MINI PROJECT

(2021-22)

“TRANSLATOR APP”

Project Report

Institute of Engineering & Technology

Submitted By –

Shivani Balyan(191500770)

Suraj Singh Chauhan(191500829)

Yogita Balyan (191500944)

Under the Supervision of

Mr. Akash Kumar Choudhary

Technical Trainer

Department of Computer Engineering & Applications

Department of Computer Engineering
and Applications GLA University,
17km. Stone NH#2, Mathura-Delhi
Road,Chaumuha, Mathura – 281406
U.P (India)

Declaration

I/we hereby declare that the work which is being presented in the Bachelor of technology. Project **“TRANSLATOR App”**, in partial fulfillment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mr. Akash Kumar choudhary, Technical Trainer, Dept. of CEA,GLA University.**

The contents of this project report, in full or in parts, have not been submittedto any other Institute or University for the award of any degree.

Sign: ShivaniBalyan

Name of Candidate: Shivani Balyan

University Roll No.:191500770

Sign: SurajChauhan

Name of Candidate: Suraj Chauhan

University Roll No.:191500829

Sign: YogitaBalyan

Name of Candidate: YogitaBalyan

University Roll No.:191500944

Department of Computer Engineering
and Applications GLA University, 17
km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P
(India)

Certificate

This is to certify that the project entitled " **TRANSLATOR App**", carried out in Mini Project – I Lab, is a bonafide work by Shivani Balyan, Yogita Balyan, Suraj Chauhan and is submitted in partial fulfilment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor: Mr. Akash Kumar Choudhary

Date:

Department of Computer Engineering
and Applications GLA University, 17
km. Stone NH#2, Mathura-Delhi Road,
Chaumuha, Mathura – 281406 U.P
(India)

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we would like to place my deep gratitude to GLA University for providing us the instructor Mr. Akash Kumar Choudhary, our technical trainer and supervisor.

He has been helping us since Day 1 in this project. He provided us with the roadmap, the basic guidelines explaining on how to work on the project. He has been conducting regular meeting to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And at last but not the least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Sign: *Shivani Balyan*

Name of Candidate: Shivani Balyan

University Roll No.:191500770

Sign: *Suraj Singh Chauhan*

Name of Candidate: Suraj Singh Chauhan

University Roll No.:191500829

Sign: *Yogita Balyan*

Name of Candidate: Yogita Balyan

University Roll No.:191500944

ABSTRACT

In this project, we are creating an android application, basically a Book Finder App which we have named Bookopedia. A mobile translator is a mobile application that can be utilised for translating from English to any other dialect, and vice versa. The problem of language difference has hindered effective information communication over the years. There have been difficulties in information communication amid countries over the years. In modern times, language interpreters must understand and speak both the language been translated to and verse-visa. This traditional approach used for solving the problem of language differences has not been productive and favourable. Also, the teaching of different languages can be difficult due to language difference problems. The individual will also have to be taught by a tutor who will incur extra expenses and may not be the most efficient and favourable method. Therefore, the study develops an android phone language converter app in other to make learning and language translation easy and facilitates stress-free communication. The proposed language translation uses Google's real-time translation API natural language processing with Java programming language to develop the application. The most used languages globally (i.e., English, Spanish, Arabic, Hindi, French, and Chinese) were used for the android application translation. This application can be useful for Tourists for communication purposes, thus allowing them to integrate with the local people and access the right information. The system will also be able to evaluate language translation to determine their suitability for

everyday conversation; given the fact that it is an android application, one will always be willing to use their phone to learn, compared to having them on a computer or learn from a physical tutor when your phone can be your tutor. The application was evaluated based on the classification time the memory usage, and the battery life all through distinctive use.

Android App ecosystem is diverse and is changing people’s life all over the world. Android users are expected to increase because of the advance changes of the operating system and the way it deals with issues and compatibility with other mobile devices. Furthermore, designing solutions for the problems that we may face in future is essential. Like this application definitely stands the need of students at any time at their fingertips without any barrier of place**.**

CONTENTS

LIST OF FIGURES

CHAPTER-1

INTRODUCTION

This Android Application “Translate plus” has been submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering at GLA University, Mathura supervised by Mr. Akash Kumar Choudhary. This project has been completed approximately three months and has been executed in modules, meetings have been organised to check the progress of the work and for instructions and guidelines.

• MOTIVATION

With the rapid evolution of the online market, serving a global client base is now no more limited within geographical boundaries or borders of only one large enterprise. As an increasing number of smaller companies enter the foreign market to assert their presence worldwide, the need to overcome language barriers are as a result higher than ever.

Choosing the right translation service is crucial because many things can be misinterpreted or lost. Miscommunication in translation can harm your business in more than one way. Apart from financial loss, it may result in reputation damage and make you vulnerable to industrial disasters.

As a result, companies spend a significant amount on translation services to ensure clear communication within their network of customers, partners, agents, and government agencies. According to a survey, the amount of money enterprises spends on translation services every year is expected to reach $45 billion by 2020.

The rise in globalization has led to the growth of artificial intelligence (AI) to bring down the cost of translation. Today we have AI enabled translation apps like Google Translate, Amazon Translate, and Microsoft Translator that have highly evolved in the recent times to deliver accurate translation.

OBJECTIVE

The main objective of this application is to create a Translator app named "Translate Plus" which will have improved dramatically over the last few years. A recent development in apps is the inclusion of voice-recognition software that enables them to decode words and phrases as you speak into text or audio format. The more they are used, the better they are able to interpret text and sound.

Translation apps vary widely in terms of user interface, price, functionality, and features. Their accuracy primarily depends on your dialect, the words you use, and the environment. Some apps may be better at translating to Spanish, but fare poorly in French; while others may be great with technical words but perform miserably when you use culinary terms.

To get the best results from voice-recognition software, make sure you speak slowly and form short sentences. The apps that stand out in terms of usability and reliability are Google Translate, Jibbigo, I translate, and Vocre.

Translation apps have helped companies in the retail sector and service industry reach out to larger markets and serve people from different countries. Smaller enterprises find it easier to offer their products/ services globally without having to worry about high translation costs.

- ## SOURCES

  The source of our project (including all the project work, documentations and presentations) will be available at the following link https://github.com/suraj8941/TranslatePlus

CHAPTER -2

SOFTWARE REQUIREMENT ANALYSIS

- **IMPACT OF BOOKS ON DAILY LIFE**

Translation is saving lives. Translators Without Border's Words of Relief Crisis Response Network is a global translation and localisation initiative that is leveraging both human and technological resources. The project builds capacity to facilitate and improve communication among victims, field workers, and relief agencies during and after crises.

Translation is creating jobs. The translation market it will be worth $37 billion by 2018 (Source: IBIS World). There are numerous small companies offering translation and interpreting services around the globe and they are not only employing translators and interpreters but hiring a sales force, people who work in marketing, project managers, accountants, lawyers and so on.

Translation keeps the economy growing. When doing business globally, the key element is mutual understanding, which is made possible generally by professional translators and interpreters. Companies that are able to communicate effectively in more than one language are most likely to sell their products and services all around the world, consequently boosting the economy.

- **PROBLEM STATEMENT**

The Book Finder App “Translate Plus” is an Android Application which will allow the users to search any words using either book name, author of the book, the subject into which the book falls in or any other keyword related to the book. Actually this app is connected to the Google Translate API and every time the user searches for the Words, words is searched on the API , and a few more details are reverted back to the user’s screen.

Along the side, for the users a library space is being provided for them to store the words they like in the “Favourites Section”. As a help to the users, there will be FAQ Section containing the basic details and guidelines on how to use the app. The profile of each user is created and can be updated anytime as per the requirements.

This app is a complete translator with all the facilities a user desires and with the proper User Interface as well.

• HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirement

- Processor: intel i5

- Operating System: Any Operating System
- RAM: 8 GB (or higher)
- Hard disk: 256GB

Software Requirement

- Software used: Android Studio
- Language used: Kotlin, XML
- Database: Firebase
- User Interface Design: Android Application

• BOOKOPEDIA ON ANDROID APPLICATION

translate Plus is actually a library of Words. Most of us find a use for the translation services while traveling. Google's technology is getting better every day. Now, translate whole sentences and phrases in context. You also don't have to keep rummaging around for the Google Translate app. Just tap on the floating bubble and start your conversation.

