**Shahmeer Khan – 199310, Talha Ismail - 199368, Yahya Tahir - 198005**

BSCS6A  NUST - SEECS

**A Search App**

**Semester Project: CS212 - Object Oriented Programming**

**Dr. Anis Ur Rehman**

**BSCS6A**

# Abstract

For the semester project, using Java as a base language, our group weighed-in our options of what would make the most usable piece of software and which medium it should exist on. We concluded that making an app for a handheld device would be far superior in terms of portability and ease of use. Next, we decided on the idea, the best of which seemed to be an app that almost all of us wanted that didn’t already exist in the app market. An app that would increase our level of productivity in university for the years to come. An app that automatically captures, scans and searches whatever the user throws at it in order to provide useful results that were previously obtained by manually typing in the questions. The resulting app brought this idea to reality.

# Introduction

The app is based around the Android Ecosystem for API Versions 24+ (6.0+). It’s a bare minimum app that is at time of download, only 647kb. The app uses the camera to capture the image and the rest is handled by the OCR and Search algorithms. The OCR algorithms are from Googles own open source API library that they’ve setup for implementations just like our app. It’s also baked right into Android Studio in the form of classes like TextRecogniser (discussed later), also making implementation easier. The search API is also that of Google’s, since it’s the most complete search engine out there.

