A Project Synopsis On

# Med-Tracker: Android app for medicine tracking

Submitted in partial fulfillment of the requirement of University of Mumbai for the Degree of

**Bachelor of Engineering**

In

**Information Technology**

Submitted By **Shrutika Deokar Sourabh Khandke Shubham Ture**

Supervisor

**Prof. Deepti Lawand**



**Department of Information Technology PILLAI COLLEGE OF ENGINEERING**

**New Panvel – 410 206 UNIVERSITY OF MUMBAI**

**Academic Year 2019 – 20**



DEPARTMENT OF INFORMATION TECHNOLOGY

Pillai College of Engineering New Panvel – 410 206

CERTIFICATE

This is to certify that the requirements for the BE Project entitled ‘**Med Tracker: Android app for medicine tracking**’ have been successfully completed by the following students:

### Name Roll No.

Shrutika Deokar B804

Sourabh Khandke B811

Shubham Ture B870

in partial fulfillment of Bachelor of Engineering of Mumbai University in the Department of Information Technology, Pillai College of Engineering, New Panvel – 410 206 during the Academic Year 2019 – 2020.

### Supervisor (Prof. Deepti Lawand)

**Head of Department Principal**

**Dr. Satishkumar L. Varma Dr. Sandeep M. Joshi**



DEPARTMENT OF INFORMATION TECHNOLOGY

Pillai College of Engineering New Panvel – 410 206

PROJECT APPROVAL FOR B.E

This Project entitled “Med Tracker: Android app for medicine tracking” by Shrutika Deokar, Sourabh Khandke and Shubham Ture are approved for the degree of Bachelor of Engineering in Information Technology.

Examiners:

1.

2.

Supervisors:

1.

2.

Chairman:

1.

Date:

Place:



DEPARTMENT OF INFORMATION TECHNOLOGY

Pillai College of Engineering New Panvel – 410 206

DECLARATION

We declare that this written submission for BE Project “**Med Tracker: App for medicine tracking**” represent our ideas in our own words and where others’ ideas or words have been included. We have adequately cited and referenced the original sources. We also declared that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by institute and also evoke penal action from the sources which have thus not been properly cited or from whom paper permission have not been taken when needed.

**Project Group Members:**

Shrutika Deokar & Sign:

Sourabh Khandke & Sign:

Shubham Ture & Sign:

Date:

Place:

# Table of Contents

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Abstract | | | | i | | |
| List of Figures | | | | ii | |
| List of Tables | | | | iii | |
| **1.** | Introduction | | | | 1 |
|  | **1.1** | Fundamentals | | 1 | |
|  | **1.2** | Objectives | | 1 | |
|  | **1.3** | Scope | | 1 | |
|  | **1.4** | Organization of the Project Report | | 2 | |
| **2.** | Literature Survey | | | 3 | |
|  | **2.1** | Introduction | | 3 | |
|  | **2.2** | Literature Review | | 4 | |
|  | **2.3** | Summary of Literature Survey | | 6 | |
| **3.** | Med-Tracker: Android app for medicine tracking | | | 8 | |
|  | **3.1** | Overview | | 8 | |
|  |  | 3.1.1 | Existing System | 9 | |
|  |  | 3.1.2 | Proposed System | 12 | |
|  | **3.2** | Implementation Details | | 14 | |
|  |  | 3.2.1 | Algorithm | 19 | |
|  |  | 3.2.2 | Activity Diagram | 20 | |
|  |  | 3.2.3 | Hardware and Software Details | 22 | |
| **4.** | Testing and Result Discussion | | | 24 | | |
|  | 4.1 | Input Details and Screen Shots | | 24 | | |
|  | 4.2 | Output Details and Screen Shots | | 31 | | |
| **5.** | Conclusion and Future Scope | | | 35 | | |
|  | 5.1 | Conclusion | | 35 | | |
|  | 5.2 | Future Scope | | 35 | | |
| Reference | | | | 36 | | |
| List of Published Papers / Certificates | | | | 37 | | |
| Acknowledgement | | | | 38 | | |

## Abstract

Medicine tracker is a native android application meant to aid the busy people with remembering their daily medications. It is designed for users who need a little help keeping track of their medical schedule. This application will allow the users to store medicine data and multiple alarms for those medicines. Reminders will occur once a day, but can also be set to occur multiple times a day and week. User can view the medicine currently in stock and get approximate date when the stocks will run out. Also, this system uses NLP algorithms that can extract data from text using deep learning techniques. In addition, the application will also store the history of medications taken. This will aid the user as well as doctor to keep track of their medication usage. This application will provide users with an option to search for local clinics and medicinal stores.

