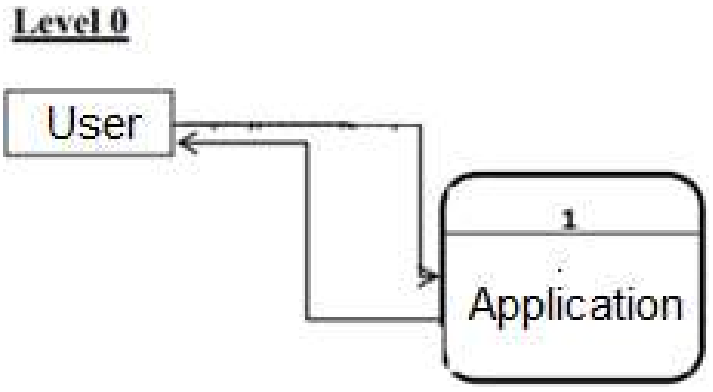


FIGURE 5-1: DFD LEVEL-0

5.2 DFD LEVEL-1

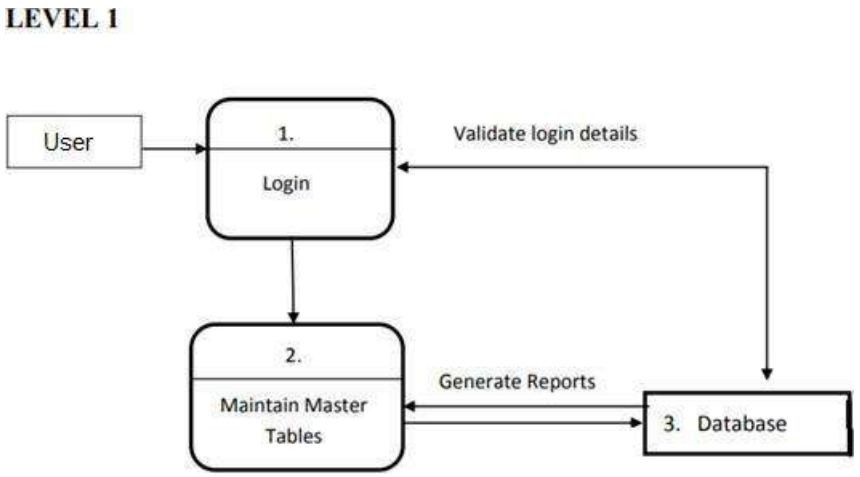


FIGURE 5-2: DFD LEVEL-1

5.3 DFD LEVEL-2

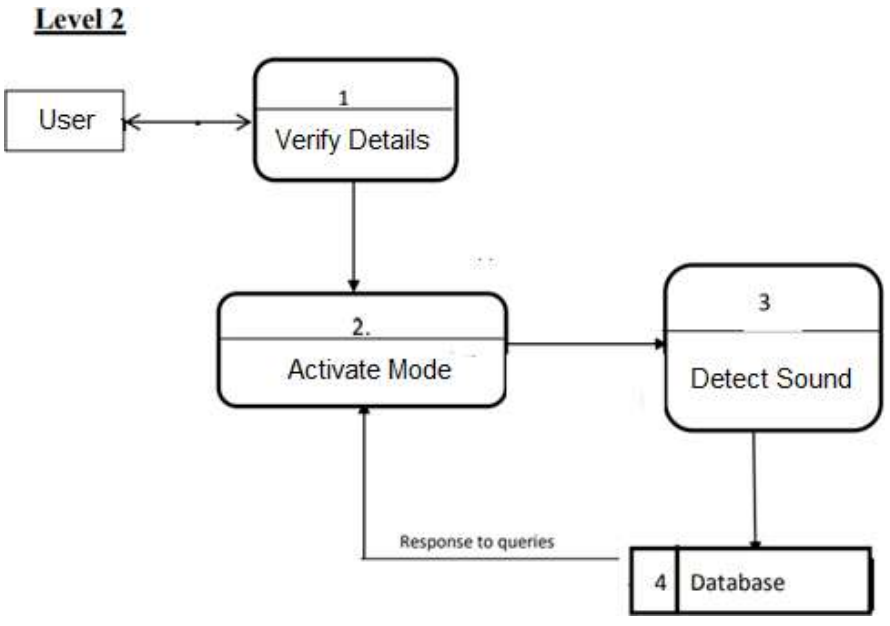


FIGURE 5-3: DFD LEVEL-2

5. ARCHITECTURE

6.1 SYSTEM ARCHITECTURE:

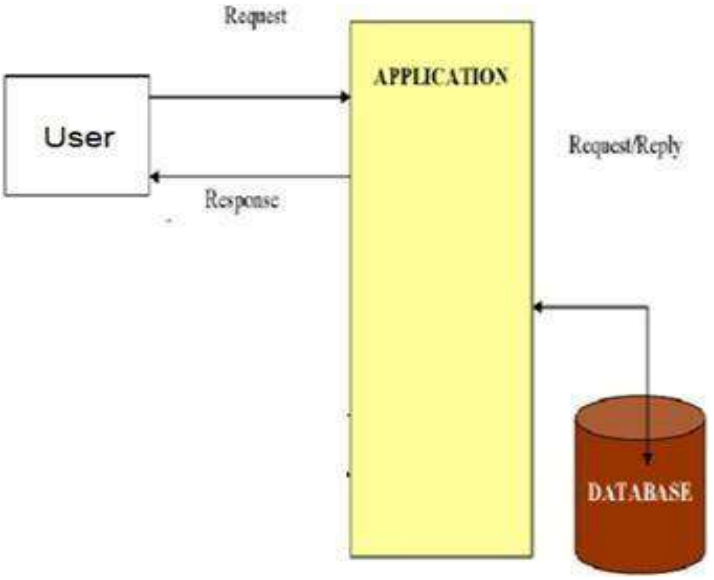


FIGURE 6-1: SYSTEM ARCHITECTURE

6.2 MODULES DESCRIPTION:

- User Module
- Login
- Change password
- Activate cards for sound detection.
- My Profile
- Log Out

1. **User Module**: User will register himself/herself and create their account on application. After successful login, user can use application. User has right to change his/her password. To login to the application, you need to enter the email and password if already have an account and if you don't have an account you can click on the create profile button. Assuming that you are new to the app you can click on the create profile button which will take you to the signup page of the app and there you to have to specify your full name, username, email, password and the confirm password and as you click on the create profile button your account will be created and it will be registered to the database.
2. **Login:** This module allows users to create a new account and then with a valid email id and password user can log in to the account. For account creation, we have used firebase authentication.
3. **Change Password:** The app allows users to reset their passwords if the user forgot. User receive forgot password reset link on the registered mail id. After clicking the link user, can the reset password.
4. **Sound Detection:** This module detects the sound and then performs analysis as well as compares the sound. After analysis, the color is generated.
5. **My Account:** This module allows users to get current user logged information.
6. **Logout:** This module allows users to get logout user from the application.

.