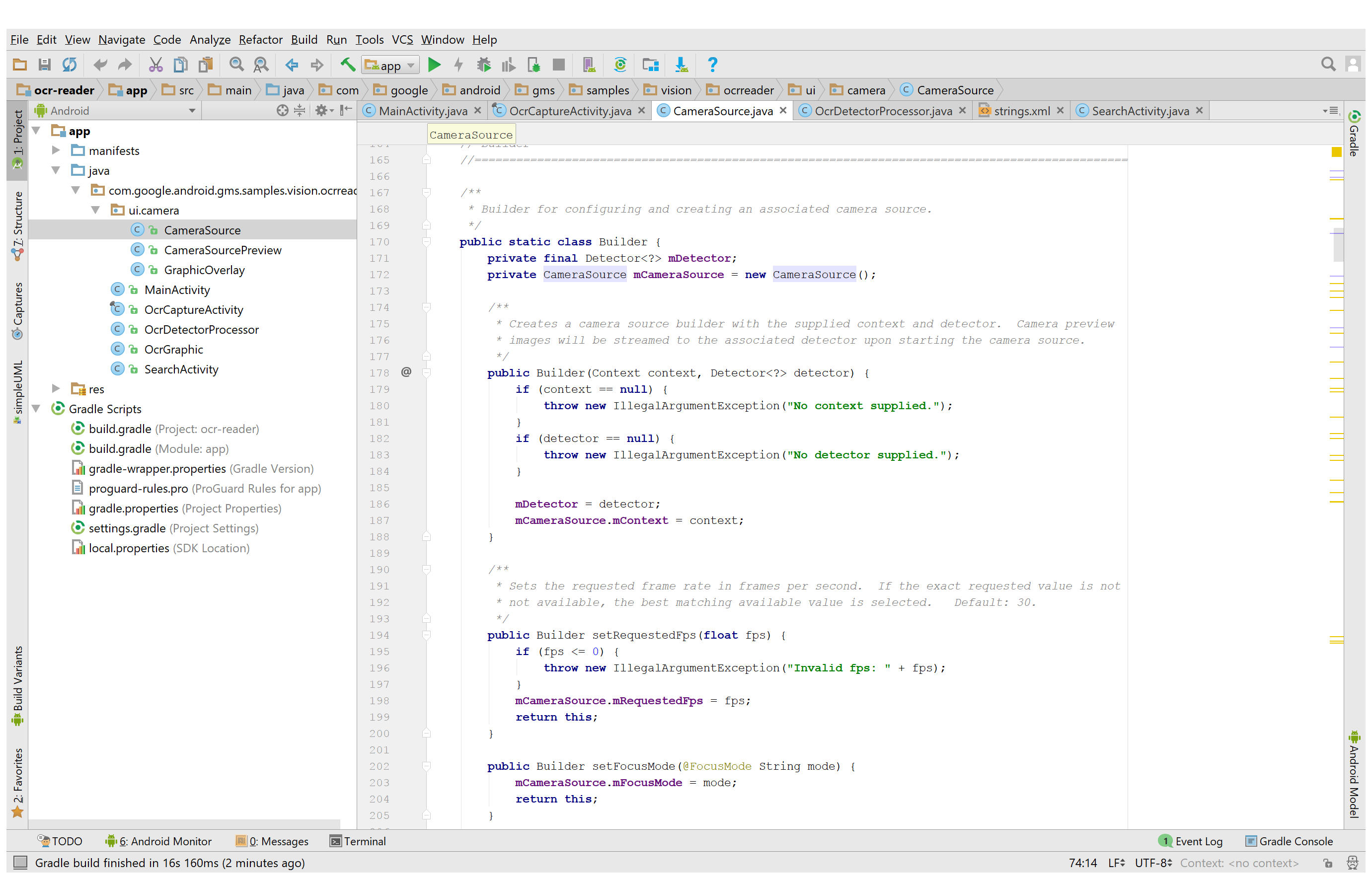
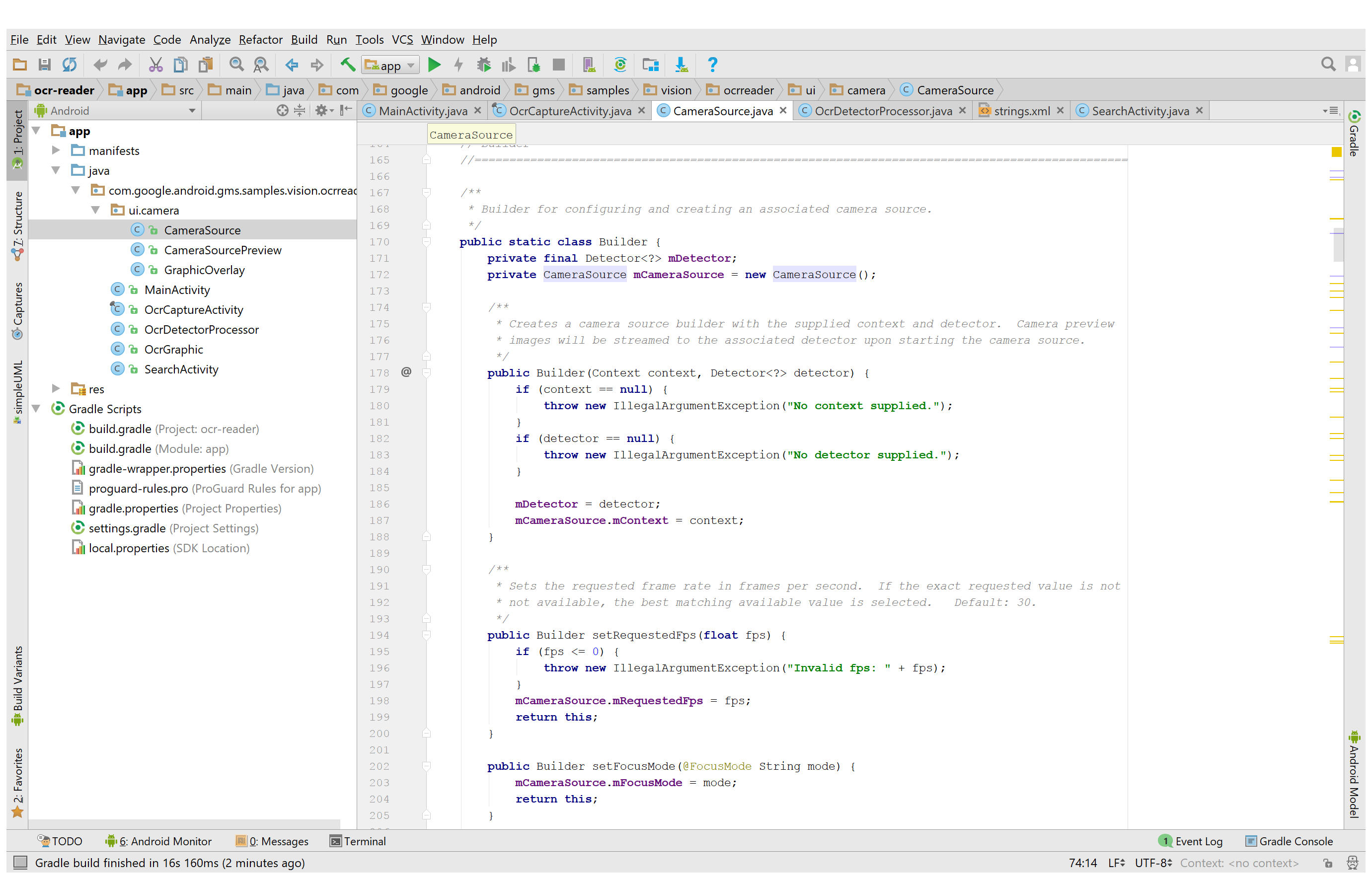