## List of Figures

|  |  |  |
| --- | --- | --- |
| Figure 2.2.1 | GCM Implementation | 5 |
| Figure 3.1.1 | Flow diagram of reminder process for medicine | 8 |
| Figure 3.1.2 | Android Architecture | 10 |
| Figure 3.1.3 | Text Extraction with google vision API | 11 |
| Figure 3.1.4 | Proposed System | 12 |
| Figure 3.1.5 | System functional module | 14 |
| Figure 3.1.6 | Google Cloud Vision API | 16 |
| Figure 3.1.7 | Google Cloud Vision API diagram 2 | 17 |
| Figure 3.1.8 | Text Extraction Diagram | 18 |
| Figure 3.2.1 | Activity Diagram of Reminder | 20 |
| Figure 3.2.2 | Activity Diagram of Text Extraction | 21 |
| Figure 4.1.1 | Home UI | 24 |
| Figure 4.1.2 | Login Details | 25 |
| Figure 4.1.3 | OTP Confirmation | 25 |
| Figure 4.1.4 | User Profile | 26 |
| Figure 4.1.5 | Medicine Search Bar | 27 |
| Figure 4.1.6 | Prescription added Confirmation | 28 |
| Figure 4.1.7 | Text Extraction | 28 |
| Figure 4.1.8 | Nearby Hospital Location | 29 |
| Figure 4.1.9 | Stored Contact List | 30 |
| Figure 4.1.10 | Medicine Stock Screen | 31 |
| Figure 4.1.11 | Stock Reminder | 31 |
| Figure 4.1.12 | Reminder Notification | 32 |
| Figure 4.1.13 | Low Stock Reminder | 32 |
| Figure 4.1.14 | SOS Emergency (nearby hospital) | 33 |
| Figure 4.1.15 | SOS Emergency Message | 34 |
|  | | |
|  |  |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Table 1 | Summary of Literature Survey | 6 |
| Table 2 | Hardware Details | 22 |
| Table 3 | Software Details | 22 |

**Chapter 1 Introduction**

### Fundamentals

As people are getting busier day by day, it’s very natural to forget about the medications provided by the doctor and to remember taking them on time. By this application we aim to make sure that people take their medications on time. This is especially helpful for the old people who have a natural tendency to forget normal things let alone medications prescribed for a certain time. To reduce the chance of missing these times this is our way to try to solve this problem.

### Objective

The objective of this work is as follows:

1.The main objective of this project is to remind people of their daily medication.

2.Remind user of low stocks.

3.Search for nearby medicals and hospitals.

4.Scan printed prescriptions to auto-fill data which will save user’s time and the data will be stored in proper tabular format.

### Scope

In today’s busy world it’s very easy for people to forget minor things and unfortunately the most common thing people forget it to take medications, especially if they are working. So, by implementing handy features into an application we aim to make sure people don’t forget to take their medicines as prescribed by doctors. The application will have features like searching for nearby medicals and clinics, maintaining medical history of the patient which can be accessed easily and also a stock reminder which will alert user when stocks run low (i.e. falls below a certain threshold).

1

### Outline

The report is organized as follows: The introduction is given in Chapter 1. It describes the fundamental terms used in this project. It persuades to study and understand the different techniques used in this work. This chapter also presents the outline of the objective of the report. The Chapter 2 describes the review of the relevant various techniques in the literature systems. It describes the pros and cons of each technique. The Chapter 3 presents the Theory and proposed work. It describes the major approaches used in this work. The societal and technical applications are mentioned in Chapter 4. The summary of the report is presented in Chapter 5.

# Chapter 2 Literature Survey

**2.1 Introduction**

Now a day’s technology innovation is updating the way sufferers are receiving care services. Smartphones aren’t best used for calling purpose but now may be used as an ensemble of embedded sensors that together allow new packages in widespread areas including healthcare, e-trade**,** homecare, healthcare, social networks, environmental tracking, transportation and protection. Today in healthcare system, the utilization of mobile devices is becoming an increasing number of useful. In addition, with cellular generation is gambling critical position in continual ailment management, empowering the elderly and pregnant ladies, to take remedy on the right time, extending services to underserved regions, and enhancing health situations and scientific system performance.