6.3 USE CASE DIAGRAM

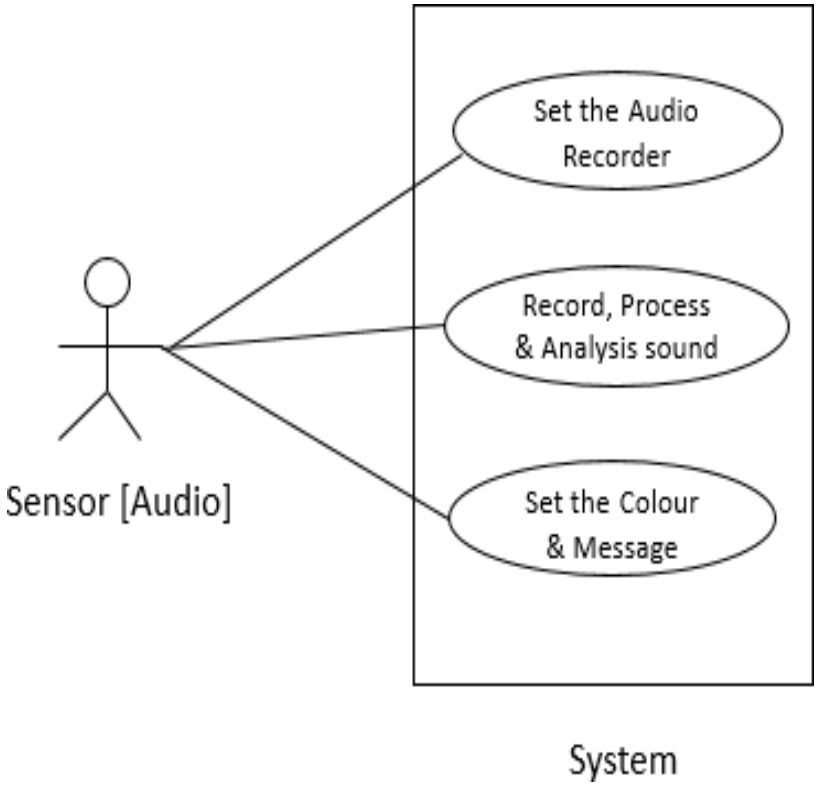


FIGURE 6-1: Use case Diagram for Color Detection for Various Sounds

6.4 SEQUENCE DAIGRAM:

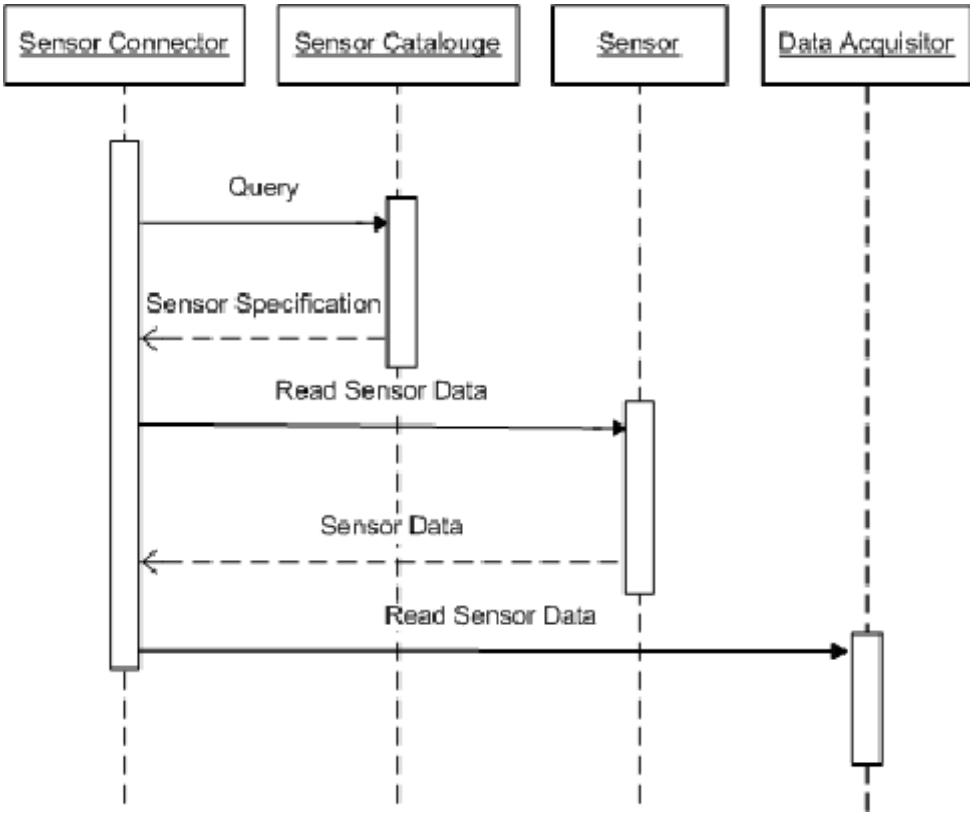


FIGURE 6-2: Sequence Diagram for Color Detection for Various Sounds

6.5 ACTIVITY DIAGRAM:

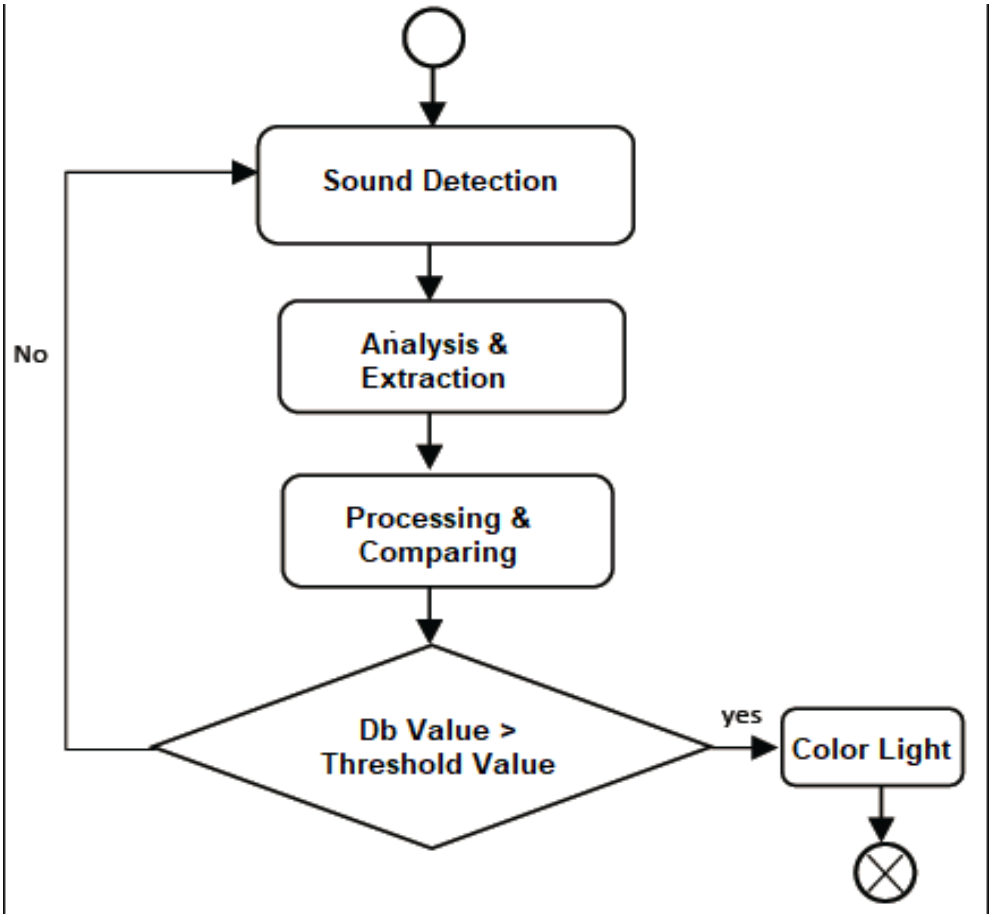


FIGURE 6-3: Activity Diagram for Color Detection for Various Sounds

7. IMPLEMENTATION

IMPLEMENTATION

CODE:

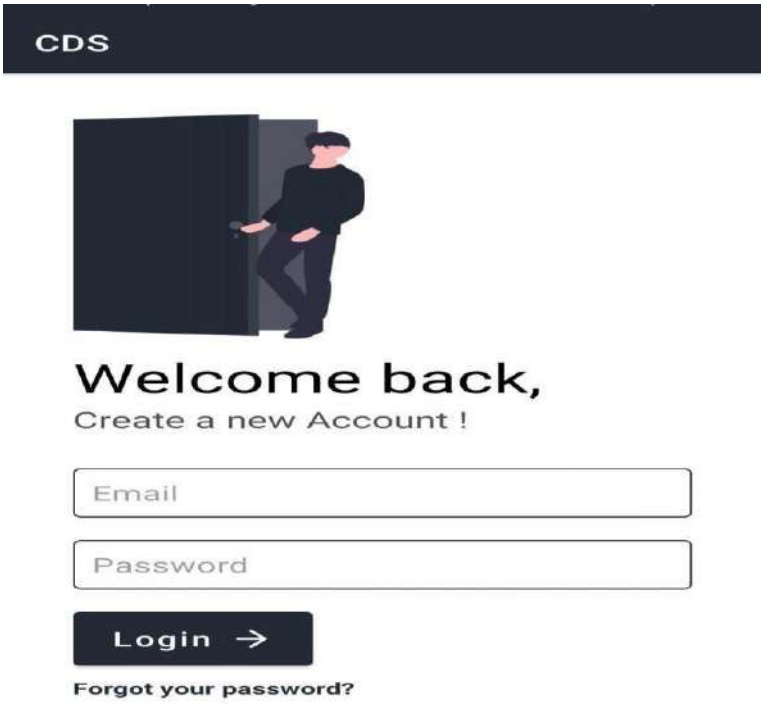
```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.textfield.TextInputLayout;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity {

    EditText email, password;
    Button login;
    TextView register,tvForget;
    boolean isEmailValid, isPasswordValid;
    TextInputLayout emailError, passError;
    private FirebaseAuth firebaseAuth;
    ProgressDialog progressDialog;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        email = (EditText) findViewById(R.id.email);
        password = (EditText) findViewById(R.id.password);
        login = (Button) findViewById(R.id.login);