But when you are an international traveller, it helps to have alternatives to Google Translate on your phone. There are other mobile translation apps to convert any language to your preferred language of choice.

CHAPTER- 3

SOFTWARE DESIGN

- ### USE-CASE DIAGRAM:

Along with the development of information technology in recent era, a number of new applications emerge, especially on mobile phones. The use of mobile phones, besides as communication media, is also as media of learning, such as translator application. Translator application can be a tool to learn a language, such as English to Bahasa Indonesia translator application. The purpose of this research is to allow user to be able to translate English to Bahasa Indonesia on mobile phone easily. Translator application on this research was developed using Java programming language (especially J2ME) because of its advantage that can run on various operating systems and its open source that can be easily developed and distributed. In this research, data collection was done through literature study, observation, and browsing similar application. Development of the system used object- oriented analysis and design that can be described by using case diagrams, class diagrams, sequence diagrams, and activity diagrams. The translation process used rule-based method.

Result of this research is the application of Java-based translator which can translate English sentence into Indonesian sentence. The application can be accessed using a mobile phone with Internet connection. The application has

spelling check feature that is able to check the wrong word and provide alternative word that approaches the word input. Conclusion of this research is the Conclusion of this research is the application can translate sentence in daily conversation quite well with the sentence structure corresponds and is close to its original meaning.

- **SEQUENCE DIAGRAM**

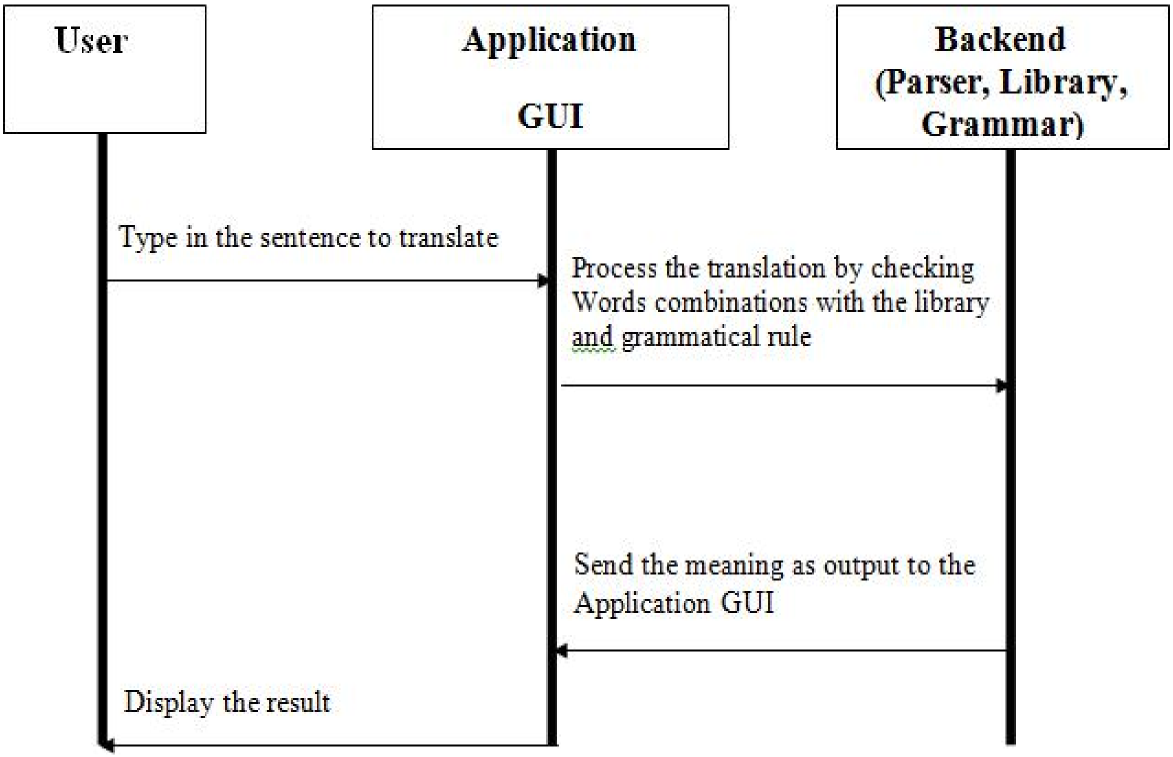

User
Application
GUI
Backend
(Parser, Library,
Grammar)
Type in the sentence to translate
Process the translation by checking
Words combinations with the library
and grammatical rule
Send the meaning as output to the
Application GUI
Display the result

Figure-4: Sequence Diagram

CHAPTER-4

TECHNOLOGY USED

- **ANDROID**

Android is a Linux-based operating system designed primarily for touch screen devices such as smart phone tablets and computers. Released in 2008, is now owned by Google. So android is an operating system like Windows, Ubuntu and Mac OS and a lot number of devices use Android these days like mobile phones, watches, laptop and television. So, we also created an android application “Translate Plus”, a library of Words. Play Store is a market place for all the Android Apps. So, we need to know what basically an android app is. An Android app is software running on a Android Platform. So, this can be concluded that like all the software it is a combination of Backend and Frontend. Backend to design the logical parts of the app, for the functionality whereas Front End to develop the User Interface. And to implement the various parts of the android app, we require a number of tools and technologies which will come into picture. But first it would be great to see the three different types of Android Apps: -

- **Native Apps:** An executable program coded in the machine language of the hardware platform it is running in. **Native**

applications are compiled into the machine language of that CPU. For example, **Windows** and Mac executable **apps** are in x86 machine language, while **mobile apps** are ARM based. Native apps are the most common. They're coded in a specific language like Swift for **iOS** or Java for Android. A popular example isWhatsApp.

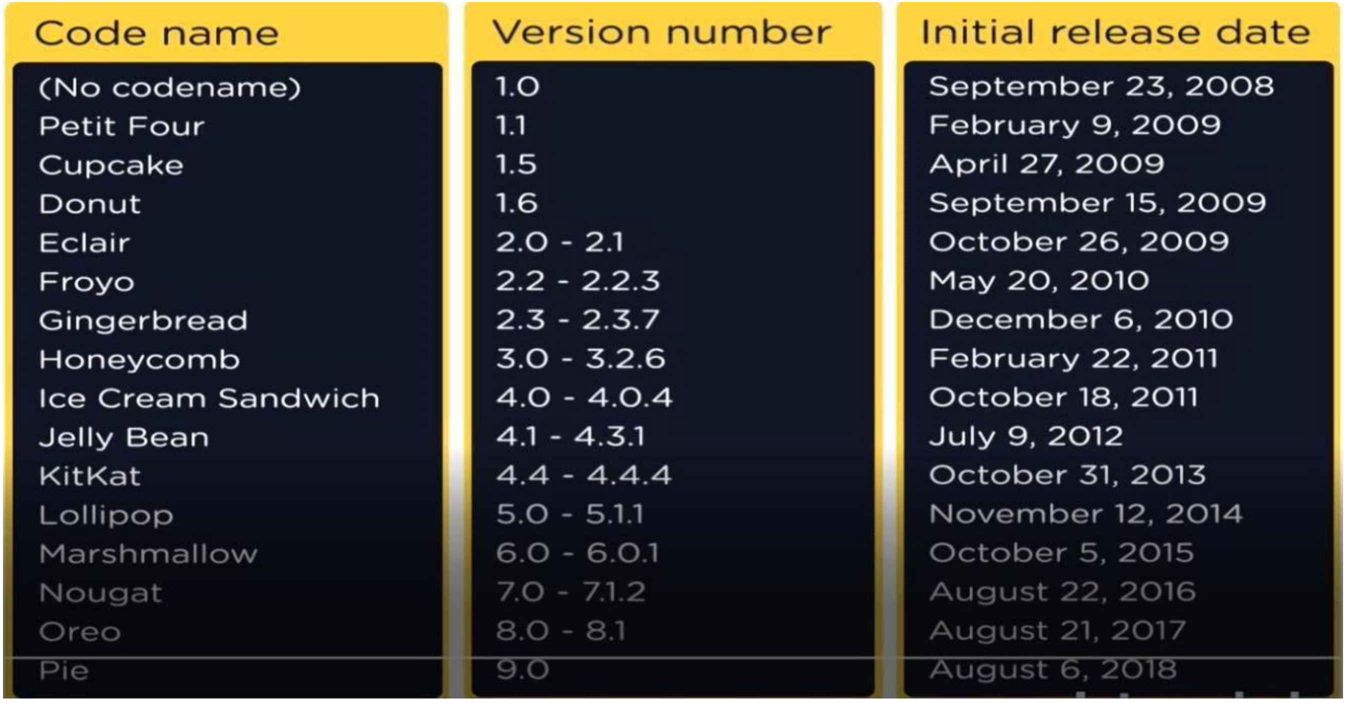
- **Web Apps:** are accessed via the internet browser and will adapt to whichever device you’re viewing them on. They are not native to a particular system, and don’t need to be downloaded or installed. Due to their responsive nature, they do indeed look and function a lot like mobile apps — and this is where the confusion arises.

