PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PRN No.: Roll no: 156

1. Name of the Student
Aamil Arif Nakhdua

2. Title of the Project
Smart Lens

3. Name of the Guide
Priya Singh

4. Teaching experience of the Guide

5. Is this your first submission? Yes ☐ No ☐

Signature of the Student Signature of the Guide

Date: …… Date: …….

Signature of the Coordinator Date: ……

SMART LENS

A Project Report
Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Aamil Arif Nakhuda

Under the esteemed guidance of
Priya Singh
Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
SHANKAR NARAYAN COLLEGE OF ARTS AND COMMERCE
(Affiliated to University of Mumbai)
MUMBAI, 401107
MAHARASHTRA
2021-22

Shree Shankar Narayan Education Trust's

SHANKAR NARAYAN COLLEGE OF ARTS & COMMERCE

B.M.S., B.Sc.I.T., B.C.S., B.B.I., B.A.F., B.F.M., M.Sc.I.T., M.Com.

Navghar, Mahavidyalaya Marg, Bhayandar (E), Thane - 401 105. (Maharashtra State),
(Affiliated to the University of Mumbai)

NAAC Accredited 'A'

Prin. Dr. V. N. Yadav M.Com., M.Phil, Ph.D.
Website - www.sncollege.com

T-35

Tel. : 2804 65 64, 2804 82 35
E-mail : info@sncollege.com • Fax : 2804 0966

Ref. No.

Date _____

CERTIFICATE

This is to Certify that Mr./Miss _____

_____ *Studying in T.Y.* _____

(Semester) in our college having Roll No. _____ *and Exam*

Seat No. _____ *has successfully completed the project on*

_____ *in the academic Year 20 - 20*

Date : / /

Project Guide *Co-ordinator* *Principal*

Internal Examiner *External Examiner*

SHANKAR NARAYAN COLLEGE OF ARTS & COMMERCE
(Affiliated to University of Mumbai)
MUMBAI-MAHARASHTRA-401107

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **"Smart Lens"**, is bonafide work of **AAMIL ARIF NAKHUDA** bearing Seat.No: submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Priya Singh **Coordinator**

(Don't write names of lecturers or HOD)

External Examiner

Date: **College Seal**

ABSTRACT

We do get curious about something and want to search and learn about it, but we don't know how to frame those things in words in order to search, also when we travel and come across an important foreign language document and we cant read it nor understand it so it gets important for these things to be searched via images and translated, as well as spoken in order for us to understand these things. There are times when code examples are printed on textbooks and you wonder whether it works or gives errors, with just a click of a photo.

The purpose of this project is to save time and search for those things that we can’t express in words and search it via images. Also scan the text and display, detect as well as translate it.

This Android Application contains a lot of user-friendly features. Also the app’s main motive and feature is the image, text and code search. The Object Recognition brings a bunch of results which contains a lot of information on which clicking will redirect to the website to read more about it. The Text Recognition will scan the text in almost any language and display it to the user where they can copy, share, detect and translate to the desired language they want. The Code/Word Recognition will scan the code and then compile/interpret the scanned code and also provide the output for it. The following will have support for multiple languages with different text editors and IDE.

ACKNOWLEDGEMENT

Achievement is finding out what you would be doing rather than what you have to do. It is not until you undertake such a project that you realize how much effort and hard work it really is, what are your capabilities and how well you can present yourself or other things. It tells us how much we rely on the efforts and goodwill of others. It gives me immense pleasure to present this report towards the fulfillment of my project.

It has been rightly said that we are built on the shoulders of others. For everything I have achieved, the credit goes to all those who have helped me to complete this project successfully. I take this opportunity to express my profound gratitude to management for giving me this opportunity to accomplish this project work.

I am very much thankful to Dr. V. N. Yadav - Principal of Shankar Narayan College for their kind cooperation in the completion of my project.

A special vote of thanks to my faculty Prof. Smita Dalvi and, Prof. Vaishali Kadam who is our HOD & also our project guide Prof. Project Guide for their most sincere, useful and encouraging contribution throughout the project span ,without them we couldn’t start and complete the project on time.

Finally, last but not the least I would like to thank my parents & entire Information Technology (IT) department who directly or indirectly helped me in completion of this project & to my entire family without whose support, motivation & encouragement this would not have been possible.

DECLARATION

I hereby declare that the project entitled, "**Smart Lens**" done at the place **Mira Bhayandar (East)**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

Name and Signature of the Student
Aamil A. Nakhuda

TABLE OF CONTENTS

Chapter 1

Introduction

"Smart Lens" is an Android application, developed to save time and lots of effort taken in order to search for Images, Text, Translated text, Code with output. This app is targeted to all the age grouped users no matter what's their age as the UI and the UX will be simple, interactive and most importantly engaging.

The Android app is developed using Java as the programming language and two Google Services which are Firebase ML Kit and Google Search API. It is very simple and user friendly and just asks the user to follow some steps. It is very fast because the app follows the native approach.

The main features of the app are to search objects in day to day life with just a click of photo, word search with language detection and translation in desired languages with help of just a click of the photo. Also an important feature for teachers, students and software developers is to scan the code from any document or other places and detects the programming language as well as provide the output from the scanned image using third party online compilers/interpreters and also famous text editors and IDE's are included.

1.1 Background

In the beginning, to find an image, you would come up with some words to describe what you wanted to see. For example, "top of Mount Everest." Then, you'd type those words into search. The search engine would find a matching image by scanning the text of pages that images were on image titles, descriptions, and any accompanying text—and return something that appeared to match your search, based on its textual context.

This started in the late ’90s, when AltaVista launched an image feature for its search that allowed you to put in a text term and return images. This was a proximate way to get people to the pictures they desired. Early image searches like AltaVista relied not on the ability to process the image itself and match that to text, but on image descriptions or corresponding text on web pages. Google Images, which followed in 2001, quickly amassed a huge store of pictures, 250 million within the year. It was developed because people wanted to get not to approximate images based on the context of web pages but to the exact image that they had in mind.

Technically accurate, but not JLo. If you searched “green dress,” you would have gotten green dresses back, because those writing the descriptions were writing accurate descriptions of the contents.

But users had to search by proxy and hope for the best instead of getting an immediate payoff of what they could picture in their mind.The next step in developing a better, more intuitive image search was CBIR, or content-based image retrieval. CBIR analyzes the contents of the image rather than just scanning for keywords around it. Things like color and shape can be directly connected with an image without the need for any text clarification. Then came the most effective form of CBIR to date: “reverse” image search, where the input from the user was not text, but an image itself. This obviously removed a layer of guesswork from search queries, because it didn’t rely on a user to describe the image they were thinking of with text. Instead, they were using the image itself.

The common theme throughout every iteration of image search thus far has been that on one or both ends of the search, we’ve been relying on humans *writing* descriptions of images. Even when search started to get into pairing similar images, it was either limited to finding the source of an image or finding similar images to one that had been searched for with a string of text.

1.2 Objective

The main objective of Smart Lens app is to search for objects, text and code through the method of Image which is very quick and a large number of other features as well in a user friendly, interactive and engaging manner. Other objectives include to provide a simple and easily understandable User Interface and an exceptional and interactive User Experience to all age grouped users. There are mainly four features with an objective which are as follows:-

Object Recognition and Search through Image The aim of this is to provide information of day to day or unknown objects through a search of image easily.

Word Recognition in multiple languages through Image with language detection and translation to desired language. The aim of this is to provide easy search of words through images, also when visiting a foreign country we can search their local language, detect and translate into our desired language with just a click of a picture.

Live Word Recognition in multiple languages through continuous Image with language detection and translation to desired language. The aim is to provide similar features like the above but with a live word search.

Code/Program Word Recognition through Image with computer language detection and output for it through third party online compiler/interpreter with well known text editor and IDE's. The aim is to provide a code search with output to teachers, students and software developers with just a click of a photo.

1.3 Purpose, Scope and Applicability

1.3.1 Purpose

This project is being done because today in this 21st century in which we live, people usually don't have time for searching something in a quick way. The purpose of this project is to save time and efforts that are made in order to search and provide information of day to day or

even unknown objects, easy search of words when visiting a foreign country, search their local language, detect and translate into our desired language with just a click of a picture through image recognition. Also this includes live word search which enables more quick and fast search and not even a need to click a picture. Teachers, students and software developers do need a faster method to run or debug a code or even check whether it works or if it has a bunch of errors without doing much of hardwork and spending more time typing it. To solve this problem and overcome it, my app has code word search, computer language detection and output also with famous text editors and IDE's with this purpose with just a click of a photo.

1.3.2 Scope

The main issue that has been covered in this project is how to save time without much effort? Normally, we require a lot of time recognizing and searching for objects both which we use in daily life as well as which are unknown. Also we need to write the entire code just to check whether it works or not which is really time taking. The following will be the main features of the app and the main issues that are been covered in this project which are follows:-

Object Recognition and Search through Image. One of the main issue that is been covered is to recognize and search day to day or unknown objects which we want to know about it but cannot search about it just because we don't know what it is been said for example an **Aglet** which is a metal or plastic tube fixed tightly round each end of a shoelace. Also to do so we need to give some time and effort in order to do this, but in this app it can be done in just a click of a photo, which saves quite an amount of time.

Word Recognition in multiple languages through Image with language detection and translation to desired language. This is also one of the main issues that has been covered is to recognize text/words in multiple languages, detect the language of the recognized text/word and also translate into our desired language. This can be highly useful when we are travelling to a country or a city where we are very unfamiliar or unknown to the local language, in such cases we do many times come across where we are in a heavy need of a translator that can help to break the

linguistic barrier and help understanding it. My app recognizes the text in multiple languages and also detect which language it is, this is really helpful to distinguish between similar looking languages like Chinese, Japanese, Korean and Hindi, Marathi. Also we can convert the recognized language to our local or desired language.

Live Word Recognition in multiple languages through continuous Image with language detection and translation to desired language. There are main issues where text/words are changing like the ones in news channels where the text/words keeps changing, it is known as **news ticker** (sometimes called a "**crawler**", "**crawl**", "**slide**", or "**zipper**"). It is a horizontal or vertical (depending on a language's writing system) text-based display either in the form of a graphic that typically resides in the lower third of the screen space on a television station or network (usually during news programming) or as a long, thin scoreboard-style display seen around the facades of some offices or public buildings dedicated to presenting headlines or minor pieces of news. If the news ticker is changing and is in another language it becomes difficult to search it even via photo, also detect and translate into other languages. My app recognizes the text in multiple languages and also detects which language it is without clicking the picture, this is really helpful to distinguish between similar looking languages like Chinese, Japanese, Korean and Hindi, Marathi. Also after the text selection we can convert the recognized language to our local or desired language.