        register = (TextView) findViewById(R.id.register);
        emailError = (TextInputLayout) findViewById(R.id.emailError);
        passError = (TextInputLayout) findViewById(R.id.passError);
        tvForget = findViewById(R.id.forgot);

        firebaseAuth = FirebaseAuth.getInstance();
        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("LOGGING IN ...");
        login.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
```

```
            SetValidation();
        }
    });

    register.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            // redirect to RegisterActivity
            Intent intent = new Intent(getApplicationContext(), RegisterActivity.class);
            startActivity(intent);
        }
    });

    tvForget.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            startActivity(new Intent(LoginActivity.this,ResetPasswordActivity.class));
        }
    });
  }
  public void SetValidation() {
    if (email.getText().toString().isEmpty()) {
        emailError.setError(getResources().getString(R.string.email_error));
        isEmailValid = false;
    } else if (!Patterns.EMAIL_ADDRESS.matcher(email.getText().toString()).matches()) {
        emailError.setError(getResources().getString(R.string.error_invalid_email));
        isEmailValid = false;
    } else  {
        isEmailValid = true;
        emailError.setErrorEnabled(false);
    }
    if (password.getText().toString().isEmpty()) {
        passError.setError(getResources().getString(R.string.password_error));
        isPasswordValid = false;
    } else if (password.getText().length() < 6) {
        passError.setError(getResources().getString(R.string.error_invalid_password));
        isPasswordValid = false;
    } else  {
        isPasswordValid = true;
        passError.setErrorEnabled(false);
    }

    if (isEmailValid && isPasswordValid) {
        // Toast.makeText(getApplicationContext(), "Successfully",
Toast.LENGTH_SHORT).show();

        firebaseAuth.signInWithEmailAndPassword(email.getText().toString(),
password.getText().toString())
                .addOnCompleteListener(LoginActivity.this, new
OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
```