- **Hybrid Apps:** Hybrid apps are deployed in a native container that uses a mobile Web View object. When the app is used, this object displays web content thanks to the use of web technologies (CSS, JavaScript, HTML, HTML5). It is in fact displaying web pages from a desktop website that are adapted to a Web View display. The web content can either be displayed as soon as the app is opened or for certain parts of the app only i.e. for the purchase funnel. In order to access a device's hardware features (accelerometer, camera, contacts…) for which the native apps are installed, it is possible to include native elements of each platform’s user interfaces (iOS, Android): native code will be used to access the specific features in order to create a seamless user experience. Hybrid apps can also rely on platforms that offer JavaScript APIs if those functionalities are called within a Web View

- VERSION OF ANDROID

Each year Android releases a new version with better features, better security and better User Interface experience and a new symbol. Here is the table of list of versions

Figure-5: **Android KitKat**



Code name	Version number	Initial release date
(No codename)	1.0	September 23, 2008
Petit Four	1.1	February 9, 2009
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15, 2009
Eclair	2.0 - 2.1	October 26, 2009
Froyo	2.2 - 2.2.3	May 20, 2010
Gingerbread	2.3 - 2.3.7	December 6, 2010
Honeycomb	3.0 - 3.2.6	February 22, 2011
Ice Cream Sandwich	4.0 - 4.0.4	October 18, 2011
Jelly Bean	4.1 - 4.3.1	July 9, 2012
KitKat	4.4 - 4.4.4	October 31, 2013
Lollipop	5.0 - 5.1.1	November 12, 2014
Marshmallow	6.0 - 6.0.1	October 5, 2015
Nougat	7.0 - 7.1.2	August 22, 2016
Oreo	8.0 - 8.1	August 21, 2017
Pie	9.0	August 6, 2018

Table -1: Versions of Android

- TOOLS AND LANGUAGES

Tools used to build the Android App are: -

- **Android Studio**: Android Studio is an environment that help us create and edit Android applications. It is the official IDE for Android App Development. It has IntelliJ’s powerful code editor and developer tools and various features that enhance productivity while developing apps.

- **Software Development Kit (SDK)**: Android Studio requires a collection of libraries and data therefore SDK is mandatory.

Languages used in building an Android Application are classified as per the Front End and Back End. For designing the Front End of an application we have used XML and for designing the Back End we have used Kotlin.

- **XML:** XML is the extensible Markup Language. It is the met language which allows users to define their own customized markup language especially in order to display documents on Internet. It is the language that contains tags that store information. And the tags can be used to present data on the screen.

- **Kotlin:** Kotlin is statically typed programming language based on Java Virtual Machine. Kotlin is the fundamental language of Android since 2017 as declared by Google. Developers of Android also prefer to use Java for the backend but Kotlin has a upper-hand due to many features like Java has a length syntax and hence sometimes the code is also redundant. To remove the boiler Plate code, Kotlin is preferred. Kotlin is cross platform, general purpose programming language with type inference. It can inter-operate fully with Java but type inference allows its syntax to be more concise.

- BASIC TERMINOLOGY

- **Layout:** Layout is the parent of view. It arranges all the views in a proper manner on the screen.

- **Activity**: An activity can be referred as your device's screen which you see. User can place UI elements in any order in the created window of user's choice.

- **View**: A view is an UI which occupies rectangular area on the screen to draw and handle user events.

- **Emulator**: An emulator is an Android virtual device through which you can select the target Android version or platform to run and test your developed application.

- **Manifest file**: Manifest file acts as a metadata for every application. This file contains all the essential information about the application like app icon, app name, launcher activity, and required permissions etc.

- **API:** Short for Application Programming Interface. APIs are the functions those developers can call on to access specific features by calling upon programs, code, and services that others have written. For example, if a developer wants to draw a button on the screen, she can insert a small bit of code that says "draw this kind of button, with this colour and size and style, at this location" instead of dozens of lines of code that tells the graphics processor, in detail, exactly how to draw a button. If the application wants your location, it can use the location API to "get the device's location" and let Google's code handle the rest, instead of requiring the developer to build an entire location service from scratch just for her own app. There are thousands of APIs in Android, covering everything from drawing interface elements, to the cameras, to location access, to accessing storage, to 3Dgraphics (see: OpenGL ES) and much more.

- **Intent:** Intents are an essential part of the Android ecosystem. They are used to express an action to be performed. Intents allow you to interact with components from the same applications as well as with components contributed by other applications. It can be classified into implicit and explicit intents.

- **Implicit intent:** It does not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it.

- **Explicit Intent:** It specifies the component to start by name. You'll typically use an explicit intent to start a component in your own app, because you know the class name of the activity or service you want to start.

- **APK**: Short for "Android application package." The extension used in Android app installation files (e.g., app.apk). Similar in nature to an EXE file on Windows.

- **SDK:** Short for "Software Development Kit." As it pertains to Android, the SDK is a set of tools such as code libraries, a debugger, and a handset emulator that can be run on Windows, Mac, or Linux to facilitate the creation of Android apps by developers. While the SDK is generally intended for use by developers, end users can install the software on their home computer to execute ADB and Fast boot commands.

- **Action Bar**: The action bar is an important design element, usually at the top of each screen in an app that provides a consistent familiar look between Android apps. It is used to provide better user interaction and experience by supporting easy navigation through tabs and drop-down lists.

- **Navigation bar**: Android Navigation Drawer is a sliding left menu that is used to display the important links in the application. Navigation drawer makes it easy to navigate to and fro between those links. It's not visible by default and it needs to opened either by slidingfrom left or clicking its icon in the Action Bar.

- **Fragment**: A Fragment represents a behavior or a portion of user interface in a Fragment Activity. You can combine multiple fragments in a single activity to build a multi-pane UI and reuse a fragment in multiple activities.

- **Firebase** is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program,

  which stores data in JSON-like documents. Firebase has three core services: a real-time database, user authentication and hosting. With the Firebase iOS SDK, you can use these services to create apps without writing any server code.

  JSON stands for JavaScript Object Notation. It is an independent data exchange format and is the best alternative for XML. JSON is used for data interchange (posting and retrieving) from the server. Hence knowing the syntax and it’s usability is important. JSON is the best alternative for XML and its more readable by human

CHAPTER -5

IMPLEMENTATION AND USER INTERFACE

Creating an app concept design with screen sketches and functional flow diagrams is the best way to communicate your vision to the mobile app developer. Making the concept clear to the developer is probably the most important factor in successful mobile app development. Yet it is one of the most common problems or obstacles in a mobile app development outsourcing project.

No matter what the marketing and profit goals are or if you are outsourcing an app for your personal use, you need to fully design and document the app concept if you expect a programmer to make your vision a reality. Developers are not mind readers and even descriptions given during conversations can be very fleeting or interpreted differently. Fully documenting your concept, therefore, leaves little to chance. The two most important things to do are: A) make a comprehensive description of how the app works and what it does (functionality) and B) create a comprehensive description of what the user sees and does (look and feel).

- **Implementation of the Translate Plus:**

We will be building a simple application in which we will be showing an edit text field and we will add any input to that text field. Along with that, we will be displaying a button to translate that text to the German language. After clicking that button our text will be translated to the German language which we can get to see in the text view. A sample video is given below to get an idea about what we are going to do in this article. Note that we are going to implement this project using the Java language.

Step by Step Implementation

Step 1: Create a New Project

To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio. Note that select Java as the programming language.

Step 2: Connect your app to Firebase

After creating a new project in Android Studio connect your app to Firebase. For connecting your app to firebase. Navigate to Tools on the top bar. After that click on Firebase. A new window will open on the right side. Inside that window click on Firebase ML and then click on Use Firebase ML kit in Android. You can see the option in the below screenshot.