Mobile phone is effective and rich in function and also much less highly-priced due to advances made in various generation domains. Now a days Mobiles are now most effectively used for personal conversations and amusements, but it could also curiously utilize in numerous fitness and Wellness monitoring packages. Normally, patient’s health information facts had been recorded inside the shape of paper and stored. Now we can use mobile phones to remind the dosage of medication by using sending text message via GSM module.

Medication nonadherence is a common, complex, and costly problem that contributes to poor treatment outcomes and consumes health care resources. So, we are introducing an Android application whose objective is to remind the patient of their dosage timing through Alarm Ringing system so that they can stay fit and healthy. Through navigation they can search doctors and hospitals and contact details so that they can get easily get proper treatment on time. This application focuses on the people who forget to take medicines on time. It allows users to set an alarm along with the fields of date, time and medicine description.

### Literature Review

1. **R.Sunder Reddy, C.Niharika, Rohit Sharma, Bharat Kumar, “Intelligent Reminder System of Having Medicine for Chronic”, 2017.**

In the paper “Intelligent Reminder System of Having Medicine for Chronic Patients”, author HSU CHUN-LIANG have proposed a system who reminds the medicine on time. The whole system structure of the medicine box for certain individual patient, in which an intelligent monitoring system was designed and whole circuit was PCB-layout in the box. The system includes GSM communication module that collects patient’s information of having medicine and transmits it to the nursing center in the hospital, it also consists of LCD display to remind patient to have medicine properly.

### Mayuresh Waykole, Vatsalya Prakash, Himanshu Singh, “Ardu Med – Smart Medicine Reminder for Old People”, 2016.

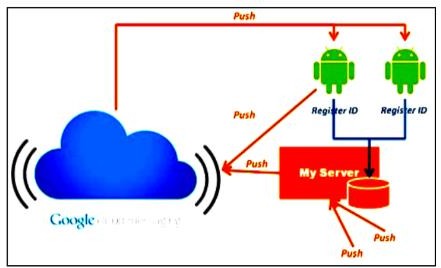
In the paper “ArduMed – Smart Medicine Reminder for Old People”, the author Mayuresh Waykole, Vatsalya Prakash, Himanshu Singh, Nalini N proposed a system which is a combination of physical and digital reminder. The system is divided into two categories: software and hardware. The software portion will do the reminder part of the task, which is to remind patients to take their medicines along with how many spoons or pills they are supposed to take. The reminder can be set in two ways: using the web application, or by using the mobile app. Both the applications require users to login, so that their medications can be synced with their calendar. Hardware Dependencies: The web hosting server should be up and available to host the web app. For the android app the mobile device should have a working internet connection for authentication and adding the events to calendar.

### Akshay Pandey, Rahul Kumar, Vinay Yadav, “Smartphone Based Medicine Intake Reminder Using GCM”, October 2016.

In the paper “SMARTPHONE BASED MEDICINE INTAKE REMINDER USING GCM” the

authors Akshay Pandey, Rahul Kumar, Vinay Yadav proposed with the GCM system. With the beginning of pervasive and mobile computing era, smartphones have become very common, and wearable devices are getting improvements and opportunities. A significant portion of the applications for these devices rely on remote servers on the cloud, and Google Cloud Messaging (GCM) is a very popular service for client/server communication for Android. GCM is a service which allows developers to send push notifications to Android devices from the server. GCM

handles the queuing of the messages and then delivers those messages to the destination applications on the devices. GCM is a free service provided by Google, and has no quotas. It is the automatic push messaging solution for the Android devices.



### Fig 2.2.1 GCM Implementation

1. **MD. Aby Sayeed Mondol, Ifat Afrin Emi, John Stankovic, “MedRem: An Interactive Medication Reminder and Tracking System on Wrist Devices”, September 2016.**

In the paper “MedRem: An Interactive Medication Reminder and Tracking System on Wrist Devices”, the authors Md Abu Sayeed Mondol, Ifat Afrin Emi, and John A. Stankovic have proposed a mobile health application called MedRem. They have used a system named voice command recognition. There are case when the touch screen is not feasible for exchanging information between the user and the system. MedRem addresses the limitation of the touch screen through enabling voice interactions. MedRem uses off-the-shelf speech recognition tools like the Android Speech Recognizer that are available on the smart wrist devices. The actual commands provided by the users, and the words generated by the speech recognition system often differ significantly, particularly for the non-native speakers.