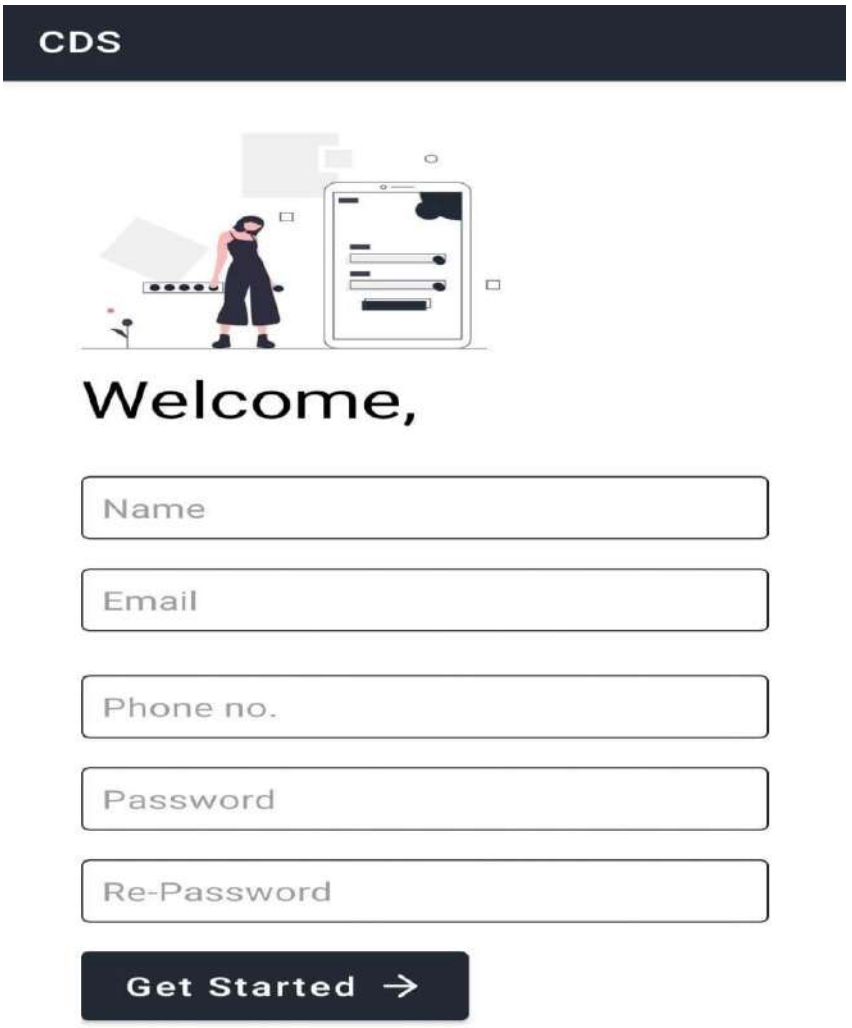
```
                        // Sign in success, update UI with the signed-in user's information
                        progressDialog.dismiss();
                        startActivity(new Intent(getApplicationContext(), MainActivity.class));


                    } else {
                        Toast.makeText(LoginActivity.this, "Login Failed or User Not Found",
Toast.LENGTH_SHORT).show();
                        progressDialog.dismiss();
                    }
                }
            });

        }

    }
    protected void onStart() {
        super.onStart();
        FirebaseUser currentUser = firebaseAuth.getCurrentUser();
        updateUI(currentUser);
    }
private void updateUI(FirebaseUser currentUser) {

        if (currentUser != null) {
            Intent i = new Intent(LoginActivity.this, MainActivity.class);
            i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(i);
        } else {

        }

    }
```