Code/Program Word Recognition through Image with computer language detection and output for it through third party online compiler/interpreter with well known text editor and IDE's. One of the main issues for teachers, students and software developers is that they do need to write a lot of code just to check whether it works or it has bugs. Also many of the students' families could not afford a laptop or computer where they can code. My app has code/word recognition and computer language detection which helps in distinguishing between similar syntactic computer languages such as C, C++. Also my app will provide third party online compiler/interpreter with well known text editor and IDE's, so that we can easily check whether the code works or not in just a click of

photo and also who cannot afford high end laptop/computer for learning Android they can use IDE or any text editor in their mobile virtually.

1.3.3 Applicability

This Android App can work on Android devices with Android 6.0 Marshmallow and up. This app can be used basically by anyone in the world, no matter if it is a particular company, organization or a single user. This app is a perfect fit for travellers, students, teachers and software developers.

1.3.4 Achievements

I have gained a tremendous amount of knowledge and a great idea of how a software project is being developed and designed on a real industrial level. How much hard work, time, effort, planning, research and development is required in order to make a full fledged Android app which can help solve all the issues which are mentioned above.

Chapter 2

System Analysis

2.1 Existing System

2.1.1 Mobile Application

A mobile application, also referred to as a mobile app or simply an app, is a computer program or software application designed to run on a mobile device such as a phone, tablet, or watch. Mobile applications often stand in contrast to desktop applications which are designed to run on desktop computers, and web applications which run in mobile web browsers rather than directly on the mobile device.

Apps were originally intended for productivity assistance such as email, calendar, and contact databases, but the public demand for apps caused rapid expansion into other areas such as mobile games, factory automation, GPS and location-based services, order-tracking, and ticket purchases, so that there are now millions of apps available. Many apps require Internet access. Apps are generally downloaded from app stores, which are a type of digital distribution platform.

The term "app", short for "software application", has since become very popular; in 2010, it was listed as "Word of the Year" by the American Dialect Society.

Apps are broadly classified into three types: native apps, hybrid and web apps. Native applications are designed specifically for a mobile operating system, typically iOS or Android. Web apps are written in HTML5 or CSS and typically run through a browser. Hybrid apps are built using web technologies such as JavaScript, CSS, and HTML 5 and function like web apps disguised in a native container.

2.1.1.1 Android Applications

An Android app is a software application running on the Android platform. Because the Android platform is built for mobile devices, a typical Android app is designed for a smartphone or a tablet PC running on the Android OS. Android apps are written in the Java programming language and use Java core libraries. Developers may download the Android software development kit (SDK) from the Android website. The SDK includes tools, sample code and relevant documents for creating Android apps.

2.1.1.2 iOS Applications

iOS (formerly iPhone OS) is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that powers many of the company's mobile devices, including the iPhone and iPod Touch; the term also included the versions running on iPads until the name iPadOS was introduced with version 13 in 2019. The iOS SDK (Software Development Kit) allows for the development of mobile apps on iOS. While originally developing the iPhone prior to its unveiling in 2007, Apple's then-CEO Steve Jobs did not intend to let third-party developers build native apps for iOS, instead directing them to make web applications for the Safari web browser. The SDK was released on March 6, 2008 and is a free download for users of Mac personal computers. It is not available for Microsoft Windows PCs. The SDK contains sets giving developers access to various functions and services of iOS devices, such as hardware and software attributes. It also contains an iPhone simulator to mimic the look and feel of the device on the computer while developing.

2.1.2 Web Applications

A web application (or web app) is application software that runs on a web server, unlike computer-based software programs that are run locally on the operating system (OS) of the device. Web applications are accessed by the user through a web browser with an active network connection. These applications are programmed using a client–server modeled structure—the user ("client") is provided services through an off-site server that is hosted by a third-party. Writing web applications is often simplified by the use of web application frameworks. These frameworks facilitate rapid application development by allowing a development team to focus on

the parts of their application which are unique to their goals without having to resolve common development issues such as user management. Many of the frameworks in use are open-source software.

2.1.3 Desktop Applications

Desktop software apps are created to run on desktop computers (PCs) and laptops. Windows applications can be deployed as executable or as Windows Store apps. MacOS applications are usually deployed through the Apple Store and Linux apps are deployed as executables. There are many frameworks and languages used to create desktop applications. They can be categorized by the platform they support or as cross platform technologies along with the programming languages and libraries used. A native application is a software program that is developed for use on a particular platform (macOS - Objective C or Windows -.Net, C#) or device. Native apps are written in a code, preliminarily used for the target device and its OS. Cross-Platform application is an application that can be run with little or no modification on Windows, Macintosh, and Linux/Unix platforms without compromising functionality, usability, or quality.

2.2 Proposed System

2.2.1 Mobile Application

I have chosen Mobile Application as my project's system. And the reason is simply today, the number of mobile users is greater than the number of desktop users.With mobile apps, it's easy to treat users with a personalized experience. Using a mobile app A/B testing tool, you can also test out different experiences for your customers. Mobile apps have the advantage of utilizing features of a mobile device like a camera, contact list, GPS, phone calls, accelerometer, compass, etc. Such device features, when used within an app, can make the user experience interactive and fun.

Moreover, these features can also reduce the efforts users would have to make otherwise. For instance, users completing a form on a banking app might need to submit their photographs

to complete the process. The app can enable users to use their mobile camera to capture and submit photographs. The interrelated features significantly shorten the time taken to perform a particular task in an app and boost conversions. It is probably the most fundamental difference between a mobile website and an app.

Like a website, apps too might require internet connectivity to perform most tasks, but here's the difference: an app can still offer basic content and functionality to users in offline mode. Let's take the example of a banking app again. The app can provide features like tax calculation, installment calculation, and determination of loan limits. These features can work even without the help of an internet connection.

With all the technological advancements in web designing, mobile websites still have to rely on browsers to perform even the most elementary functions. Mobile websites depend on browser features like 'back button,' 'refresh button,' and 'address bar' to work. Mobile apps, however, don't have any of these restrictions. A mobile app gets designed with several elaborate functions based on advanced gestures like 'tap,' 'swipe,' 'drag,' 'pinch,' 'hold,' and more.

Since a mobile app is distinct from a company's website, it has the liberty of offering a new brand experience to users. It means that the company can experiment with new branding styles for the app, which can be different from the regular brand style of the company's website (or the company altogether). Going a step further, companies can build mobile apps specifically to transition into a new brand style for themselves. Additionally, a mobile app can also allow users to customize its appearance, as per users' liking. This can further help on the personalization front of the app.

Mobile users spend **88%** of their time on mobile apps and just **12%** of the time on mobile websites. A well-designed mobile app can perform actions much quicker than a mobile website. Apps usually store their data locally on mobile devices, unlike websites that generally use web servers. For this reason, data retrieval happens swiftly in mobile apps.

2.2.2 Android Platform

Android holds a massive global market share lead over iOS. iOS holds a slight market share lead over Android in the U.S. According to Statcounter, the global market share looks like this:
Android: 72.2% while iOS: 26.99%. However, within the U.S., that market share looks like this:

iOS: 59.17% while Android: 40.54%. How is it that Android can lead iOS globally by nearly 50% while trailing iOS in the U.S. by nearly 20%? Unless you'd been following the trend for years, you'd think that's an anomaly. It's not. The primary reason why Android rules the global market is cost. In many countries (with significantly higher populations than the U.S.), people have far less disposable income. Because of that, the first factor in deciding on a phone is cost. In that arena, Apple simply cannot compete with Google. It does go beyond cost. In fact, according to Mobileapps: Almost every age group prefers Android devices over iPhones.

The Google Play Store has more apps than the Apple counterpart.

Android has an increasingly larger market share in Asian countries.

Android is the most popular mobile OS in the world's most populous continent (with over 83.53%).

2.2.3 Front-End

The front end of the app can be designed or let's say the layout of the app can be developed by XML.XML stands for Extensible Markup language.The declaration of a layout are in two ways:

Declare UI elements in XML. Android provides a straightforward XML vocabulary that corresponds to the View classes and subclasses, such as those for widgets and layouts.

We can also use Android Studio's Layout Editor to build your XML layout using a drag-and-drop interface.

We can instantiate layout elements at runtime. Our app can create View and ViewGroup objects (and manipulate their properties) programmatically.

Declaring the UI in XML allows you to separate the presentation of your app from the code that controls its behavior. Using XML files also makes it easy to provide different layouts for different screen sizes and orientations. The Android framework gives you the flexibility to use either or both of these methods to build your app's UI. For example, you can declare your app's default layouts in XML, and then modify the layout at runtime.

Using Android's XML vocabulary, we can quickly design UI layouts and the screen elements they contain, in the same way we can create web pages in HTML — with a series of nested elements.

Each layout file must contain exactly one root element, which must be a View or ViewGroup object. Once you've defined the root element, you can add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines your layout. After you've declared your layout in XML, save the file with the .xml extension, in your Android project's res/layout/ directory, so it will properly compile.

2.2.4 Back-End

2.2.4.1 Java

Java is one of the powerful general-purpose programming languages, created in 1995 by Sun Microsystems (now owned by Oracle). Java is Object-Oriented. However, it is not considered as pure object-oriented as it provides support for primitive data types (like int, char, etc). Java syntax is similar to C/C++. But Java does not provide low-level programming functionalities like pointers. Also, Java code is always written in the form of classes and objects. Android heavily relies on the Java programming language and all the SDKs required to build for android applications use the standard libraries of Java. If one is coming from a traditional programming background like C, C++, Java is easy to learn.

Java has platform independent features so it is used for android development. Java is a very popular language due to its awesome features and performance. The community of Developers that have proficiency is really big. Thus android developers choose java as there is already a good base of java programmers available that can help in creating, improving android applications plus with many libraries and tools of java make developers life easier. Large java developer base enables developers to develop a lot of android apps fast so it is based on java.