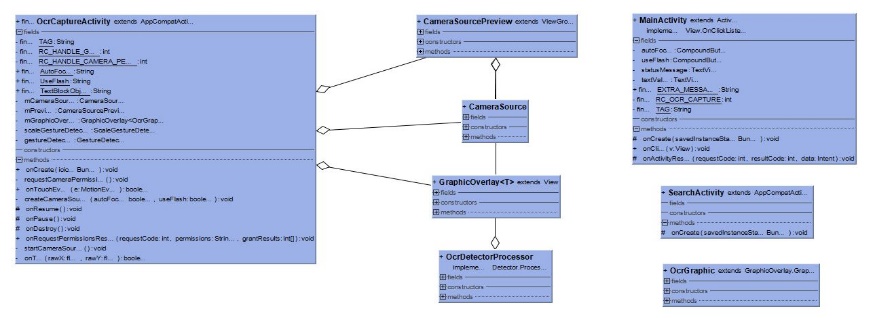
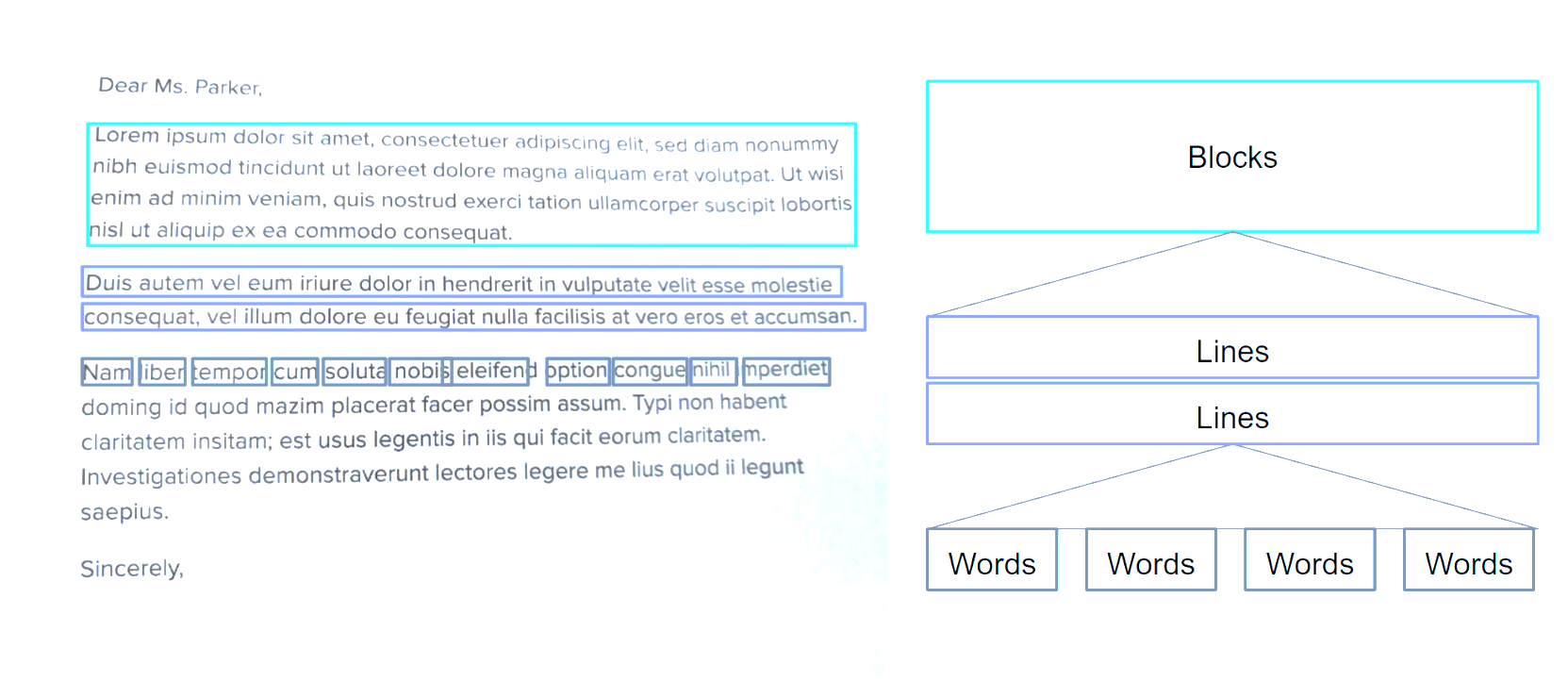