8. SCREENSHOTS

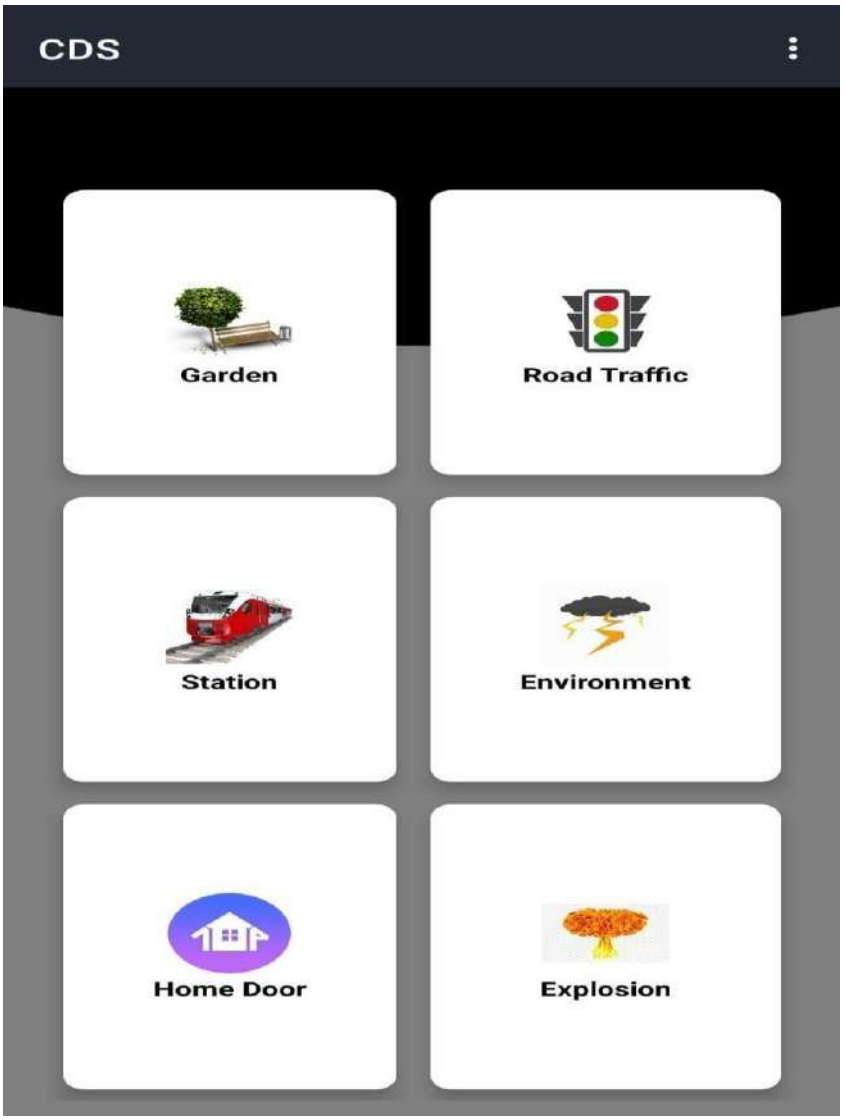
RESULT:



SCREENSHOT 8-1: LOGIN PAGE OF APP



8-2: SIGN UP CREDENTIALS OF APP



SCREENSHOT 8-3: HOME PAGE OF APP

SCREENSHOT 8-4: DETECTION OF THE SOUND PAGE

SCREENSHOT 8-5: MY ACCOUNT PAGE OF THE USER

9. CONCLUSION

9. CONCLUSION

Application software has been computed successfully and was also tested successfully by taking “test cases”. It is user friendly, and has required options, which can be utilized by the user to perform the desired operations. Application software meets the information requirements specified to a great extent. The system has been designed keeping in view the present and future requirements in mind and made very flexible. The goals that are achieved by the software are Instant access, Improved productivity, Optimum utilization of resources, Efficient management of records Simplification of the operations , less processing time and getting required information , User friendly , Portable and flexible for further enhancement.