Developers that do not use Java have to deal with serious problems like memory leads and bad pointer usage. Sometimes these problems harm at the highest level like the crash of an application or crash of the OS. Android easily implements and fixes common problems with other programming languages with the help of Java. These are some problems that never occur

when you program with java. Java is machine independent and runs only in JVM space so it protects you from these problems.

Java is the best and the first language for native software development. That is really the only option for native applications, native applications are the heart of android. Java's important core features also motivated android developers to use it for development, these features include easy learning, platform-independent, secure, object-orientation.

2.2.4.2 Firebase ML Kit

ML Kit is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use package. Whether you're new or experienced in machine learning, you can implement the functionality you need in just a few lines of code. There's no need to have deep knowledge of neural networks or model optimization to get started. On the other hand, if you are an experienced ML developer, ML Kit provides convenient APIs that help you use your custom TensorFlow Lite models in your mobile apps.

ML Kit makes it easy to apply ML techniques in your apps by bringing Google's ML technologies, such as the Google Cloud Vision API, TensorFlow Lite, and the Android Neural Networks API together in a single SDK. Whether you need the power of cloud-based processing, the real-time capabilities of mobile-optimized on-device models, or the flexibility of custom TensorFlow Lite models, ML Kit makes it possible with just a few lines of code.

ML Kit SDK is a rather new product from Google that was presented in 2018. ML Kit is a software development kit that makes it possible for developers to simplify the integration of machine learning models into their mobile apps. Moreover, even a junior developer can deal with this task.

ML Kit is still available in the beta version only. It can be launched using Firebase. At the moment, there are 11 features with APIs available. By the way, only 5 features were available on Google I/O presentation in 2018, and today, this number is doubled. Features are the following:

Text recognition

Face detection

Barcode scanning

Image labeling

Object detection & tracking

Landmark recognition

Language identification

Translation

Smart reply

AutoML model inference

Custom model inference

ML Kit SDK allows developers to make everything faster and easier. Developers need to pass data to the required API and wait for the response from SDK. Google's representatives assure that developers don't need to be highly skilled in neural networks to implement their APIs. Everything developers need is to add several lines of codes, and a new app will get cool AI-based features

2.2.4.3 Serp API: Google Search API

SerpApi is a real-time API to access Google search results. They handle proxies, solve captchas, and parse all rich structured data for you. SERP stands for Search Engine Results Pages. SERP refers to the pages returned on the search engine after typing a keyword on the search bar. The APIs refer to software applications that help you scrape search results from the search engine in real-time, running on a specific programming language to make requests and responses in a given format. These APIs allow you to analyze, review, track, and improve your website's visibility on the search engine.

For an example: Whenever you search for something on Google (or other search engines), you get a page full of helpful information related to your query. If you're searching for cat photos, you'll get cat photos and relevant articles.

Google SERP has evolved recently. You get the results on the Google search page. This means you don't need to visit another site to find out what you are looking for.

Chapter 3

Requirements and Analysis

3.1 Problem Definition

The main issue that has been covered in this project is how to save time without much effort? Normally, we require a lot of time recognizing and searching for objects both which we use in daily life as well as which are unknown. Also we need to write the entire code just to check whether it works or not which is really time taking. The following will be the main features of the app and the main issues that are been covered in this project which are follows:-

Object Recognition and Search through Image. One of the main issue that is been covered is to recognize and search day to day or unknown objects which we want to know about it but cannot search about it just because we don't know what it is been said for example an **Aglet** which is a metal or plastic tube fixed tightly round each end of a shoelace. Also to do so we need to give some time and effort in order to do this, but in this app it can be done in just a click of a photo, which saves quite an amount of time.

Word Recognition in multiple languages through Image with language detection and translation to desired language. This is also one of the main issues that has been covered is to recognize text/words in multiple languages, detect the language of the recognized text/word and also translate into our desired language. This can be highly useful when we are travelling to a country or a city where we are very unfamiliar or unknown to the local language, in such cases we do many times come across where we are in a heavy need of a translator that can help to break the linguistic barrier and help understanding it. My app recognizes the text in multiple languages and also detect which language it is, this is really helpful to distinguish between similar looking languages like Chinese, Japanese, Korean and Hindi,

Marathi. Also we can convert the recognized language to our local or desired language.

Live Word Recognition in multiple languages through continuous Image with language detection and translation to desired language. There are main issues where text/words are changing like the ones in news channels where the text/words keeps changing, it is known as **news ticker** (sometimes called a "**crawler**", "**crawl**", "**slide**", or "**zipper**"). It is a horizontal or vertical (depending on a language's writing system) text-based display either in the form of a graphic that typically resides in the lower third of the screen space on a television station or network (usually during news programming) or as a long, thin scoreboard-style display seen around the facades of some offices or public buildings dedicated to presenting headlines or minor pieces of news. If the news ticker is changing and is in another language it becomes difficult to search it even via photo, also detect and translate into other languages. My app recognizes the text in multiple languages and also detects which language it is without clicking the picture, this is really helpful to distinguish between similar looking languages like Chinese, Japanese, Korean and Hindi, Marathi. Also after the text selection we can convert the recognized language to our local or desired language.

Code/Program Word Recognition through Image with computer language detection and output for it through third party online compiler/interpreter with well known text editor and IDE's. One of the main issues for teachers, students and software developers is that they do need to write a lot of code just to check whether it works or it has bugs. Also many of the students' families could not afford a laptop or computer where they can code. My app has code/word recognition and computer language detection which helps in distinguishing between similar syntactic computer languages such as C, C++. Also my app will provide third party online compiler/interpreter with well known text editor and IDE's, so that we can easily check whether the code works or not in just a click of photo and also who cannot afford high end laptop/computer for learning Android they can use IDE or any text editor in their mobile virtually.

3.2 Requirement Specification

1. Android SDK

The Android SDK is a software development kit that includes a comprehensive set of development tools. These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Android applications.

Until around the end of 2014, the officially-supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin. As of 2015, Android Studio] is the official IDE; however, developers are free to use others, but Google made it clear that ADT was officially deprecated since the end of 2015 to focus on Android Studio as the official Android IDE. Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).

2. Java JDK

The Java Development Kit (JDK) is a distribution of Java Technology by Oracle Corporation. It implements the Java Language Specification (JLS) and the Java Virtual Machine Specification (JVMS) and provides the Standard Edition (SE) of the Java Application Programming Interface (API). It is a derivative of the community driven OpenJDK which Oracle steward. It provides software for working with Java applications. Examples of included software are the virtual machine, a compiler, performance monitoring tools, a debugger, and other utilities that Oracle considers useful for a Java programmer.

Oracle has released the current version of the software under the Oracle No-Fee Terms and Conditions (NFTC) license. Oracle releases binaries for the x86-64 architecture for Windows, macOS, and Linux based operating systems, and for the aarch64 architecture for

macOS and Linux. Previous versions have supported the Oracle Solaris operating system and SPARC architecture.Oracle's primary implementation of the JVMS is known as the HotSpot (virtual machine).

3.3 Planning and Scheduling

Gantt chart and pert chart

3.4 Software and Hardware Requirements

3.4.1 Software Requirements

Android Studio
Visual Studio Code extension - Thunder Client
Firebase ML Kit account
SerpAPI account

3.4.2 Hardware Requirements

Operating System:- 64-bit Microsoft® Windows® 8/10
Processor:- x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor
RAM:- 8 GB RAM or more
ROM:- 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
Screen Size:- 1280 x 800 minimum screen resolution

3.5 Conceptual Models

Data Dictionary

Element or value display name	Description	Data type	Character length	Acceptable values	Required?	Acceptable null value?
Image	This is the Image that is given by the user to recognize object, word and others	Bitmap Image	640x480/720x1280	Jpg, jpeg	yes	no

Fig 3.5.1. Data Dictionary

A data dictionary, or metadata repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format". Oracle defines it as a collection of tables with metadata. The term can have one of several closely related meanings pertaining to databases and database management systems (DBMS):

A document describing a database or collection of databases

An integral component of a DBMS that is required to determine its structure

A piece of middleware that extends or supplants the native data dictionary of a DBMS

E-R Diagrams

No E-R Diagrams as my project does not have any database as a backend.