After clicking on this option, you will get to see the below screen. On this screen click on Connect to Firebase option to connect your app to Firebase. Click on Connect option to connect your app to Firebase and add the below dependency to your build.

Gradle file.

Step 3: Adding dependency for language translation to build. Gradle file

Navigate to the Gradle Scripts > build. Gradle(Module :app) and add the below dependency in the dependencies section.

Step 4: Adding permissions to access the Internet in your Android App

Navigate to the app > AndroidManifest.xml file and add the below code to it. Comments are added in the code to get to know in more detail.

Step 5: Working with the activity_main.xml file

Navigate to the app > res > layout > activity_main.xml and add the below code to that file. Below is the code for the activity_main.xml file.

Flow Chart for the User is given below:

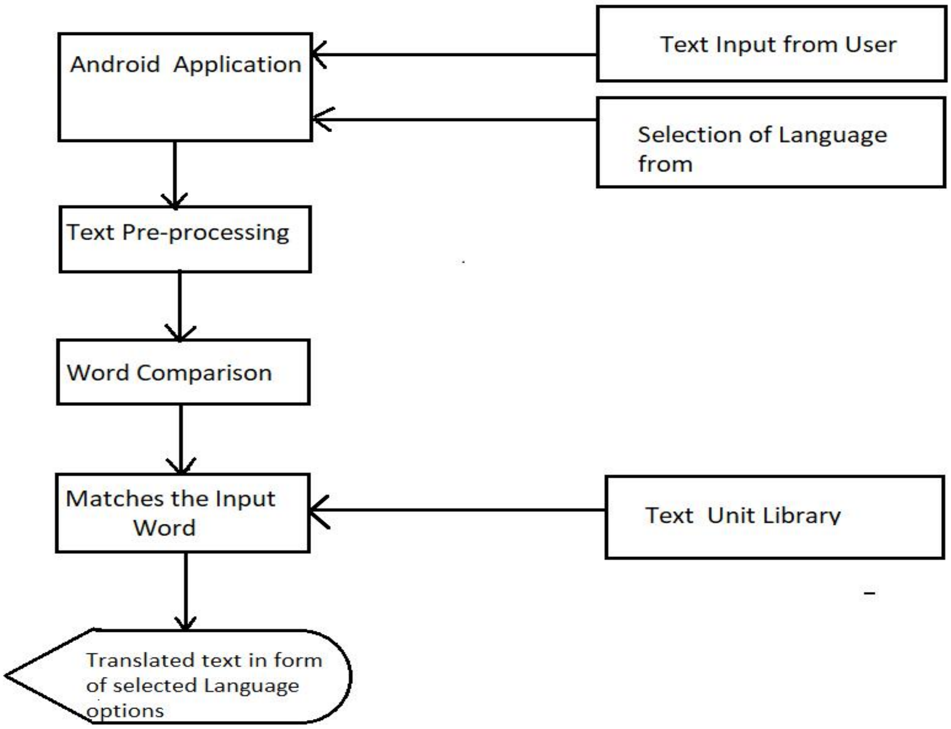


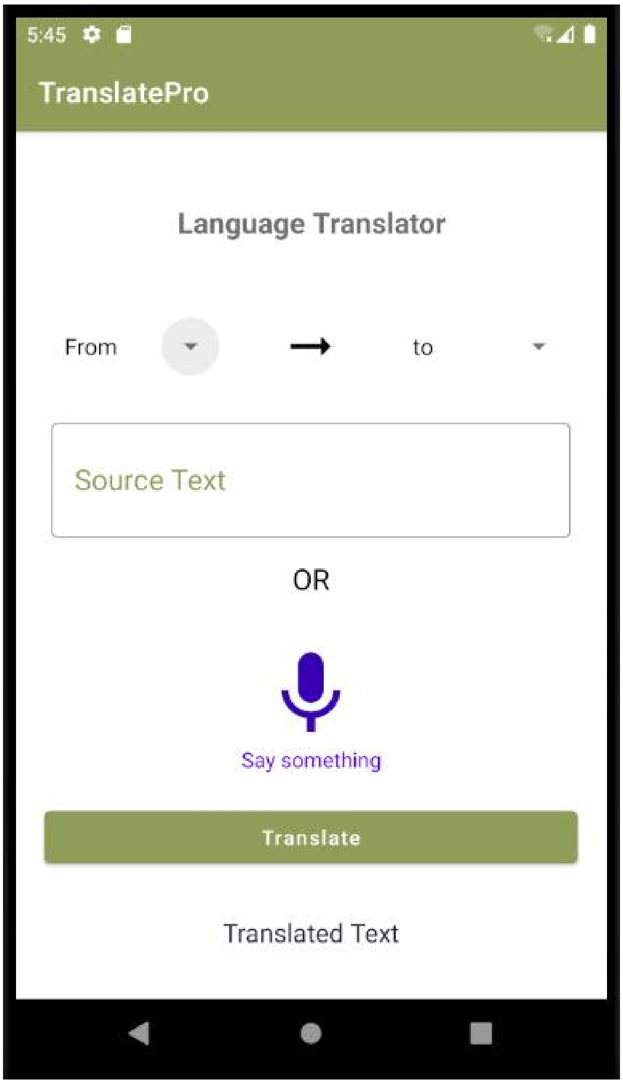
Figure-6: Flow Chart for User

Step to be followed by the user

1. Open an app with text you can copy.
2. Highlight the text you want to translate › **Copy**.
3. On your current screen, tap Google Translate .
4. Choose the language you want.