### Summary of Literature Survey

A literature review is an objective, critical summary of published research literature relevant to a topic under consideration for research. The summary is presented here.

### Table 1 Summary of Literature Survey

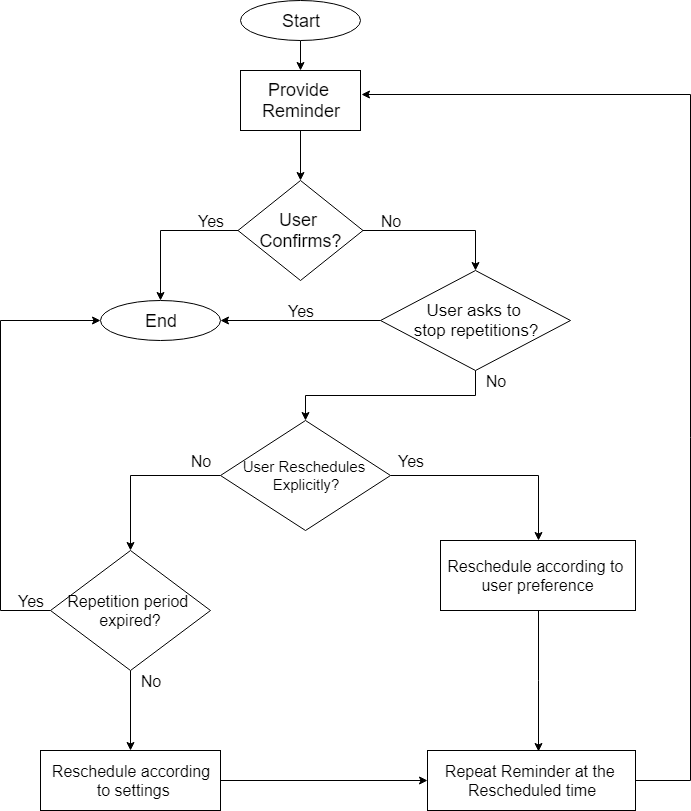
|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Techniques** | **Author & Year of**  **Publication** | **Advantages & Disadvantages** |
| 1 | GSM  Communication module | R.Sunder Reddy, C. Niharika, Rohit Sharma, Bharat Kumar., “An Intelligent Patient  Medicine Reminder  System,” February 2017 | Advantages: Easy to use.  Disadvantages: If the device gets damaged, the complete information is lost. |
| 2 | Google’s Oauth 2.0 | Mayuresh Waykole, Vatsalya Prakash, Himanshu Singh., “Ardu Med – Smart Medicine Reminder for Old People”., May 2016 | Advantages: Serves reliable reminders, has a good and easy to use user interface and supports a lot of features adhering to medicines.  Disadvantages: System is not up on alarms so if the user skips the alarm, the app won’t bug him continuously. If the data is not saved on a daily basis it is hard for the patient to maintain the history/progress towards their medications. |

|  |  |  |  |
| --- | --- | --- | --- |
| **SN** | **Techniques** | **Author & Year of**  **Publication** | **Advantages & Disadvantages** |
| 3 | Google Cloud Messaging | Akshay Pandey, Rahul Kumar, Vinay Yadav., “Smartphone Based Medicine Intake  Reminder Using GCM.”, October 2016 | Advantages: This system will provide the information about the medicine timings and quantity. The appointments which are scheduled with the doctor including the contact details including visiting time. This system focuses on improving the number of visits or revisits at healthcare appointments. The personal phone notifications and reminders are a very strong support tool for improving medication adherence strategies. |
|  |  |  | Disadvantages: If A patient misses or skips the dose or medication, this app will not continuously prompt the user to forcefully take their medications and there is no snooze button in this app so it will not remind the patient to take their medications after some interval of time. |
| 4 | Android Speech Recognizer | MD. Aby Sayeed Mondol, Ifat Afrin Emi, John Stankovic. “MedRem: An Interactive Medication Reminder and Tracking System on Wrist Devices.”, September 2016 | Advantages: State of the art speech recognition tools are combined with novel approaches that makes the system very usable and robust. It overcomes many of the limitations of the state of the art, this system would be very effective toward improving medication adherence.  Disadvantages: If the device malfunctions, the data might get lost and the user might not be able to track his progress for some time until he purchases a new device. |