Data Flow Diagrams / UML

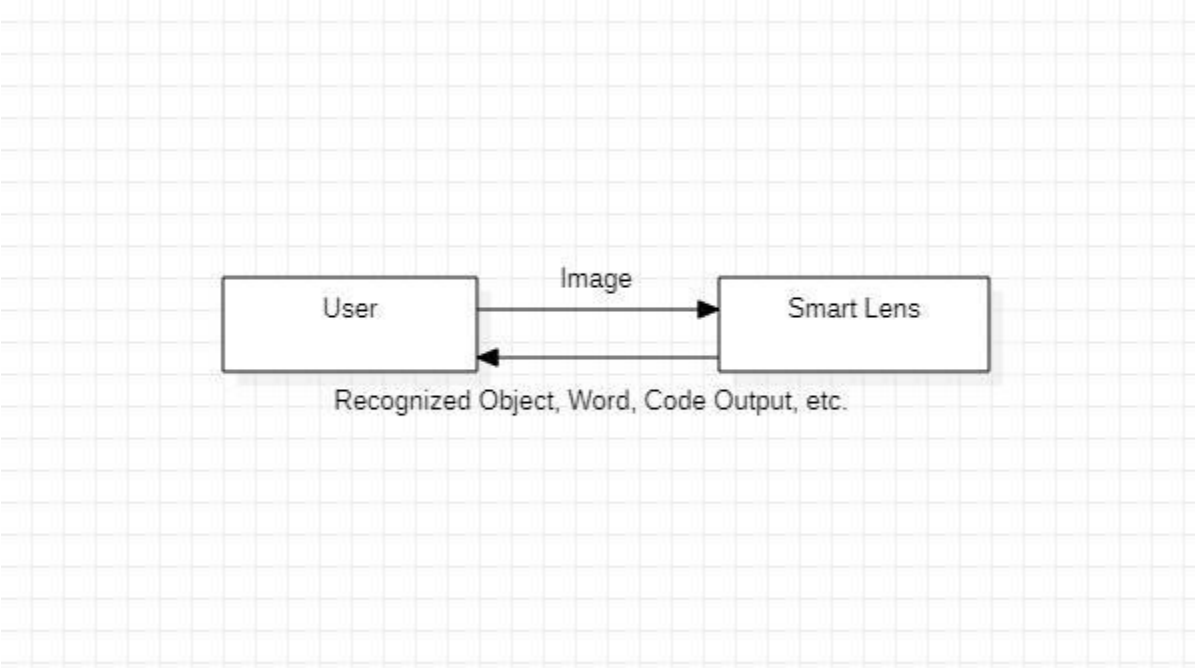


Fig. 3.5.2: Zeroth Level Data Flow Diagram of Smart Lens

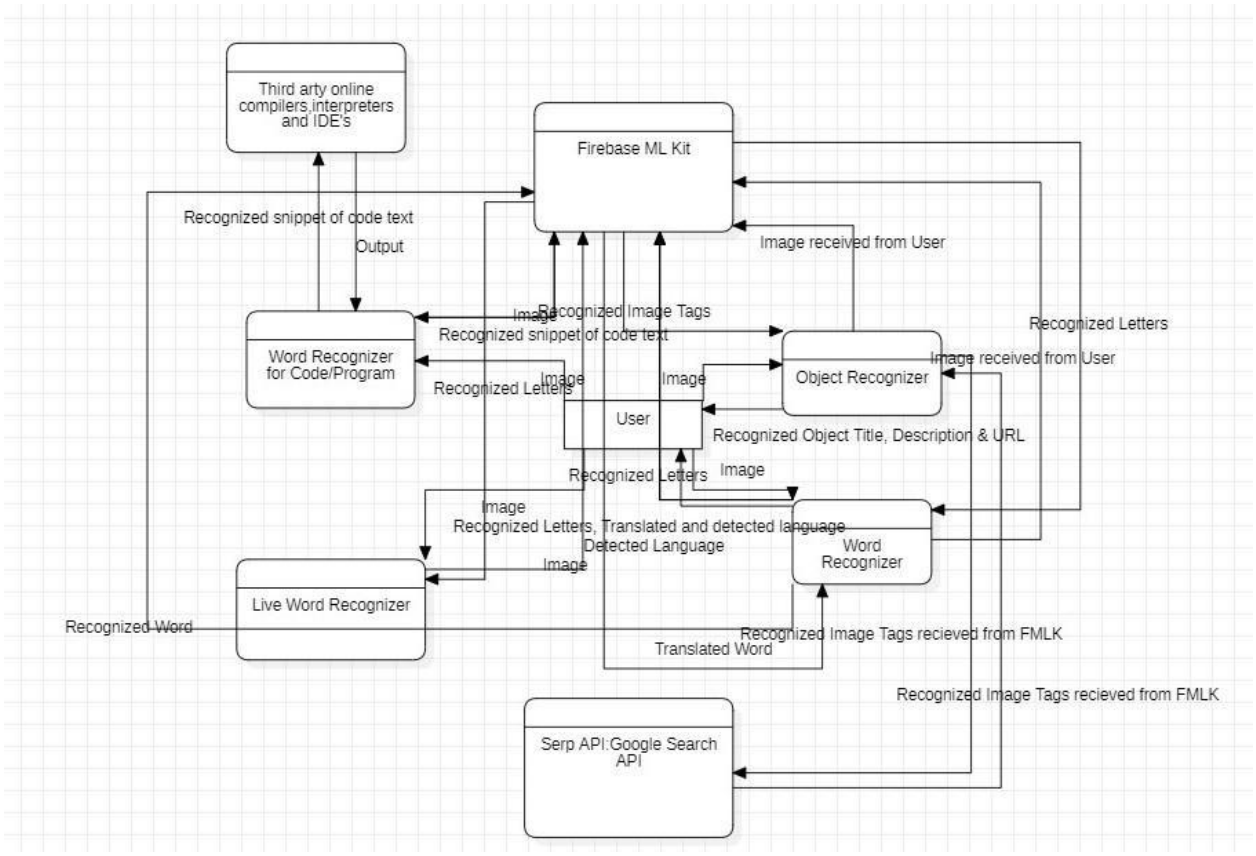


Fig. 3.5.3: FirstLevel Data Flow Diagram of Smart Lens

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

There are several notations for displaying data-flow diagrams. The notation presented above was described in 1979 by Tom DeMarco as part of structured analysis.

For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into sub-processes.

The data-flow diagram is a tool that is part of structured analysis and data modeling. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of data-flow plan is a site-oriented data-flow plan.

Data-flow diagrams can be regarded as inverted Petri nets, because places in such networks correspond to the semantics of data memories. Analogously, the semantics of transitions from Petri nets and data flows and functions from data-flow diagrams should be considered equivalent.

Chapter 4

Implementation and Testing

4.1 Code

AndroidManifest.xml
One of the most important file in android that keeps track of everything.

```
<uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SmartLens">
        <activity
            android:name=".SplashScreenActivity"
            android:exported="true"
            android:screenOrientation="portrait"
            android:theme="@style/Theme.Splash"
            tools:ignore="LockedOrientationActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ImageSearchResultActivity"
            android:exported="false"
            android:parentActivityName=".MainActivity"
            android:screenOrientation="portrait"
            tools:ignore="LockedOrientationActivity" />
        <activity
            android:name=".MainActivity"
```

```
        android:exported="true"
        android:screenOrientation="portrait"
        tools:ignore="LockedOrientationActivity">
        <meta-data
            android:name="com.google.firebase.ml.vision.DEPENDENCIES"
            android:value="ocr,model2,model3" />
        <!-- To use multiple models: android:value="ocr,model2,model3" -->
    </activity>
  </application>
```

MainActivity.java
Core Activity that keeps almost everything undr it.

```
public class MainActivity extends AppCompatActivity implements
BottomNavigationView.OnNavigationItemSelectedListener {
  @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      bottomNavigationView = findViewById(R.id.bottom_navigation_bar);
      bottomNavigationView.setOnNavigationItemSelectedListener(this);

fm.beginTransaction().add(R.id.frame_layout, fragment4, "4").hide(fragment4).commit();
      fm.beginTransaction().add(R.id.frame_layout, fragment3, "3").hide(fragment3).commit();
      fm.beginTransaction().add(R.id.frame_layout, fragment2, "2").hide(fragment2).commit();
      fm.beginTransaction().add(R.id.frame_layout, fragment1, "1").commit();



  }

@Override
  public boolean onNavigationItemSelected(@NonNull MenuItem item) {
      int itemId = item.getItemId();
      if (itemId == R.id.image_search) {
          fm.beginTransaction().hide(active).show(fragment1).commit();
          active = fragment1;

      } else if (itemId == R.id.word_search) {
          fm.beginTransaction().hide(active).show(fragment2).commit();
          active = fragment2;
```

```
        } else if (itemId == R.id.live_word_search) {
            fm.beginTransaction().hide(active).show(fragment3).commit();
            active = fragment3;
        } else if (itemId == R.id.code_search) {
            fm.beginTransaction().hide(active).show(fragment4).commit();
            active = fragment4;
        }

        return true;

    }
```

ImageFragement.java

```
public class ImageSearchFragment extends Fragment {

    private static final byte REQUEST_IMAGE_CAPTURE = 10;
    private ImageView clickedImg;
    private MaterialButton clickBtn, getResultsBtn;
    private TextView actionBarTv;
    private Bitmap bitmap;
    final Handler handler = new Handler();

    public ImageSearchFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_image_search, container, false);

        clickedImg = view.findViewById(R.id.placeholder_image);
        clickBtn = view.findViewById(R.id.photo_btn);
        getResultsBtn = view.findViewById(R.id.fetch_resultsbtn);
        actionBarTv = view.findViewById(R.id.activity_name);
        actionBarTv.setText("Image Search");
```

```
        int[] placeHolderImages = {R.drawable.img_search, R.drawable.add_img,
R.drawable.img_search2
                , R.drawable.img_search3, R.drawable.img_search4};
        Runnable runnable = new Runnable() {
            int i = 0;

            @Override
            public void run() {
                clickedImg.setImageResource(placeHolderImages[i]);
                i++;
                if (i >= placeHolderImages.length) {
                    i = 0;
                }
                handler.postDelayed(this, 3000);
            }
        };
        handler.postDelayed(runnable, 0);

        getResultsBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent resultsIntent = new Intent(getContext(), ImageSearchResultActivity.class);
                //this is required in next activity don't forget
                resultsIntent.putExtra("BitmapImage", bitmap);

                startActivity(resultsIntent);
            }
        });

        //ClickPic Btn
        clickBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (checkPermissions()) {
                    clickImage();
                } else {
                    requestPermissions();
                }

            }
```

```
        });


        return view;
    }

    private boolean checkPermissions() {
        int cameraPermission =
ContextCompat.checkSelfPermission(getActivity().getApplicationContext(), CAMERA);
        return cameraPermission == PackageManager.PERMISSION_GRANTED;
    }

    private void requestPermissions() {
        int PERMISSION_CODE = 200;
        requestPermissions(new String[]{
                CAMERA
        }, PERMISSION_CODE);
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (grantResults.length > 0) {
            boolean cameraPermission = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
            if (cameraPermission) {
                Toast.makeText(getActivity(), "Permission Granted!",
Toast.LENGTH_SHORT).show();
                clickImage();
            } else {
                Toast.makeText(getActivity(), "Permission Denied!",
Toast.LENGTH_SHORT).show();
            }
        }
    }

    private void clickImage() {
        Intent takeImageFromObjSearch = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

```
        if (takeImageFromObjSearch.resolveActivity(getActivity().getPackageManager()) != null)
{
            startActivityForResult(takeImageFromObjSearch, REQUEST_IMAGE_CAPTURE);
        }
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
            Bundle extras = data.getExtras();
            bitmap = (Bitmap) extras.get("data");
            handler.removeCallbacksAndMessages(null);
            clickedImg.setImageBitmap(bitmap);
            getResultsBtn.setVisibility(View.VISIBLE);
        }
    }
}
```

WordSerachFragment.java

```
public class WordSearchFragment extends Fragment {
 public WordSearchFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View root = inflater.inflate(R.layout.fragment_word_search, container, false);
  int[] placeHolderImages = {R.drawable.word_search, R.drawable.word_search1};
        Runnable runnable = new Runnable() {
            int i = 0;

            @Override
            public void run() {
                placeholderImage.setImageResource(placeHolderImages[i]);
                i++;
                if (i >= placeHolderImages.length) {
                    i = 0;
```