User Interface

. Home page



5:45

TranslatePro

Language Translator

From → to

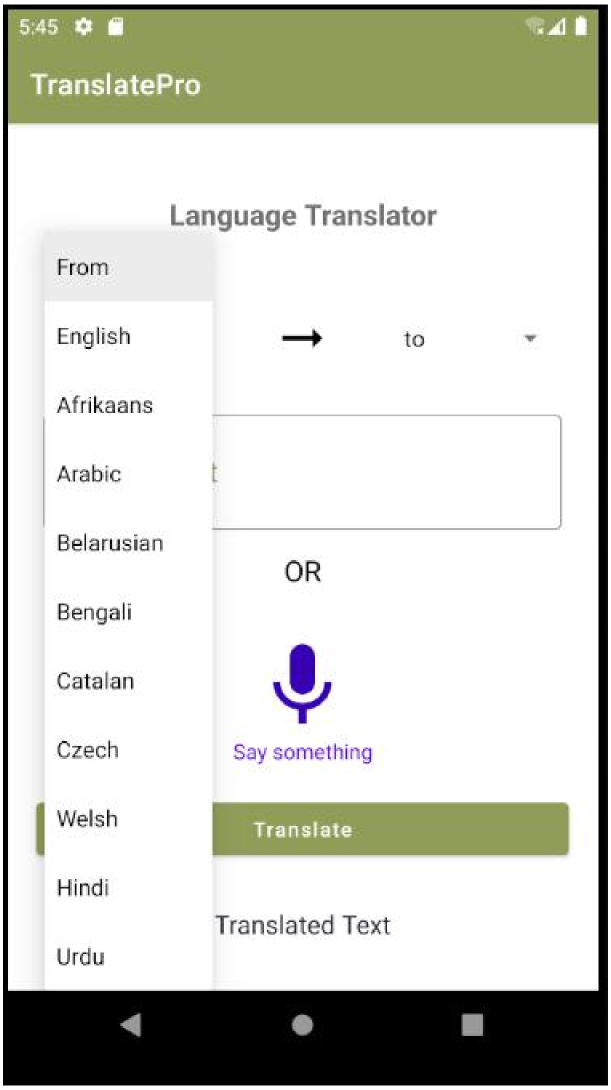
Source Text

OR

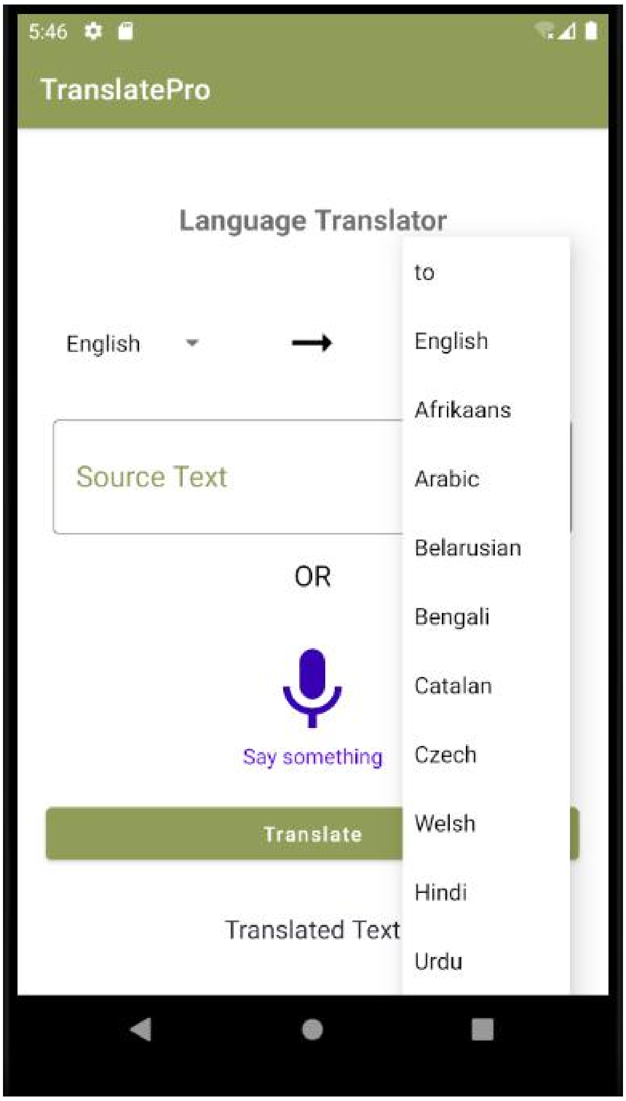
Say something

Translated Text

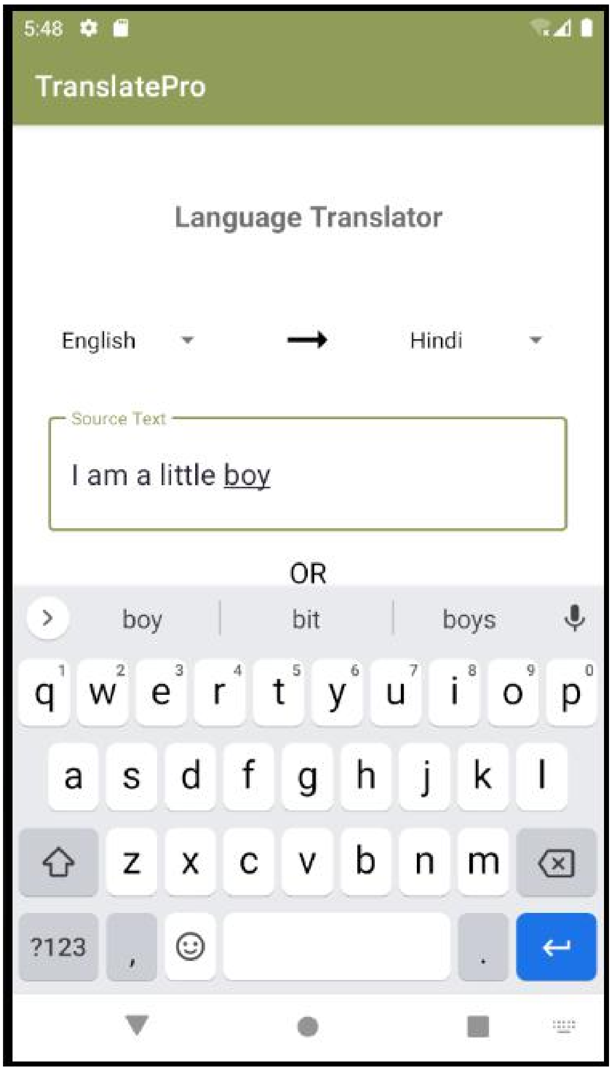
. Primary level selection



.Target language selection


5:46
TranslatePro
Language Translator
English
to
English
Afrikaans
Arabic
Belarusian
Bengali
Catalan
Czech
Welsh
Hindi
Urdu
Source Text
OR
Say something
Translate
Translated Text