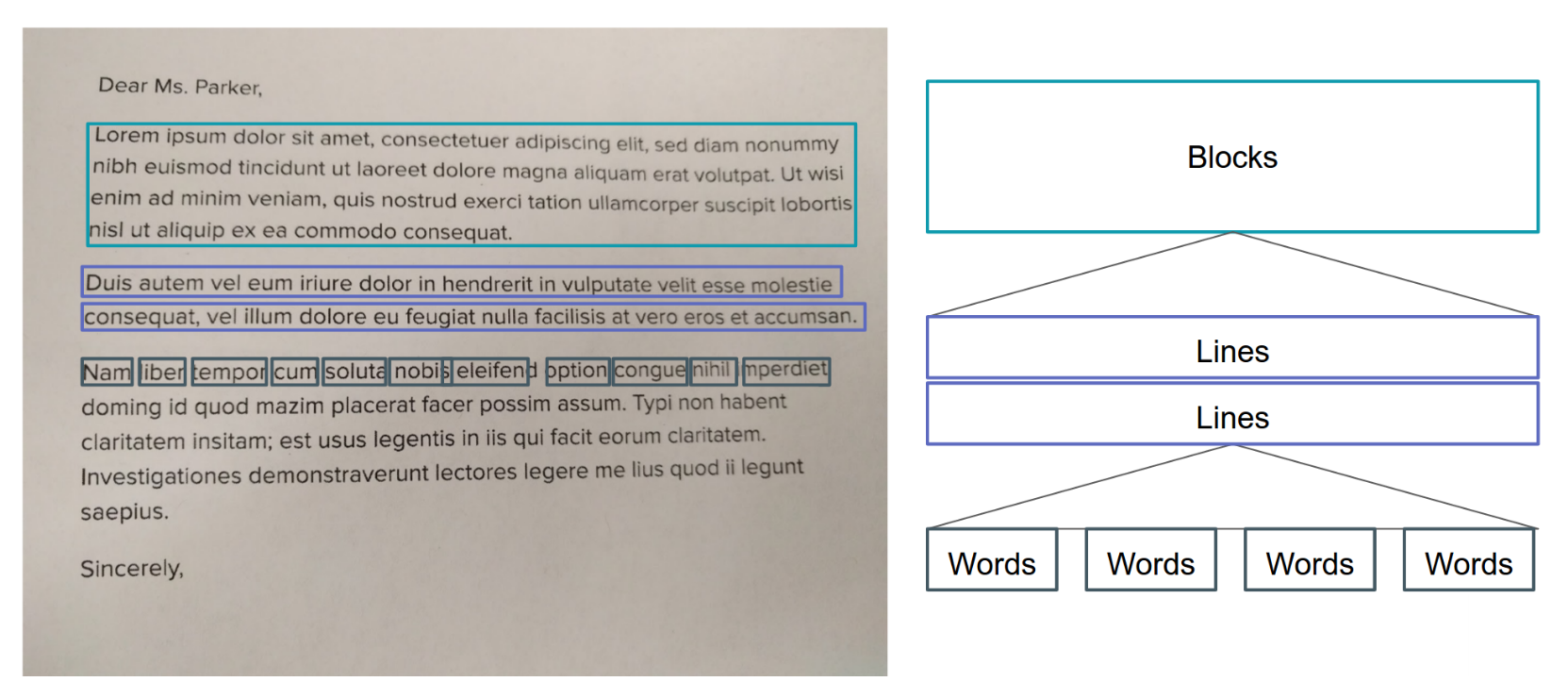
# Problem and Challenges