```
            }
            handler.postDelayed(this, 3000);
        }
    };
    handler.postDelayed(runnable, 0);

    photoBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (checkPermissions()) {
                clickImage();
            } else {
                requestPermissions();
            }
        }
    });

    fetchResultBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            fetchResults();
        }
    });

 private boolean checkPermissions() {
    int cameraPermission =
ContextCompat.checkSelfPermission(getActivity().getApplicationContext(), CAMERA);
    return cameraPermission == PackageManager.PERMISSION_GRANTED;
  }

  @Override
  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (grantResults.length > 0) {
        boolean cameraPermission = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
        if (cameraPermission) {
            Toast.makeText(getActivity(), "Permission Granted!",
Toast.LENGTH_SHORT).show();
```

```
                clickImage();
            } else {
                Toast.makeText(getActivity(), "Permission Denied!",
Toast.LENGTH_SHORT).show();
            }
        }
    }

    private void clickImage() {
        resultsTextView.setText("");
        Intent takeImage = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        if (takeImage.resolveActivity(getActivity().getPackageManager()) != null) {
            startActivityForResult(takeImage, REQUEST_IMAGE_CAPTURE);
        }
    }

    @Override
    public void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
            Bundle extras = data.getExtras();
            bitmap = (Bitmap) extras.get("data");
            handler.removeCallbacksAndMessages(null);
            placeholderImage.setImageBitmap(bitmap);
            fetchResultBtn.setVisibility(View.VISIBLE);
        }
    }

    private void requestPermissions() {
        int PERMISSION_CODE = 200;
        requestPermissions(new String[]{
                CAMERA
        }, PERMISSION_CODE);
    }

    private void fetchResults() {
//placeholderImage.getBackground();
        TextRecognizer textRecognizer = new TextRecognizer.Builder(getContext()).build();

        if (!textRecognizer.isOperational()) {
```

```
        } else {
            Frame frame = new Frame.Builder().setBitmap(bitmap).build();
            SparseArray<TextBlock> items = textRecognizer.detect(frame);
            StringBuilder stringBuilder = new StringBuilder();
            for (int i = 0; i < items.size(); i++) {
                TextBlock myItem = items.valueAt(i);
                stringBuilder.append(myItem.getValue());
                stringBuilder.append("\n");
            }
            String result = stringBuilder.toString();
            resultsTextView.setText(result);
            if (resultsTextView.getVisibility() == View.VISIBLE && result.length() >= 1) {
                translateChipGroup.setVisibility(View.VISIBLE);
            }
        }
    }
}
```

LiveWordSearchFragment.java

```
public class LiveWordSearchFragment extends Fragment {
 public LiveWordSearchFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_live_word_search, container, false);

    private void startCamera() {
        final TextRecognizer textRecognizer = new TextRecognizer.Builder(getContext()).build();

        if (!textRecognizer.isOperational()) {
//            Log.i("LiveWordSearchFragment", "startCamera: Dependencies not loaded yet");
        } else {
            cameraSource = new CameraSource.Builder(getContext(), textRecognizer)

.setFacing(CameraSource.CAMERA_FACING_BACK).setRequestedPreviewSize(1280, 1024)
                    .setAutoFocusEnabled(true)
```

```
                .setRequestedFps(2.0f).build();

        surfaceView.getHolder().addCallback(new SurfaceHolder.Callback() {
            @Override
            public void surfaceCreated(@NonNull SurfaceHolder surfaceHolder) {
                try {
                    if (ActivityCompat.checkSelfPermission(getContext(),
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
                        ActivityCompat.requestPermissions(getActivity(), new
String[]{Manifest.permission.CAMERA}, ASK_PERMISSION);
                        return;
                    }
                    cameraSource.start(surfaceView.getHolder());
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

            @Override
            public void surfaceChanged(@NonNull SurfaceHolder surfaceHolder, int i, int i1, int
i2) {
                //Not required but we still need to override to release the resources
            }

            @Override
            public void surfaceDestroyed(@NonNull SurfaceHolder surfaceHolder) {
                cameraSource.stop();
            }

        });
    }

    textRecognizer.setProcessor(new Detector.Processor<TextBlock>() {
        @Override
        public void release() {
            //d
        }

        @Override
        public void receiveDetections(Detector.Detections<TextBlock> detections) {
```

```
            final SparseArray<TextBlock> items = detections.getDetectedItems();
            if (items.size() != 0) {
               textView.post(new Runnable() {
                  @Override
                  public void run() {
                     StringBuilder stringBuilder = new StringBuilder();

                     for (int i = 0; i < items.size(); i++) {
                        TextBlock item = items.valueAt(i);
                        stringBuilder.append(item.getValue());
                        stringBuilder.append("\n");
                     }
                     textView.setText(stringBuilder.toString());
                  }
               });
            }
         }
      });

   }
}
```

CodeSearchFragment.java

```
public class CodeSearchFragment extends Fragment {
 public CodeSearchFragment() {
      // Required empty public constructor
   }


   @Override
   public View onCreateView(LayoutInflater inflater, ViewGroup container,
                            Bundle savedInstanceState) {
      // Inflate the layout for this fragment
      View root = inflater.inflate(R.layout.fragment_code_search, container, false);
 int[] placeHolderImages = {R.drawable.code_search, R.drawable.code_search1,
R.drawable.code_search2};
      Runnable runnable = new Runnable() {
         int i = 0;

         @Override
```

```
        public void run() {
            placeholderImage.setImageResource(placeHolderImages[i]);
            i++;
            if (i >= placeHolderImages.length) {
                i = 0;
            }
            handler.postDelayed(this, 3000);
        }
    };
    handler.postDelayed(runnable, 0);

    startScan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if (startScan.getText().toString().equals("Stop Scan")) {
                placeholderImage.setVisibility(View.VISIBLE);
                copyResultBtn.setEnabled(false);
                cardView.setVisibility(View.GONE);
                surfaceView.setVisibility(View.GONE);
                cameraSource.stop();
                startScan.setText("Start Scan");
            } else {
                placeholderImage.setVisibility(View.GONE);
                cardView.setVisibility(View.VISIBLE);
                surfaceView.setVisibility(View.VISIBLE);
                try {
                    if (ActivityCompat.checkSelfPermission(getContext(),
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
                        ActivityCompat.requestPermissions(getActivity(), new
String[]{Manifest.permission.CAMERA}, ASK_PERMISSION);
                        return;
                    }
                    cameraSource.start(surfaceView.getHolder());
                } catch (Exception e) {
                    e.printStackTrace();
                }
                startCamera();
                copyResultBtn.setEnabled(true);
                startScan.setText("Stop Scan");
            }
```

```
        }
    });
  private void startCamera() {
    final TextRecognizer textRecognizer = new TextRecognizer.Builder(getContext()).build();

    if (!textRecognizer.isOperational()) {
//          Log.i("LiveWordSearchFragment", "startCamera: Dependencies not loaded yet");
    } else {
        cameraSource = new CameraSource.Builder(getContext(), textRecognizer)

.setFacing(CameraSource.CAMERA_FACING_BACK).setRequestedPreviewSize(1280, 1024)
                .setAutoFocusEnabled(true)
                .setRequestedFps(2.0f).build();

        surfaceView.getHolder().addCallback(new SurfaceHolder.Callback() {
            @Override
            public void surfaceCreated(@NonNull SurfaceHolder surfaceHolder) {
                try {
                    if (ActivityCompat.checkSelfPermission(getContext(),
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
                        ActivityCompat.requestPermissions(getActivity(), new
String[]{Manifest.permission.CAMERA}, ASK_PERMISSION);
                        return;
                    }
                    cameraSource.start(surfaceView.getHolder());
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }

            @Override
            public void surfaceChanged(@NonNull SurfaceHolder surfaceHolder, int i, int i1, int
i2) {
                //Not required but we still need to override to release the resources
            }

            @Override
            public void surfaceDestroyed(@NonNull SurfaceHolder surfaceHolder) {
                cameraSource.stop();
            }
```

```
        });
    }

    textRecognizer.setProcessor(new Detector.Processor<TextBlock>() {
        @Override
        public void release() {
            //d
        }

        @Override
        public void receiveDetections(Detector.Detections<TextBlock> detections) {
            final SparseArray<TextBlock> items = detections.getDetectedItems();
            if (items.size() != 0) {
                resultsTextView.post(new Runnable() {
                    @Override
                    public void run() {
                        StringBuilder stringBuilder = new StringBuilder();

                        for (int i = 0; i < items.size(); i++) {
                            TextBlock item = items.valueAt(i);
                            stringBuilder.append(item.getValue());
                            stringBuilder.append("\n");
                        }
                        resultsTextView.setText(stringBuilder.toString());
                        if (resultsTextView.getText().toString().length() >= 1) {
                            resultsTextView.setVisibility(View.VISIBLE);
                        }
                    }
                });
            }
        }
    });

  }
}
```

SearchRVAdapter.java

```
public class SearchRVAdapter extends ArrayAdapter<SearchRVModel> {
  //Constructor
```

```
    public SearchRVAdapter(Activity context, ArrayList<SearchRVModel> searchRVModels) {
        super(context, 0, searchRVModels);
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

        View listItemView = convertView;
        //This is scrapView
        //Now we need to check if the scrapView is null then add a view from the custom list_item
we made

        if (listItemView == null) {
            listItemView = LayoutInflater.from(getContext()).inflate(R.layout.list_item, parent,
false);
        }

        //get the current position from the  arrayList
        SearchRVModel currentSearch = getItem(position);

        //Bind all the textViews now by findViewById and set its text
        ShapeableImageView imageURL = listItemView.findViewById(R.id.image_url);
        Glide.with(getContext()).load(currentSearch.getImageUrl()).into(imageURL);

        TextView titleTextView = listItemView.findViewById(R.id.title_tv);
        titleTextView.setText(currentSearch.getTitle());

        TextView urlTextView = listItemView.findViewById(R.id.site_tv);
        urlTextView.setText(currentSearch.getLink());


        listItemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Intent.ACTION_VIEW);
                intent.setData(Uri.parse(currentSearch.getLink()));
                getContext().startActivity(intent);
            }
        });
```