. Text input



5:48

TranslatePro

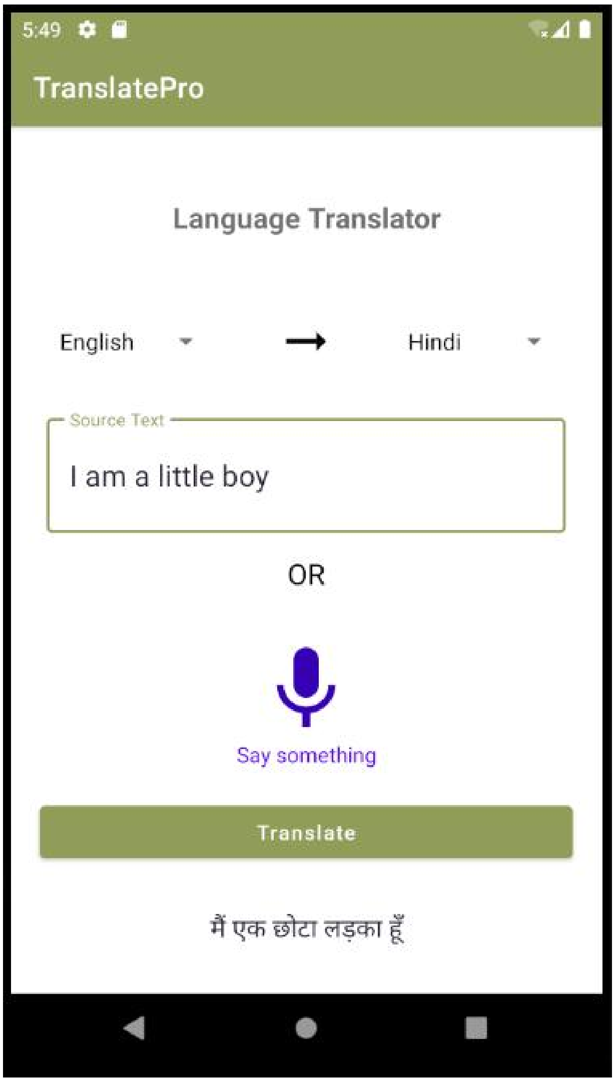
Language Translator

English → Hindi

Source Text: I am a little boy

OR

. Final result



5:49

TranslatePro

Language Translator

English → Hindi

Source Text

I am a little boy

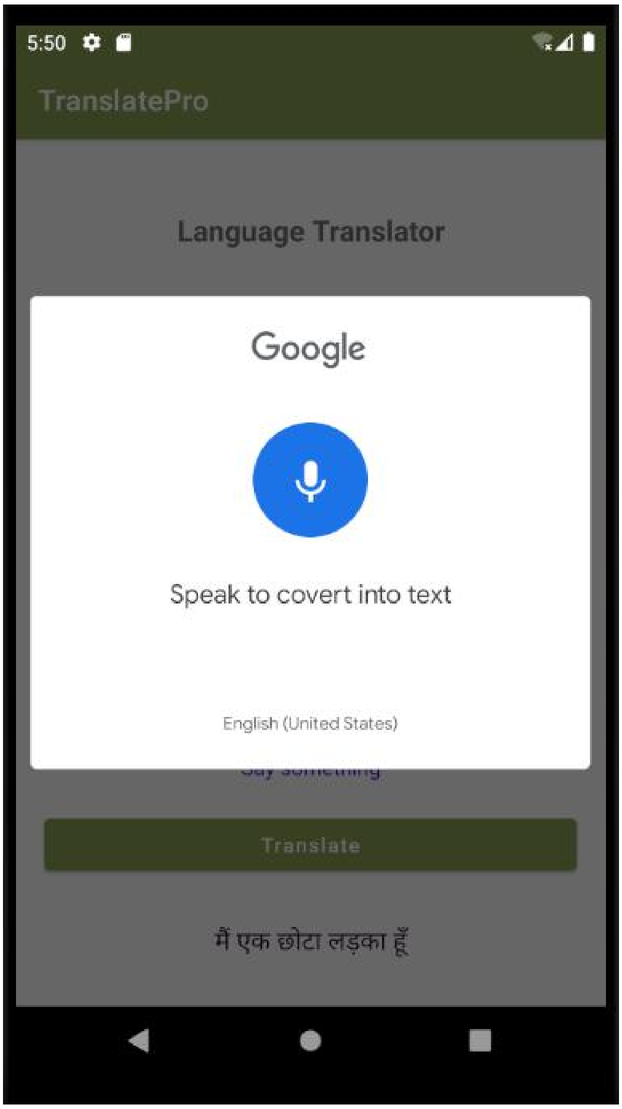
OR

Say something

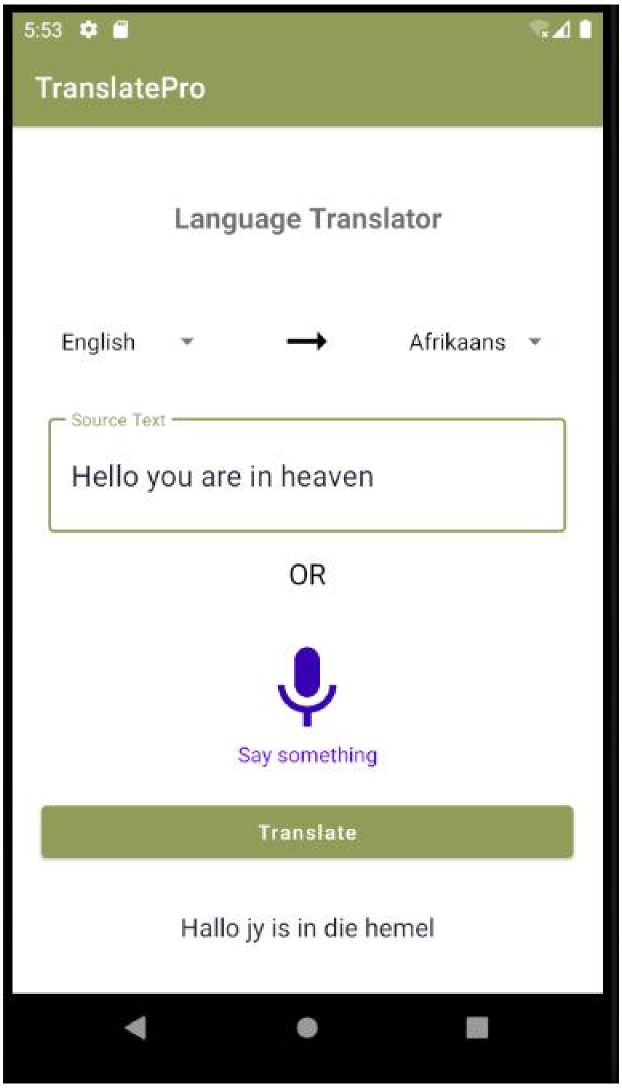
Translate

मैं एक छोटा लड़का हूँ

. Speech synthesis for translation



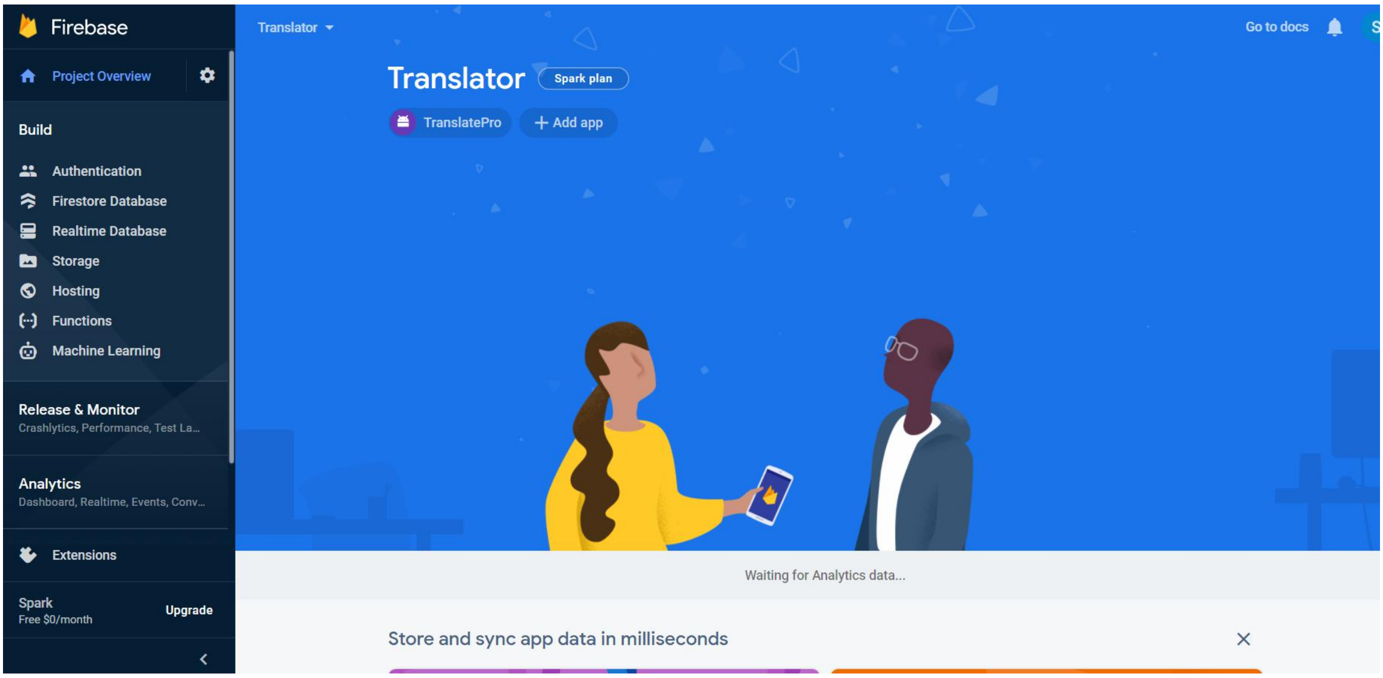
.Resulted text



Hello you are in heaven

Hallo jy is in die hemel

Firebase



CHAPTER - 6

TESTING

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing include designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques.

System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data.

In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. The Android framework includes an integrated testing framework that helps you test all aspects of your application and the SDK tools include tools for setting up and running test applications. Whether you are working in Eclipse with ADT or working from the command line, the SDK tools help you set up and run your tests within an emulator or the device you are targeting. There are different types of testing some of them are listed below:

6.1 Installation Testing:

There are two types of apps on an Android device i.e., Pre-installed applications and the applications which are installed later by the user.

For both of the above, installation testing is carried out by our teammates. It is ensuring smooth installation of the application without ending up in errors, partial installation etc.

6.2 Unit Testing

It focuses on smallest unit of software design. In this we test an individual unit or groups of inter related units. It is often done by programmer by using sample input and observing its corresponding outputs. In this testing technique we are primarily focuses on

- Loop methods and function is working fine or not.
- Misunderstood or incorrect Arithmetic precedence
- Incorrect Initialization

Unit Testing of the app:

Test cases	**Description**	**Expected Outcome**	**Result**
1.	Home page	Should display splash screen with animated text	pass
2.	Primary level selection	Should display the languages	pass
3.	Target language selection	Should display the targeted languages	pass
4.	Text input	Should display the input text	pass
5.	The Final result	Should display the final result	pass
6.	Speech synthesis for translation	Should follow voice command	pass
7	Resulted text	Should display the resulted text	pass

6.3 User Testing

User testing is the process through which the interface and functions of a website, app, product, or service are tested by real users who perform specific tasks in realistic conditions. The purpose of this process is to evaluate the usability of that website or app and to decide whether the product is ready to be launched for real users.Android Application - translate plus . This app was tested by our team mates and friends who are using different mobile phones (and having different android version) also tested on different emulator to check its performance and it seems to be working fine and users of this app are satisfied with the facilities and performance of the app and like the way how the app is worked.

6.4 Performance Testing

In this type of testing we have checked the performances of our application under some peculiar conditions are checked. Those conditions include:

- Low memory in the device.
- The battery in extremely at a low level.
- Poor/Bad network reception.

Performance is basically tested from 2 ends, application end, and the application server end. Our app is also performing well in this phase of testing as well. And we are getting positive feedback from user of our app.