10. REFERENCES

10. REFERENCES

1. Firebase: https://firebase.google.com/android/setup
2. YouTube: https://www.youtube.com
3. Android: https://www.android.com
4. Stack: https://www.stackoverflow.c

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Color Depiction for Various Sounds

Sharanya Peteri, Keerthana Yadav , Keerth Koumudh

P.Sharanya (CSE,CMR Technical Campus)
R.Keerthana Yadav(CSE,CMR Technical Campus)
A.Keerth Koumudh(CSE,CMR Technical Campus)

Abstract –

Human perception of surrounding events is strongly dependent on audio cues. Thus, acoustic insulation can seriously impact situational awareness. We present an exploratory study in the domain of assistive computing, eliciting requirements and presenting solutions to problems found in the development of an environmental sound recognition system, which aims to assist deaf and hard of hearing people in the perception of sounds. To take advantage of smartphones computational ubiquity, we propose a system that executes all processing on the device itself, from audio features extraction to recognition and visual presentation of results. Our application also presents the confidence level of the classification to the user. A test of the system conducted with deaf users provided important and inspiring feedback from participants.

1. INTRODUCTION

Consciousness of what happens in the surrounding environment is strongly dependent on an individual's capacity to perceive sounds and accurately identify events related to them. Acoustic insulation can seriously compromise the ability of a person to acquire situational awareness, which is important for the execution of daily tasks, social interaction, and even personal safety. Hearing individuals may not realize how much they depend on auditory ability to perceive what is happening around them. Environmental sound awareness is necessary in a much higher number of situations than commonly imagined. Examples of this problem are provided by Matthews et al, who propose an environmental sound recognition (ESR) system for deaf users. Among other cases presented in their study, a participant reported that once had forgotten the vacuum cleaner on all night, since the device did not provide any visual cues that it was in operation. Based on previous research, the present document describes the design, development, and test of a mobile ESR system that aims to expand deaf individuals' situational awareness. We present an exploratory study in the field of assistive computing, describing solutions for problems encountered during the development of an ESR system specifically designed for deaf users. Currently there are still few studies on this topic, despite its importance. To foster future works, we formalize the requirements that guided the development of our system and provide details about the implementation of our solution.

In general terms, existing studies on sound recognition are divided into three categories: speech, music, and environmental sound. In the latter case, the most common approach is the use of predefined environmental sound classes, which can then be applied to the indexing/retrieval of audio/video documents and in surveillance systems, for instance. In our study, due to the diverse and ever-changing nature of sounds that the system is supposed to cover, we had to address the open-set problem regarding ESR as well as the representation of uncertainty. During recognition tasks, with the system being executed in an environment where there is no control over the occurrence of sounds, results can be quite inconsistent. One alternative to minimize this problem is to provide the user with information on the confidence level for the classification results.

2. Body of Paper

This application helps the deaf people to understand what is happening in the surrounding. This application will detect noise from surrounding and display message of detected noise. So that user can know, what is happening in the surrounding. We get signals from our surrounding and our brain process those signals. We hear those signals via our ear. Keeping these important words in mind we present this project to mainly focus on aiding the speech impaired and paralyzed patients. Android is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers, with specialized user interfaces for televisions (Android TV), cars (Android Auto), and wrist watches (Android Wear).

The OS uses touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, and a virtual keyboard. Despite being primarily designed for touchscreen input, it also has been used in game consoles, digital cameras, and other electronics.

Android is the most popular mobile OS. As of 2013, Android devices sell more than Windows, iOS, and Mac OS devices combined, with sales in 2012, 2013 and 2014 close to the installed base of all PCs. As of July 2013, the Google Play-store has had over 1 million Android apps published, and over 50 billion apps downloaded. A developer survey conducted in April–May 2013 found that 71% of mobile developers develop for Android. At Google I/O 2014, the company revealed that there were over 1 billion active monthly Android users (that have been active for 30 days), up from 538 million in June 2013. Android Studio and Firebase are used for developing our project which are available everywhere. It provides the technical guarantee of accuracy, reliable and security. The current system develop is technically feasible with all the

resources need for development of the apps as well as the maintenance of the same is easy.