```
        return listItemView;
    }
}
```

SplashScreen.java

```
public class SplashScreenActivity extends AppCompatActivity {

    private Handler handler;
    private Runnable runnable;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_splash_screen);
        runnable = new Runnable() {
            @Override
            public void run() {
                startActivity(new Intent(SplashScreenActivity.this, MainActivity.class));
                finish();
            }
        };

        handler = new Handler();
        handler.postDelayed(runnable, 2000);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (handler != null && runnable != null) {
            handler.removeCallbacks(runnable);
        }
    }
}
```

Action_bar.xml
Custom Action bar or app

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
android:background="@drawable/top_rounded_for_display_activity">

<ImageView
    android:id="@+id/back_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_centerInParent="true"
    android:layout_marginStart="10dp"
    android:clickable="true"
    android:focusable="true"
    android:foreground="?android:attr/actionBarItemBackground"
    android:padding="10dp"
    android:src="@drawable/ic_round_arrow_back_ios_new_24"
    android:visibility="gone"
    tools:visibility="visible" />


<TextView
    android:id="@+id/activity_name"
    style="@style/TextAppearance.AppCompat.Widget.ActionBar.Title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:ellipsize="end"
    android:maxEms="9"
    android:maxLines="1"
    android:textColor="@color/white"
    tools:text="Activity Name" />

<ImageView
    android:id="@+id/copy_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:layout_toStartOf="@+id/share_button"
    android:clickable="true"
```

```
        android:focusable="true"
        android:foreground="?android:attr/actionBarItemBackground"
        android:padding="10dp"
        android:src="@drawable/ic_round_content_copy_24"
        android:visibility="gone"
        tools:visibility="visible" />

    <ImageView
        android:id="@+id/share_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_centerInParent="true"
        android:layout_marginEnd="10dp"
        android:clickable="true"
        android:focusable="true"
        android:foreground="?android:attr/actionBarItemBackground"
        android:padding="10dp"
        android:src="@drawable/ic_round_share_24"
        android:visibility="gone"
        tools:visibility="visible" />

</RelativeLayout>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="?attr/activity_bg"
    tools:context=".MainActivity">

    <FrameLayout
        android:id="@+id/frame_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_above="@+id/bottom_navigation_bar" />
```

```
    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation_bar"
        android:layout_width="wrap_content"
        android:background="@drawable/bottom_navbar_custombg"
        app:menu="@menu/bottom_navbar_menu"
        app:labelVisibilityMode="labeled"
        app:itemIconTint="@color/white"
        app:itemTextColor="@color/white"
        android:layout_alignParentBottom="true"
        android:layout_height="72dp"/>

</RelativeLayout>
```

Activity_word_search_result.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".WordSearchActivityResult">


    <TextView
        android:id="@+id/translator_placeholder"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:gravity="center"
        android:text="Translatator"
        android:textSize="24sp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/translator_placeholder"
        android:layout_marginStart="16dp"
        android:layout_marginTop="16dp"
        android:gravity="start"
        android:text="Translate to"
```

```
    android:textSize="16sp" />

<Spinner
    android:id="@+id/tran_spin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/translator_placeholder"
    android:layout_marginTop="13dp"
    android:layout_marginEnd="16dp" />

<com.google.android.material.textfield.TextInputLayout
    android:id="@+id/idEdtSourceParent"
    style="@style/Widget.MaterialComponents.TextInputLayout.FilledBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/tran_spin"
    android:hint="Enter Text"
    android:textColorHint="@color/white"
    app:hintTextColor="@color/white">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/idEdtSource"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:ems="10"
        android:importantForAutofill="no"
        android:inputType="textImeMultiLine|textMultiLine" />

</com.google.android.material.textfield.TextInputLayout>

<Button
    android:id="@+id/translate_btn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/idEdtSourceParent"
    android:layout_centerInParent="true"
    android:text="Translate" />

<TextView
    android:id="@+id/translated_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/translate_btn"
    android:layout_marginStart="16dp"
```

```
        android:layout_marginTop="16dp"
        android:gravity="center"
        android:textSize="16sp" />

</RelativeLayout>
```

Fragment_image_search.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="?attr/activity_bg"
    tools:context=".fragments.ImageSearchFragment">

    <include
        android:id="@+id/action_bar"
        layout="@layout/action_bar" />

    <ImageView
        android:id="@+id/placeholder_image"
        android:layout_width="400dp"
        android:layout_height="400dp"
        android:layout_below="@+id/action_bar"
        android:layout_centerInParent="true"
        android:layout_marginStart="20dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="16dp"
        tools:src="@drawable/img_search" />

    <com.google.android.material.button.MaterialButton
        android:id="@+id/photo_btn"
        style="@style/MyNormalBtn"
        android:layout_below="@+id/placeholder_image"
        android:text="Take a photo"
        android:textSize="16sp"
        app:cornerRadius="30dp"
        app:icon="@drawable/ic_round_add_a_photo_24"
        app:iconSize="25dp"
        app:iconTint="@color/white" />
```

```
    <com.google.android.material.button.MaterialButton
        android:id="@+id/fetch_resultsbtn"
        style="@style/MyOutlineBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/photo_btn"
        android:text="Fetch Results"
        android:textColor="?attr/outline_btn_txt_icon"
        android:textSize="16sp"
        app:cornerRadius="30dp"
        app:icon="@drawable/ic_round_image_24"
        app:iconSize="25dp"
        app:iconTint="?attr/outline_btn_txt_icon"
        app:strokeColor="?attr/action_bar"
        tools:visibility="visible"
        android:visibility="gone"/>

</RelativeLayout>
```

Fragment_live_word_search.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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="?attr/activity_bg"
    android:orientation="vertical"
    tools:context=".fragments.LiveWordSearchFragment">

    <include
        android:id="@+id/action_bar"
        layout="@layout/action_bar" />

    <androidx.cardview.widget.CardView
        android:id="@+id/card_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp"
        android:layout_weight="2"
        app:cardBackgroundColor="@color/black"
        app:cardCornerRadius="32dp"
        app:cardPreventCornerOverlap="false">
```

```
        <SurfaceView
            android:id="@+id/surface_view"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

    </androidx.cardview.widget.CardView>

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="3">

        <TextView
            android:id="@+id/text_view"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ellipsize="end"
            android:gravity="start"
            android:maxLines="30"
            android:layout_marginBottom="5dp"
            android:layout_marginStart="15dp"
            android:layout_marginEnd="15dp"

            android:text="Place the camera over the text"
            android:textColor="?attr/outline_btn_txt_icon"
            android:textSize="16sp"
            android:textStyle="bold" />
    </ScrollView>
</LinearLayout>
```

Fragment_word_search.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="?attr/activity_bg"
    tools:context=".fragments.WordSearchFragment">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```
<include
    android:id="@+id/action_bar"
    layout="@layout/action_bar" />

<ImageView
    android:id="@+id/placeholder_image"
    android:layout_width="400dp"
    android:layout_height="400dp"
    android:layout_below="@+id/action_bar"
    android:layout_centerInParent="true"
    android:layout_marginStart="20dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="20dp"
    android:layout_marginBottom="16dp"
    tools:src="@drawable/word_search1" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/photo_btn"
    style="@style/MyNormalBtn"
    android:layout_below="@+id/placeholder_image"
    android:text="Take a photo"
    android:textSize="16sp"
    app:cornerRadius="20dp"
    app:icon="@drawable/ic_round_add_a_photo_24"
    app:iconSize="25dp"
    app:iconTint="@color/white" />

<com.google.android.material.button.MaterialButton
    android:id="@+id/fetch_resultsbtn"
    style="@style/MyOutlineBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/photo_btn"
    android:text="Fetch Results"
    android:textColor="?attr/outline_btn_txt_icon"
    android:textSize="16sp"
    android:visibility="gone"
    app:cornerRadius="20dp"
    app:icon="@drawable/ic_round_image_24"
    app:iconSize="25dp"
    app:iconTint="?attr/outline_btn_txt_icon"
    app:strokeColor="?attr/action_bar"
    tools:visibility="visible" />
```

```
<com.google.android.material.chip.ChipGroup
    android:id="@+id/translate_chip_grp"
    style="@style/Widget.MaterialComponents.Chip.Action"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/fetch_resultsbtn"
    android:layout_centerInParent="true"
    android:layout_gravity="center"
    android:layout_marginTop="16dp"
    android:visibility="gone"
    app:chipSpacingHorizontal="10dp"
    app:singleLine="true"
    app:singleSelection="true"
    tools:visibility="visible">

    <com.google.android.material.chip.Chip
        android:id="@+id/copy_chip"
        style="@style/Widget.MaterialComponents.Chip.Action"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Copy"
        android:textColor="?attr/outline_btn_txt_icon"
        app:chipBackgroundColor="?attr/activity_bg"
        app:chipIcon="@drawable/ic_round_content_copy_24"
        app:chipIconTint="?attr/outline_btn_txt_icon"
        app:chipStartPadding="10dp"
        app:chipStrokeColor="?attr/action_bar"
        app:chipStrokeWidth="1dp" />

    <com.google.android.material.chip.Chip
        android:id="@+id/share_chip"
        style="@style/Widget.MaterialComponents.Chip.Action"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Share"
        android:textColor="?attr/outline_btn_txt_icon"
        app:chipBackgroundColor="?attr/activity_bg"
        app:chipIcon="@drawable/ic_round_share_24"
        app:chipIconTint="?attr/outline_btn_txt_icon"
        app:chipStartPadding="10dp"
        app:chipStrokeColor="?attr/action_bar"
        app:chipStrokeWidth="1dp" />
```

```
        <com.google.android.material.chip.Chip
            android:id="@+id/listen_chip"
            style="@style/Widget.MaterialComponents.Chip.Action"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Listen"
            android:textColor="?attr/outline_btn_txt_icon"
            app:chipBackgroundColor="?attr/activity_bg"
            app:chipIcon="@drawable/ic_listen"
            app:chipIconTint="?attr/outline_btn_txt_icon"
            app:chipStartPadding="10dp"
            app:chipStrokeColor="?attr/action_bar"
            app:chipStrokeWidth="1dp" />

        <!--        <com.google.android.material.chip.Chip-->
        <!--            android:id="@+id/translate_chip"-->
        <!--            style="@style/Widget.MaterialComponents.Chip.Action"-->
        <!--            android:layout_width="wrap_content"-->
        <!--            android:layout_height="wrap_content"-->
        <!--            android:text="Translate"-->
        <!--            app:chipIcon="@drawable/ic_codesearch" />-->
    </com.google.android.material.chip.ChipGroup>

    <TextView
        android:id="@+id/results_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/translate_chip_grp"
        android:layout_centerInParent="true"
        android:layout_gravity="center"
        android:layout_marginStart="16dp"
        android:layout_marginTop="6dp"
        android:layout_marginEnd="16dp"
        android:textColor="?attr/outline_btn_txt_icon"
        android:textSize="18sp"
        android:textStyle="bold"
        tools:text="He read about a hike called the incline in the guidebook. It said it was a strenuous hike and to bring plenty of water. “A beautiful hike to the clouds” described one review. “Not for the faint-hearted,” said another. “Not too bad of a workout”, bragged a third review. I thought I’d hike it when I fly in from Maryland on my day off from the senior citizen's wellness conference. I hiked 2 miles a day around the neighborhood so I could handle a 1.1-mile hike. What a fo" />

</RelativeLayout>
```