6.5 Compability Testing

This application was tested and used on different devices like LG G3, Google Nexus 4. The application worked fine and is stable. The application worked fine in portrait mode and there
isn't any problem with compatibility.

On all types of testing (that we have performed above) our performing well on our app i.e. Translate plus.

CHAPTER 7

CODE

MainActivity.java

```
package com.sssappdeveloper.myapplication;


import android.content.Intent;

import android.os.Bundle;

import android.speech.RecognizerIntent;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ImageView;

import android.widget.Spinner;
import android.widget.TextView;

import android.widget.Toast;



import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;



import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.material.button.MaterialButton;

import com.google.android.material.textfield.TextInputEditText;

import com.google.mlkit.common.model.DownloadConditions;

import com.google.mlkit.nl.translate.TranslateLanguage;

import com.google.mlkit.nl.translate.Translation;

import com.google.mlkit.nl.translate.Translator;

import com.google.mlkit.nl.translate.TranslatorOptions;



import java.util.ArrayList;

import java.util.Locale;



public class MainActivity extends AppCompatActivity {



    private Spinner fromSpinner, toSpinner;
    private TextInputEditText sourceEdt;
    private ImageView micTV;
    private MaterialButton translateBtn;
    private TextView translatedTV;
    String[] fromLanguages = {"From", "English", "Afrikaans", "Arabic", "Belarusian",
"Bengali", "Catalan", "Czech", "Welsh", "Hindi", "Urdu"};
    String[] toLanguages = {"to", "English", "Afrikaans", "Arabic", "Belarusian", "Bengali",
"Catalan", "Czech", "Welsh", "Hindi", "Urdu"};

    private static final int REQUEST_PERMISSION_CODE = 1;
    String languageCode, fromLanguageCode, tolanguageCode = "";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        fromSpinner = findViewById(R.id.idFromSpinner);
        toSpinner = findViewById(R.id.idToSpinner);
        sourceEdt = findViewById(R.id.idEdtSource);
        micTV = findViewById(R.id.idIVMic);
        translateBtn = findViewById(R.id.idBtnTranslate);
        translatedTV = findViewById(R.id.idTVTranslatedTV);
        fromSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener()
{
```

```
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long
id) {
                fromLanguageCode = getLanguageCode(fromLanguages[position]);
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {

            }
        });
        ArrayAdapter fromAdapter = new ArrayAdapter(this,
R.layout.support_simple_spinner_dropdown_item, fromLanguages);

fromAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_ite
m);
        fromSpinner.setAdapter(fromAdapter);

        toSpinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long
id) {
                tolanguageCode = getLanguageCode(toLanguages[position]);
            }
```

```
      @Override
      public void onNothingSelected(AdapterView<?> parent) {

      }
    });

    ArrayAdapter toAdapter = new ArrayAdapter(this,
R.layout.support_simple_spinner_dropdown_item, toLanguages);

toAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    toSpinner.setAdapter(toAdapter);

    translateBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        translatedTV.setText("");
        if (sourceEdt.getText().toString().isEmpty()) {
          Toast.makeText(MainActivity.this, "Please Enter Your text to translate",
Toast.LENGTH_SHORT).show();
        } else if (fromLanguageCode == "") {
          Toast.makeText(MainActivity.this, "Please select source language",
Toast.LENGTH_SHORT).show();
        } else if (tolanguageCode == "") {
```

```
                    Toast.makeText(MainActivity.this, "Please select the lanaguge to make
translation", Toast.LENGTH_SHORT).show();
                } else {
                    translateText(fromLanguageCode, tolanguageCode,
sourceEdt.getText().toString());
                }
            }
        });

        micTV.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
                i.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
                i.putExtra(RecognizerIntent.EXTRA_LANGUAGE, Locale.getDefault());
                i.putExtra(RecognizerIntent.EXTRA_PROMPT, "Speak to covert into text");
                try {
                    startActivityForResult(i, REQUEST_PERMISSION_CODE);
                } catch (Exception e) {
                    e.printStackTrace();
                    Toast.makeText(MainActivity.this, "", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }


    protected void onActivitiyResult(int requestCode, int resultCode, @Nullable Intent data)
{
        super.onActivityResult(requestCode, resultCode, data);
        if (resultCode == REQUEST_PERMISSION_CODE) {
            if (resultCode == RESULT_OK && data != null) {
                ArrayList<String> result =
data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);
                sourceEdt.setText(result.get(0));
            }
        }
    }

    private void translateText(String fromLanguageCode, String tolanguageCode, String
source) {
        TranslatorOptions options =
                new TranslatorOptions.Builder()
                        .setSourceLanguage(fromLanguageCode)
                        .setTargetLanguage(tolanguageCode)
                        .build();
        final Translator MTranslator =
        Translation.getClient(options);

    DownloadConditions conditions = new DownloadConditions.Builder().build();
    MTranslator.downloadModelIfNeeded(conditions)
        .addOnSuccessListener(
            new OnSuccessListener<Void>() {
              @Override
              public void onSuccess(Void v) {
                translatedTV.setText("Translating..");
                MTranslator.translate(source).addOnSuccessListener(new
OnSuccessListener<String>(){
                  @Override
                  public void onSuccess(String s){
                    translatedTV.setText(s);
                  }
                }).addOnFailureListener(new OnFailureListener() {
                  @Override
                  public void onFailure(@NonNull Exception e) {
                    Toast.makeText(MainActivity.this, "Failed to
translate"+e.getMessage(), Toast.LENGTH_SHORT).show();
                  }
                });
              }
            })
```

```
                .addOnFailureListener(
                        new OnFailureListener() {
                            @Override
                            public void onFailure(@NonNull Exception e) {
                                Toast.makeText(MainActivity.this, "Fail to download language
modal"+e.getMessage(), Toast.LENGTH_SHORT).show();
                            }
                        });

        MTranslator.translate(source)
                .addOnSuccessListener(
                        new OnSuccessListener<String>() {
                            @Override
                            public void onSuccess(@NonNull String translatedText) {
                                translatedTV.setText(translatedText);
                            }
                        })
                .addOnFailureListener(
                        new OnFailureListener() {
                            @Override
                            public void onFailure(@NonNull Exception e) {
                                Toast.makeText(MainActivity.this, "Failed to Translate",
Toast.LENGTH_SHORT).show();
                            }
```

```
                    });
}

public String getLanguageCode(String language) {
    String languageCode = "";
    switch (language) {
        case "English":
            languageCode = TranslateLanguage.ENGLISH;
        break;
        case "Afrikaans":
            languageCode = TranslateLanguage.AFRIKAANS;
            break;
        case "Arabic":
            languageCode = TranslateLanguage.ARABIC;
            break;
        case "Belarusian":
            languageCode = TranslateLanguage.BELARUSIAN;
            break;
        case "Bengali":
            languageCode = TranslateLanguage.BENGALI;
            break;
        case "Catalan":
            languageCode = TranslateLanguage.CATALAN;
            break;
        case "Czech":
            languageCode = TranslateLanguage.CZECH;
            break;
        case "Welsh":
            languageCode = TranslateLanguage.WELSH;
            break;
        case "Hindi":
            languageCode = TranslateLanguage.HINDI;
            break;
        case "Urdu":
            languageCode = TranslateLanguage.URDU;
            break;
        default:
            languageCode="";

    }
  return languageCode;
}
```

ExampleInstrumentedTest.java

```
package com.sssappdeveloper.myapplication;


import android.content.Context;
import androidx.test.platform.app.InstrumentationRegistry;
import androidx.test.ext.junit.runners.AndroidJUnit4;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.*;

/**
 * Instrumented test, which will execute on an Android device.
 *
 * @see <a href="http://d.android.com/tools/testing">Testing documentation</a>
 */
@RunWith(AndroidJUnit4.class)
public class ExampleInstrumentedTest {
    @Test
    public void useAppContext() {
        // Context of the app under test.
        Context                              appContext                              =
InstrumentationRegistry.getInstrumentation().getTargetContext();
        assertEquals("com.sssappdeveloper.myapplication", appContext.getPackageName());
    }
}
```

ic_launcher_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="108dp"
    android:height="108dp"
    android:viewportWidth="108"
    android:viewportHeight="108">
    <path
        android:fillColor="#3DDC84"
        android:pathData="M0,0h108v108h-108z" />
    <path
        android:fillColor="#00000000"
        android:pathData="M9,0L9,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M19,0L19,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
    <path
        android:fillColor="#00000000"
        android:pathData="M29,0L29,108"
        android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M39,0L39,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M49,0L49,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M59,0L59,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M69,0L69,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M79,0L79,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M89,0L89,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M99,0L99,108"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,9L108,9"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,19L108,19"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,29L108,29"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,39L108,39"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,49L108,49"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,59L108,59"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,69L108,69"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
```