**Chapter 3**

**Med-Tracker: Android app for medicine tracking**

* 1. **Overview**

Remembering to take medication is a prospective memory task and as such it relies on a set of cognitive processes responsible for completing actions at some point in the future. There are two main types of prospective memory tasks: time-based tasks that need to be completed at a specified time (e.g. take medication at 9:00) or after a set period of time has elapsed (e.g. take antibiotics every 8 hours); and event-based tasks, where the task is linked to an existing event and the environment in which it takes place, e.g. taking medication with breakfast. After considering all the types of tasks we implement appropriate features in our app. Below is the flow diagram of the working of reminders. Depending on how user interacts with the reminder, there are multiple outcomes for every scenario. For example, if user confirms taking of medication, then it immediately decreases the total amount of medications stored in the stocks. If user snoozes the alarm, the decrement is delayed along with the reminder which would pop up after rescheduled time (5 minutes). As for the default reminders, if user chooses to set all of them, that is three per day for a time-based task, the timing is 9 AM, 3 PM and 9.30 PM respectively.

### 

### Figure 3.1 Flow diagram of a Reminder Process for medicine

### Existing System Android Architecture

Android is a mobile operating system based on a modified version of the Linux Kernel and other open source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers knows as the Open Handset Alliance, with the main contributor and commercial marketer being Google. The core Android source is known as Android Open Source Project (AOSP), which is primarily licensed under the Apache License. This has allowed variants of Android to be developed on a range of other electronics, such as game consoles, digital cameras, PCs and other, each with a specialized user interface. Android has been the best-selling OS worldwide on smartphones since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthlies active users, the largest installed base of any operating system, and as of January 2020, the Goggle Play Store features over 2.9 million apps.

The Linux Kernel

The foundation of the Android platform is the Linux kernel. For example, the Android Runtime (ART) relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

At the bottom of the layers is Linux – Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

Using a Linux kernel allows Android to take advantage of key security features and allows device manufacturers to develop hardware drivers for a well-known kernel.

HAL

The hardware abstraction layer (HAL) provides standard interface hat expose device hardware capabilities to the higher-level Java API framework. The HAL consist of multiple library modules, each of which implements an interface for a specific type of hardware components, such as the camera or Bluetooth module. When a framework API makes a call to access device hardware, the Android system loads the library modules for that hardware components.

A hardware abstraction layer is included in many Oss to avoid modifying the OS kernel to run the program on computers with varying architecture. A PC may include the HAL in the OS kernel or in the form of device drivers that provide a consistent interface for application to interact with the hardware peripherals. A device abstracts the hardware of your product. For example, an audio module, can contain a primary audio device, a USB audio device, or a Bluetooth A2DP audio device.

A device is represented by the “hw\_device\_t” struct. Similar to module, each type of device defines a detailed version of the generic “hw\_device\_t” that contains function pointers for specific features of hardware.

The HAL provides the following benefits:

* + 1. Allowing applications to extract as much performance out of the hardware devices as possible.
    2. Enabling the OS to perform regardless of the hardware architecture.
    3. Enabling device drivers to provide direct access to each hardware device, which allows programs to be device-independent.
    4. Allowing software program to communicate with the hardware devices at a general level.
    5. Facilitating portability.

Android Runtime

For devices running Android version 5.0 (API level 2.1) or higher, each app runs in its own process and with its own instance of the Android Runtime (ART). ART is written to run multiple virtual machines on low-memory devices by executing DEX files, a bytecode format designed especially for Android that’s optimized for minimal memory footprint. Build toolchains, such as Jack, compile Java sources into DEX bytecode, which can run on the Android platform. Android runtime (ART) is the managed runtime used by the application and some system services on Android. ART and its predecessor Dalvik were originally created specifically for the Android project. ART as the runtime executes the Dalvik Executable format and DEX bytecode specifications. ART and Dalvik are compatible runtimes running DEX bytecode, so apps developed for Dalvik should work when running with ART. However, some techniques that work on Dalvik do not work on ART. For information about the most important issues, see verifying app behavior on the Android runtime (ART).

Some of the major features of ART include the following:

1. Ahead-of-time (AOT) and just-in-time (JIT) compilation ART introduces ahead-of-time (AOT) compilation, which can improve app performance. ART compiles apps using the on-device dex2oat tool. This utility accepts DEX files as input and generate a compiled app executable for the target device.
2. Optimized garbage collection (GC) Garbage collection (GC) can impair an app performance, resulting in choppy display, poor UI responsiveness, and other problems, it improves garbage collection in many ways.

-One GC pause instead of two.