</ScrollView>

4.2 Testing Approach

Testing is also an important phase of any project as it tests our product whether it is properly working or not.
There are basic 4 types of testing that can be included in our project which help to test the application in different ways before running the application

4.1.1 Unit Testing

Unit tests are typically automated tests written and run by software developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended. In procedural programming, a unit could be an entire module, but it is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. By writing tests first for the smallest testable units, then the compound behaviors between those, one can build up comprehensive tests for complex applications.
To isolate issues that may arise, each test case should be tested independently. Substitutes such as method stubs, mock objects, fakes, and test harnesses can be used to assist testing a module in isolation.
During development, a software developer may code criteria, or results that are known to be good, into the test to verify the unit's correctness. During test case execution, frameworks log tests that fail any criterion and report them in a summary. For this, the most commonly used approach is test - function - expected value.
Writing and maintaining unit tests can be made faster by using parameterized tests. These allow the execution of one test multiple times with different input sets, thus reducing test code duplication. Unlike traditional unit tests, which are usually closed methods and test invariant conditions, parameterized tests take any set of parameters. Parameterized tests are supported by TestNG, JUnit and its .Net counterpart, XUnit. Suitable parameters for the unit tests may be supplied manually or in some cases are automatically generated by the test framework. In recent years support was added for writing more powerful (unit) tests, leveraging the concept of theories, test cases that execute the same steps, but using test data generated at runtime, unlike regular parameterized tests that use the same execution steps with input sets that are pre-defined

4.1.2 Module Testing

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing the whole software program at once, module testing recommends testing the smaller building blocks of the program.

Module testing is largely a white box oriented. The objective of doing Module testing is not to

demonstrate proper functioning of the module but to demonstrate the presence of an error in the module.

Module level testing allows to implement parallelism into the testing process by giving the opportunity to test multiple modules simultaneously.

A software application is composed of a number of software modules that are integrated together to form a software application. A module itself is a program written in a particular language that is composed of subroutines, subprograms, classes, procedures, and functions. Module testing definition: The testing of these module composition units can be done through Module testing.
Module testing can be classified largely into a white box orientation. Sometimes Module testing is also referred to as Program or Component Testing. The main objective of conducting Module testing is to ensure that the module is fully tested and functional in order to participate in Application testing. Module testing reduces the number of defects or errors which could be discovered during Application testing in the later stage of testing. It also introduces parallelism into the testing approach as it provides an opportunity to test multiple application modules at the same time.
Module testing is primarily focused on testing software modules or sub-program instead of testing the entire software application at once. Module testing in software engineering is very beneficial and always recommended as it is very easy to identify, understand and fix the defects at the module level instead of fixing them at the Application level. Till now the first paragraph we learned module testing definition.

4.1.3 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.
Some different types of integration testing are big-bang, mixed (sandwich), risky-hardest, top-down, and bottom-up. Other Integration Patterns are: collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration.
In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process. However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing.
Bottom-up testing is an approach to integrated testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated

until the component at the top of the hierarchy is tested. All the bottom or low- level modules, procedures or functions are integrated and then tested. After the integration testing of lower level integrated modules, the next level of modules will be formed and can be used for integration testing. This approach is helpful only when all or most of the modules of the same development level are ready. This method also helps to determine the levels of software developed and makes it easier to report testing progress in the form of a percentage.
Top-down testing is an approach to integrated testing where the top integrated modules are tested and the branch of the module is tested step by step until the end of the related module.
Sandwich testing is an approach to combine top down testing with bottom up testing.
One limitation to this sort of testing is that any conditions not stated in specified integration tests, outside of the confirmation of the execution of design items, will generally not be tested**.**

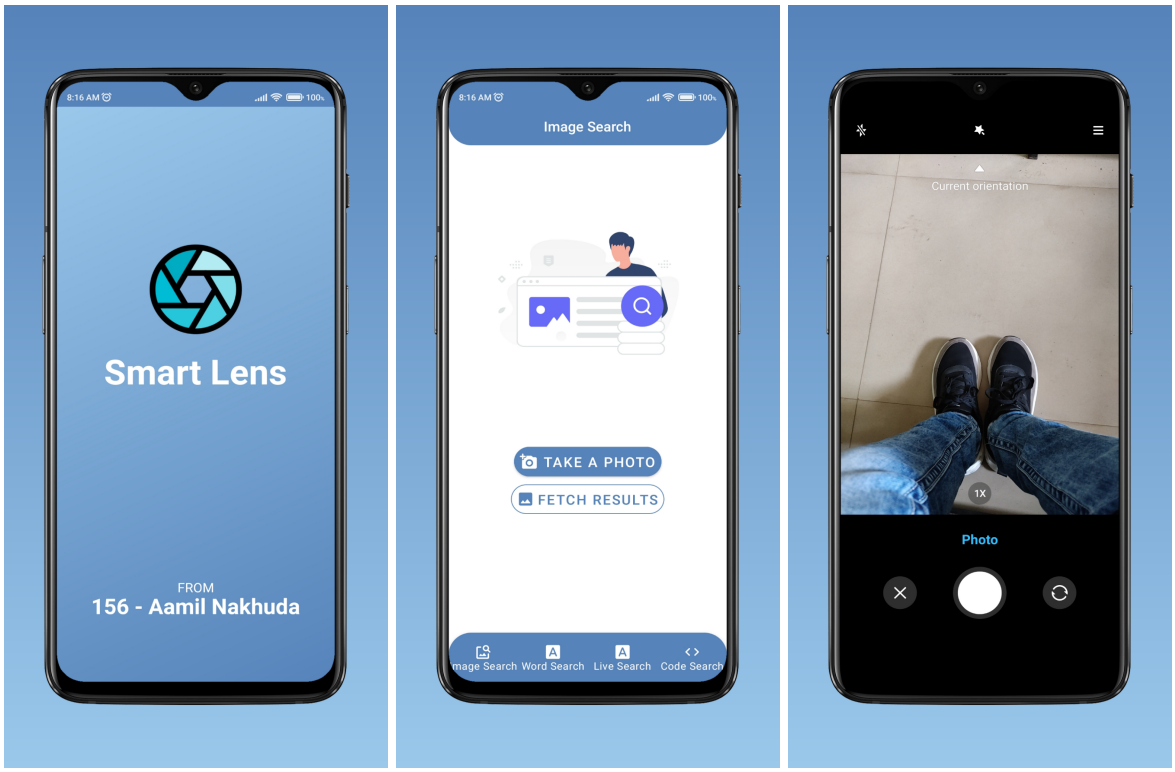
4.1.4 System Testing

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.
In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.
System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing.
System Testing is a black-box testing.
System Testing is performed after the integration testing and before the acceptance testing.
System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system..

Chapter 5

Results and Discussions



Splash Screen(Introductory Screen), Image Search Screen, Taking Image from Camera Screen