```
    android:fillColor="#00000000"
    android:pathData="M0,79L108,79"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,89L108,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M0,99L108,99"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,29L89,29"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,39L89,39"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
```

```
<path
    android:fillColor="#00000000"
    android:pathData="M19,49L89,49"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,59L89,59"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,69L89,69"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M19,79L89,79"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M29,19L29,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M39,19L39,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M49,19L49,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M59,19L59,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M69,19L69,89"
    android:strokeWidth="0.8"
    android:strokeColor="#33FFFFFF" />
<path
    android:fillColor="#00000000"
    android:pathData="M79,19L79,89"
```

```
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
</vector>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    style="@style/AppTheme.MaterialTheme"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/white"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="86dp"
        android:layout_marginTop="20dp"
        android:gravity="center"
        android:padding="4dp"
        android:text="Language Translator"
        android:textAlignment="center"
    android:textColor="@color/common_google_signin_btn_text_light"
    android:textSize="20sp"
    android:textStyle="bold" />


<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="20dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="20dp"
    android:orientation="horizontal"
    android:weightSum="3">

    <Spinner
        android:id="@+id/idFromSpinner"
        android:layout_width="9dp"
        android:layout_height="60dp"
        android:layout_margin="3dp"
        android:layout_weight="1"
        android:gravity="center"
        android:padding="3dp" />

    <ImageView
```

```
        android:layout_width="0dp"
        android:layout_height="42dp"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:src="@drawable/ic_right_arrow"
        app:tint="@color/black" />

    <Spinner
        android:id="@+id/idToSpinner"
        android:layout_width="9dp"
        android:layout_height="60dp"
        android:layout_margin="3dp"
        android:layout_weight="1"
        android:gravity="center"
        android:padding="3dp" />
</LinearLayout>

<com.google.android.material.textfield.TextInputLayout
    style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"
    android:layout_width="match_parent"
    android:layout_height="100dp"
    android:layout_marginStart="20dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="20dp"
    android:hint="Source Text"
    android:padding="5dp"
    android:textColorHint="@color/purple_700"
    app:hintTextColor="@color/purple_700">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/idEdtSource"
        android:layout_width="match_parent"
        android:layout_height="80dp"
        android:ems="10"
        android:importantForAutofill="no"
        android:inputType="textImeMultiLine|textMultiLine"
        android:textColor="@color/black_shade_1"
        android:textColorHint="@color/black_shade_1"
        android:textSize="20sp" />
</com.google.android.material.textfield.TextInputLayout>

<TextView
    android:layout_width="match_parent"
    android:layout_height="49dp"
    android:layout_gravity="center"
    android:layout_marginTop="5dp"
    android:text="OR"
    android:textAlignment="center"
    android:textAllCaps="true"
    android:textColor="@color/black"
    android:textSize="20sp" />

<ImageView
    android:id="@+id/idIVMic"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"
    android:src="@drawable/ic_mic"
    app:tint="@color/design_default_color_primary_dark" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Say something"
    android:textAlignment="center"
    android:textColor="@color/design_default_color_primary"
    android:textSize="15sp" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/idBtnTranslate"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
        android:text="Translate"
        android:textAllCaps="false" />

    <TextView
        android:id="@+id/idTVTranslatedTV"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="30dp"
        android:text="Translated Text"
        android:textAlignment="center"
        android:textAllCaps="false"
        android:textColor="@color/black_shade_1"
        android:textSize="18sp" />


</LinearLayout>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources>
  <color name="purple_200">#8F9D58</color>
  <color name="purple_500">#8F9D58</color>
  <color name="purple_700">#8F9D58</color>
  <color name="teal_200">#8F9D58</color>
  <color name="teal_700">#FF018786</color>
  <color name="black">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
  <color name="black_shade_1">#292036</color>
  <color name="yellow">#ffa500</color>
</resour
```

themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style                                        name="Theme.LanguageTranslatorApp"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
    <!-- Primary brand color. -->
    <item name="colorPrimary">@color/purple_500</item>
    <item name="colorPrimaryVariant">@color/purple_700</item>
    <item name="colorOnPrimary">@color/white</item>
    <!-- Secondary brand color. -->
    <item name="colorSecondary">@color/teal_200</item>
    <item name="colorSecondaryVariant">@color/teal_700</item>
    <item name="colorOnSecondary">@color/black</item>
```

```
        <!-- Status bar color. -->
        <item name="android:statusBarColor"
tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>
    <style name="AppTheme.MaterialTheme"
parent="Theme.MaterialComponents.Light.DarkActionBar">
        <item name="windowNoTitle">true</item>
        <item name="windowActionBar">false</item>
        <item name="colorPrimary">@color/purple_200</item>
        <item name="colorPrimaryDark">@color/purple_200</item>
        <item name="colorAccent">@color/white</item>
        <item name="android:singleLine">true</item>
    </style>
  </resources>
```

google-services.json

```
{
  "project_info": {
    "project_number": "577691258299",
    "project_id": "translator-e8a85",
    "storage_bucket": "translator-e8a85.appspot.com"
  },
  "client": [
{
  "client_info": {
    "mobilesdk_app_id": "1:577691258299:android:7509cc37711fc5ad04c89e",
    "android_client_info": {
      "package_name": "com.sssappdeveloper.myapplication"
    }
  },
  "oauth_client": [
    {
      "client_id": "577691258299-n9n33ug73r2gfo6g85oo66b31uiaead5.apps.googleusercontent.com",
      "client_type": 3
    }
  ],
  "api_key": [
    {
      "current_key": "AIzaSyDeHkpkh7jIQJPwxe-a7sdeYFRH5y9uvUo"
    }
  ],
  "services": {
    "appinvite_service": {
      "other_platform_oauth_client": [
        {
              "client_id":                                         "577691258299-
n9n33ug73r2gfo6g85oo66b31uiaead5.apps.googleusercontent.com",
              "client_type": 3
            }
          ]
        }
      }
    }
  ],
  "configuration_version": "1"
}
```

CHAPTER 8

CONCLUSION

In conclusion, translation apps are ruining the chances of learners to learn new languages.Even though the translation app saves time and resources, they are not accurate. The integrity of information is paramount, and there is no point in translating information if it is not precise.

TRANSLATING APPS6Leaners should not use translation apps as it prevents mastering of the new language. Wrong syntax, word, grammar, and sentences are among the major challenges that students face when using a translation app in a bid to learn a new language. The translation app has meticulous codesthat incorporate rules that govern different languages. However, since language differs in their use of signs, syntax errors are commonplace while using a translation app. It is also correct to assert that the translation app lacks some words in its database. This results in the translation app generating a word that is meaningless. The use of such words complicates learning of a new language as the learner tries to memorize and apply a word that was used in the wrong context; however, the learner has no idea. It is all correct that since the different word means different things in a different language, translation apps tamper with the integrity of sentences as they just replace a word for the other paying little attention to context. Therefore, a word is replaced by a different word that has a different meaning in another language, which changes the i0ntended message. Translational apps are commonplace for grammar errors due to the inability to synchronize grammar rules between different languages. A learner who has no knowledge regarding a new language will not learn much while relying on a translation app because the grammar of the translation app is not accurate. After an in-depth review of the literature regarding the translation app, it is prudent to note that they hinder learning of a new language because they are not accurate, and the learner does not know that. Therefore, the correct part of the app cannot help in mastering a new language. The learning institutions should discourage foreign students from using a translation app when learning a new language.

CHAPTER 8

REFERENCES

1. Introduction to Android:
http://developer.android.com/guide/index.html

2. ***Android API:***
http://developer.android.com/reference/packages.html

3. *Android User Interfaces:*
http://developer.android.com/guide/topics/ui/index.html

4.Layout:

http://developer.android.com/guide/topics/ui/declaring-layout.html

5. Android developer Guide:
https://developer.android.com/

6. For rectifying the error :
https://stackoverflow.com/

7. Firebase:
https://firebase.google.com/

8.developers google:
https://developers.google.com/