-Parallelized processing during the remaining GC pause.

-Collector with lower total GC time for the special case of cleaning up recently-allocated, short-lived objects.

-Compacting GC to reduce background memory usage and fragmentation.

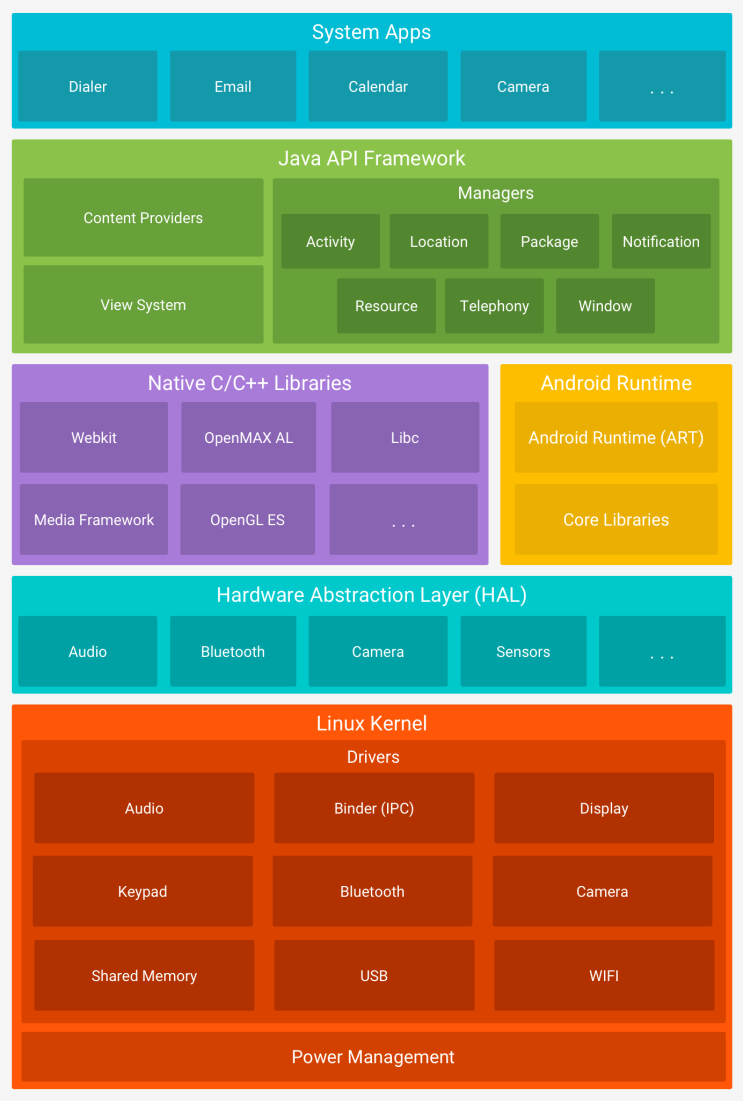
1. On Android 9 (API level 28) and higher, conversion of an app package Dalvik Executable format (DEX) files to more compact machine code.
2. Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields. Support for sampling profiler. Art adds support for a dedicated sampling profiler that does not have these limitations. This gives a more accurate view of app execution without significant slowdown. It provides with:

-Support for more debugging features.

-Improved diagnostic detail in exceptions and crash reports, ART gives you as much context and detail as possible when runtime exceptions occur. ART provides expanded exception details for “java. Lang.ClassCastException”, “java.lang.ClassNotFound”, “

Java.lang.NullPointerException”. Prior to Android version 5.0 (API level 21), Dalvik was the Android runtime. If your app runs well on ART, then it should work on Dalvik as well, but the reverse may not be true. Android also includes a set of core runtime libraries, that provide most of the functionality of the Java Programming language, including some Java 8 language features, that the Java API framework uses.