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.2, which was released in September 2018.

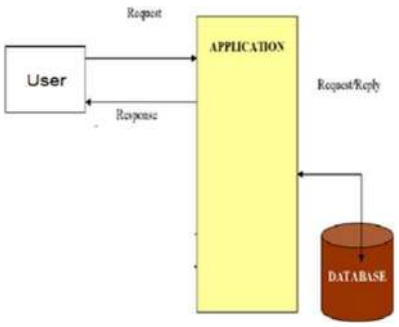


Fig-1: System Architecture

We are going to develop an application to overcome the limitations of the current system. The deaf people face many problems in their day-to-day life. As they cannot hear any kind of voice, they are not able to know what is happening in their surrounding without seeing. So, we decided why don't, we build an application, which will help deaf people to know surrounding noise. Firstly, you need to register yourself in this application with your name, email, phone number and password. After successful registration, login yourself. After login you can see six cards in front of you, named as garden, road traffic, station, environment, home door, explosion. As per place you need to activate card and keep application run in the background. When some noise will be detected, app will automatically show some messages. Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform, and it is now their flagship offering for app development.

Firebase evolved from Envolve, a prior startup founded by James Templin and Andrew Le in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Templin and Lee found that it was being used to pass application data that were not chat messages.

Developers were using Envolve to sync application data such as game state in real time across their users. Templin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in September 2011[4] and it launched to the public in April 2012.

Key features

- Android-specific refactoring and quick fixes
- Lint tools to catch performance, usability, version.
- Pro-Guard integration and app-signing capabilities.
- Template-based wizard to create common Android design

3. CONCLUSION

Application software has been computed successfully and was also tested successfully by taking "test cases". It is user friendly, and has required options, which can be utilized by the user to perform the desired operations. Application software meets the information requirements specified to a great extent. The system has been designed keeping in view the present and future requirements in mind and made very flexible. The goals that are achieved by the software are Instant access, Improved productivity, Optimum utilization of resources, Efficient management of records Simplification of the operations , less processing time and getting required information , User friendly , Portable and flexible for further enhancement.

ACKNOWLEDGEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide

Dr. G. Soma Sekhar , Assistant Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) **Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Dr. Suwarna Gothane, Mr. A. Uday Kiran, Mr. A. Kiran Kumar, Mrs. G. Latha** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We also express our sincere gratitude to Sri. Ch. Gopal Reddy,

Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project

REFERENCE

1. Lawrence Rabiner and Biing-Hwang Juang, "Fundamental of Speech Recognition", Prentice-HalEnglewood Cliffs, N.J., 1993. Shannon B.J., Paliwal K.K., "A comparative study of filter bank spacing for speech recognition", Proc.of Microelectronic engineering research conference, Brisbane, Australia, Nov. 2003.
2. Y. Linde, A. Buzo R. Gray, "An algorithm for vector quantizer design" IEEE Transactions on Communications Vol. 28, pp.84-95, 1980.
   F. Soong, E. Rosenberg, B. Juang, and L. Rabine, "A Vector Quantization Approach to Speaker Recognition", ATT Technical Journal, vol. 66, March/April 1987.
3. Zhong-Xuan, Yuan Bo-Ling, Xu Chong-Zh Yu, "Binary Quantization of Feature Vector for Robust Text-Independent Speaker Identification in IEEE Transactions on Speech and Audio Processing", Vol. 7, No. 1, Ja.
   Chachada, S., & Kuo, C. (2014). Environmental sound recognition: A survey. APSIPA Transactions on Signal and Information Processing, 3, E14. doi:10.1017/ATSIP.2014.12.
4. Sharma, Jivitesh & Granmo, Ole-Christoffer & Goodwin, Morten. Environment Sound Classification sing Multiple Feature Channels and Deep Convolutional Neural Networks. (2019).
5. Esmaeilpour, M., Cardinal, P., & Koerich, A. L. Unsupervised feature learning for environmental sound classification using cycle consistent generative adversarial network.
   Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ... & Jaitly, N. State-of-the-art speech recognition with sequence-to-sequence models. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4774-4778). IEEE nuary 1999. IEEE, New York, NY, U.S.A (2018, April).