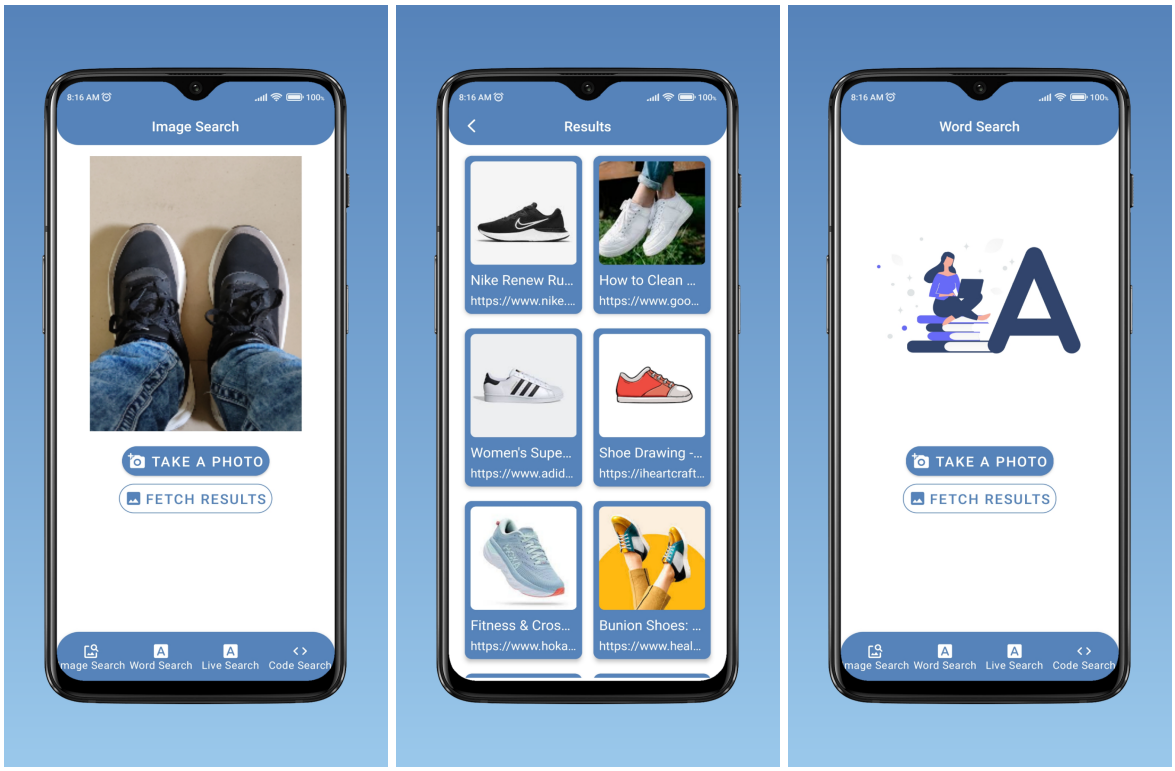
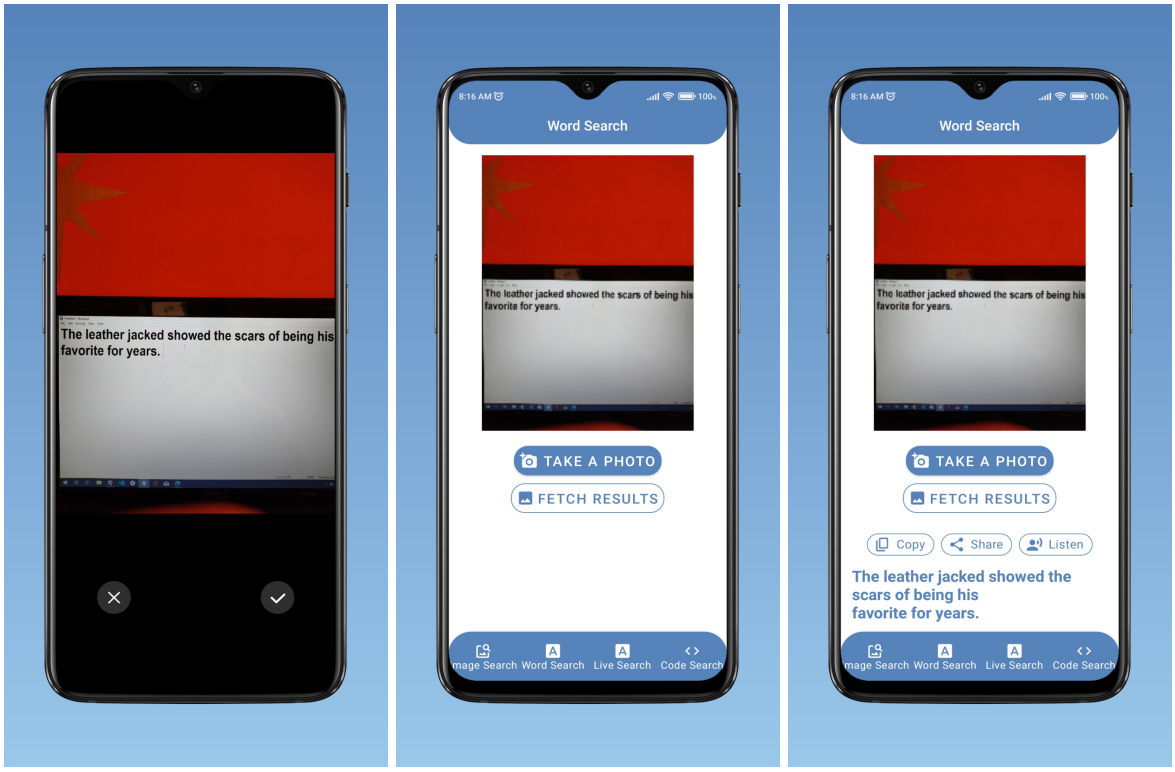
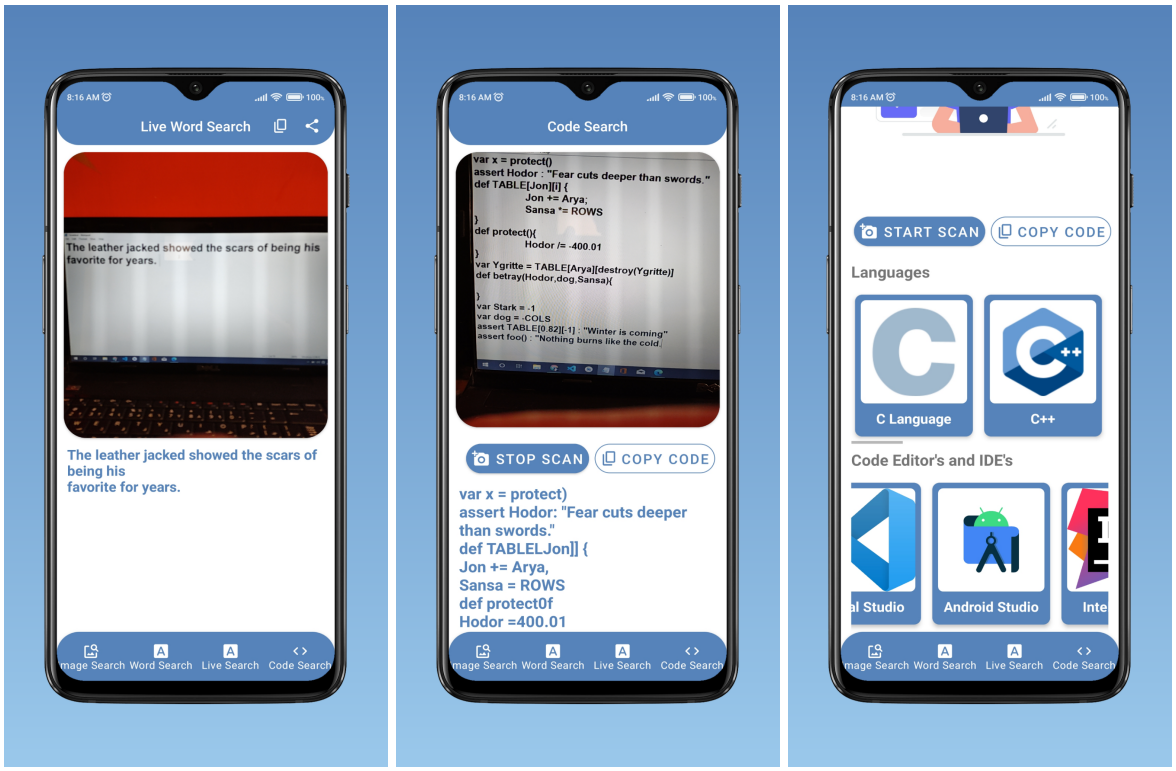


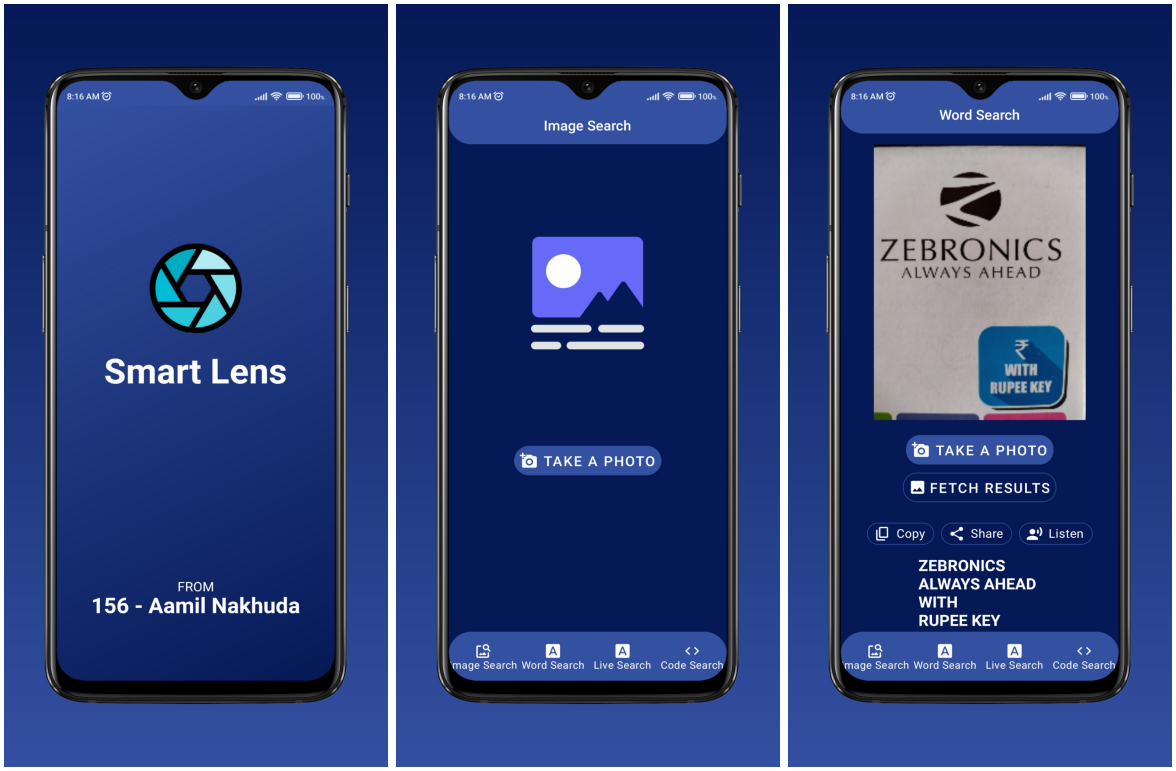
Image Search Screen, Results Screen, Word Search Screen



Taking Image from Camera Screen, Word Search Screen, Results



Live Word Search Screen with Results, Code Search Screen With Results and various Code Editors in different languages and famous Text Editors and IDE’s



Some of the above functionality in Dynamic Dark Mode

Chapter 6

Conclusion and Future Work

6.1 Conclusion

Smart Lens is an Android application inspired from Google Lens whose aim and vision was to make some of the features better than Google Lens and to provide quick and efficient results of Object, Word and Code with just a click of a photo. Its goal was also to save precious time, efforts and energy of the user in finding out results for its searches.

6.2 Limitations of the System

The SerpApi as well as the Firebase Ml Kit, which I am using as a backend has some limitations of a limited number of usages. Although these can be easily covered by paying them money with respect to the charges they provide for extra service which is given in their documentation. Other than that there is no limitation or disadvantage.

6.3 Future Scope of the Project

The conclusions can be summarized in a fairly short chapter around 300 words. Also include limitations of your system and future scope (12, justified)

SKY'S THE LIMIT.

1. Localization
2. Computer language Detection
3. Using API for compiling/interpreting instead of redirecting to website
4. More different settings
5. Animations
6. More Themes
7. Barcode and QR code Scanning
8. Landmark recognition
9. Object Detection and Tracking
10. Translation of languages
11. Language Identification
12. And much more…

Chapter 7

References

[1] https://www.youtube.com/

[2] https://www.google.com/

[3] https://www.github.com/

[4] https://www.stackoverflow.com/