Native C/C++ Libraries

Many core Android system components and services, such as ART and HAL, are built from native code that require native libraries written in C and C++. The Native Development kit (NDK) is a set of tools that allow you to use C and C++ code with Android, and provides platform libraries you can use to manage native activities and success physical device components, such as sensors and touch input. The NDK may not be appropriate for most novice Android programmers who need to use only Java code and framework APIs to develop their apps. The Android platform provides Java framework APIs to expose the functionality of some of these native libraries to apps. For example, you can access OpenGL ES through the Android framework Java OpenGL API to add support for drawing and manipulating 2D and 3d graphics in your app. Using Android Studio 2.2 and higher, you can use the NDK to compile C and C++ code into a native library and package it into your APK using Gradle, the IDE integrated build system. Your Java code can then call functions in your native library through the Java Native Interface (JNI) framework. Android Studio also supports ndk-build due to the large number of existing projects that use the build toolkit. The below figure is a representation of android architecture which includes multiple layers. App that will be created will be on the topmost layer will communicate with JAVA APIs for certain permissions like location, Activity and Notification. Following this the app will communicate with Hardware Abstraction Layer (HAL) to pinpoint the location of the user to recommend the nearby medical stores and clinics. Finally, the results will be shown on the Display. [7]

### Figure 3.1.2 Existing Android Architecture [7]

### 

Java API Framework

The Application Framework is a set of services that collectively form the environment in which Android application run and are managed. The entire feature-set if the Android OS is available to you through APIs written in the Java language. This framework implements the concept that Android applications are constructed from reusable, interchangeable and replaceable components. This concept is taken a step further in that an application is also to publish its capabilities along with any corresponding data so that they can be found and reused by other application. These APIs from the building blocks, you need to create Android apps by simplifying the reuse of core, modular system components and services, which include the following:

A rich and extensible view system you can use to build an apps UI, including lists, grids, text boxes, buttons, and even embeddable web browser.

A Resource Manager, providing access to non-code resources such as localized string, graphics, and layout fields.

The Android framework includes the following key services:

1. Activity Manager – Controls all aspect of the application lifecycle and activity stack.

2. Content Providers – Allows applications to publish and share data with other applications.

3. Resource Manager – Provides access to non-code embedded resources such as string, color settings, and user interface layout.

4. Notification Manager – Allows applications to display alerts and notifications to the user.

5. View System – An extensible set of views used to create applications user interface.

6. Package Manager – The system by which applications are able to find out information’s about other applications currently installed on the device.

7. Telephony Manager – Provides information to the application about the telephony services available on the devices such as status and subscriber information.

8. Location Manager – Provides access to the location =services allowing an application to receive updates about location changes.

A Notification Manager that enables all apps to display custom alerts in the status bar.

An Activity Manager that manages the lifecycle of apps and provides a common navigation back stack. Content Providers that enables apps to access data from other apps, such as the Contact app, or to share their own data.

System Apps

Android comes with a set of core apps for email, SMS messaging, calendars, internet browsing,

contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install. So, a third-party app can become the users default web browser, SMS messenger, or even the default keyboard. Applications are the bridge that clears the jargon between the core working of the android architecture and user. Applications are designed as user friendly as possible to make sure the user can navigate through the app as developer intended. These comprise both the native applications provided with the particular Android implementation (for example web browser and email applications) and the third-party applications installed by the user after purchasing the device. The system apps function both as apps for users and to provide key capabilities that developers can access from their own app. For example, if your app would like to deliver an SMS message, you don’t need to build that functionality yourself- you can instead invoke whichever SMS app is already installed to deliver a message to the recipient you specify.

Text Extraction

The Vision API can detect and transcribe text from PDF and TIFF files stored in Cloud Storage.

Document text detection from PDF and TIFF must be requested using ‘files:asyncBatchAnnotate’ function, which performs an offline (asynchronous) request and provides its status using the operations resources. Output from a PDF/TIFF request is written to a JSON file created in the specified Cloud Storage bucket. Any image that is scanned by user is uploaded to firebase cloud storage which is backed up by firebase hosting. Cloud functions make use of libraries to make sure the image is in proper format as an input to the vision API. The output from the vision API is then stored in Firebase Database.



### Fig 3.1.3 Text Extraction using Google Cloud Vision. [10]

**3.1.2 Proposed System**

The application that we built starts off with intro slider which highlights the key features of the application, leading user to login page via the last slide. After which user will be asked to login via his/her google account. After logging into the app initiates three main services which further lead to other functions of the application.

### Login Services

### In general computer usage, login is the procedure used to access to an operating system or application, usually in a remote computer. Almost always a login requires that the user have a user ID and a password. Often, the user ID must conform to a limited length such as eight characters and the password must contain at least one digit and not match a natural language word. The user ID can be freely known and is visible when entered at a keyboard or other input device. The password must be kept secret (and is not displayed as it is entered). Some web sites require users to register in order to use the site; registered users can the enter the site by logging in. Here we used Login with google because application is developed on Android so users will have at least one google email which would save users time to login and fill in all the details as they will be extracted directly via Google Account of User.