Since Android development was relatively new to us, we had to learn it from scratch. The fact that it uses Java made it easier to understand. Eventually we had one main developer (Shahmeer) who understood Android, and Talha and Yahya were there to help him out with the Java aspects of Android. The next part was understanding the Google Vision API – how it worked and how to integrate it into the app. Once that was done, the Search API was relatively easy to work around. Since so much time was given to these two tasks, the main UI and Layout lags behind the rest of the app.

# Proposed Solution

Our proposed solution consisted of a Camera widget the captures the Frame data, the data is sent live to be detected, leading to the text preview where the user can decide whether they want to continue, and finally the Google Search element. All these steps needed to have Activities in Android and intents to communicate data from one part to the other.

# Implementation

In Android, there is a Main Activity, that links all the activities via intents. “ui.camera” is a built in way to access the camera via an overlay on whichever activity is relevant (in our case, the OCR Capture Activity). The OCR Capture Activity is there to capture the image, and convert it into a frame object which constitutes all the data of an image as well as other details and contains **TextRecognizer textRecognizer = new TextRecognizer.Builder (context).build()** which is how the text can be detected. The OCR Detector is there for live detection of frame objects and the OCR Graphic is there to provide an overlay at specific times. OCR Detector implements Detector.Processor<TextBlock>. In this case, a “TextBlock” is basically a block of text, as shown in the diagrams. We get the TextBlocks from the detection and create OcrGraphic objects for each text block that the processor detects. Text is recognised live by using the “detect (Frame frame)” method where frame objects are passed in from the OCR Capture Activity. We could get the Lines from a TextBlock by calling getComponents (a built in method) and iterate over it to get the location and values of the text inside but we decided not to as our model was a scanner and detector of questions that could stretch to multiple lines and we wanted those whole questions to be searched. Also, the detector is smart enough to differentiate between sentences stretched between multiple lines.

The Search Activity used intents to pass the string of text to the search box. It basically used the search manager class and queries into Google Search . Intent intent = **new** Intent (Intent .***ACTION\_WEB\_SEARCH***); Is the intent used to go to search. The extra used is basically the text that was outputted from the OCR Processor and comes from the MainActivity. (intent.putExtra(SearchManager.***QUERY***, message); )

# Discussion and Conclusion

This search app started out as an idea that none of us thought could be done. As time went on, we realised it isn’t that hard to make an app and have our project physically available to take anywhere unlike what most of our classmates did. The main hurdle was learning how to develop on android and understanding Googles OCR API that went with our app. Once these things were done, the app was relatively easy to develop. With ESE’s in the way, we didn’t have time to add extra features or improve the UI or improve the OCR detection. We hope this app has covered the main OOP concepts as well as creative enough to showcase our abilities as programmers.

# Brief Instructions on how to use the app

1. Open App
2. Click “Detect New Text
3. Place camera over desired text
4. Once textbox appears around text, tap relavent text to search
5. Choose the type of search you want to do

I have included the APK in the directory.

Thanks.