**OCR**

Optical Character Recognition or Optical Character Reader is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo (for example the text on signs and billboards in a landscape photo) or from subtitle text superimposed on an image (for example from a television broadcast) [17]. Widely used as a form of data entry from printed paper data records – whether passport documents, invoices, bank statement, computerized receipts, business cards, mail, printouts of static-data, or any suitable documentation – it is a common method of digitizing printed texts so that they can be electronically edited, searched, stored more compactly, displayed on-line, and used in machine processes such as cognitive computing, machine translation, (extracted) text to speech, key data and text mining. OCR is a field of research in pattern recognition, artificial intelligence and computer vision. Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for most fonts are now common, and with support for a variety of digital image file format inputs. Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

The Text Recognizer segments text into blocks, lines, and words:

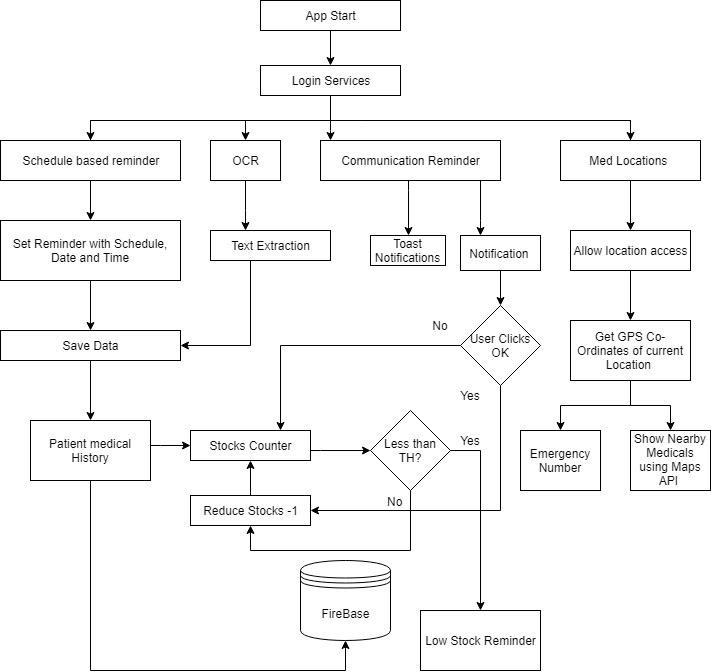
1. A Block is a contiguous set of text lines, such as a paragraph or column,

2. A Line is a contiguous set of words on the same vertical axis, and a Line is a contiguous set of

words on the same vertical axis, and

3. A Word is a contiguous set of alphanumeric characters on the same vertical axis. Below is a

pictorial representation of these.



### Fig 3.1.4 Proposed System

### GPS

The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The system provides critical capabilities to military, civil and commercial users around the world. It is maintained by he a United States government and is freely accessible to anyone with a GPS receiver.[18]

### Google Map API

### The Maps API returns helpful data about places and locations. It is called by JavaScript. It does two major things: It also lets us request an address from Google (we give them a physical location (latitude and longitude), they turn it into an address); We can do the reverse, by giving Google an address, and requesting a physical location (latitude and longitude). We can also get information about businesses, landmarks, restaurants, etc. from the Google Maps & quot;Places&quot; library. Google Maps helps autocomplete our users&#39; typing, in the search box. This makes it easier and more intuitive to type a real location that will have good results. The way we implemented this is in a way that user would have least user interaction. The locate me icon will show the uses nearby medicines based on where he is situated.

### Firebase

### Firebase is a Backend-as-a-Service — BaaS. Firebase frees developers to focus crafting fantastic

### user experiences. You don’t need to manage servers. You don’t need to write APIs. Firebase is your server, your API and your datastore, all written so generically that you can modify it to suit most needs. You’ll occasionally need to use other bits of the Google Cloud for your advanced applications. Firebase can’t be everything to everybody. Firebase first product was the Firebase Real-time Database, an API that synchronizes application data across iOS, Android, and Web devices, and stores it on Firebase cloud. The product assists software developers in building real- time, collaborative applications.

### Firebase provides various services:

### Google Analytics – It is a cost- free app measurement solution that provides insights on app usage and user engagement.

### Firebase Cloud Messaging – Firebase Cloud Messaging (FCM) is a cross-platform solution for messages and notifications for Android, iOS, and web applications, which as of 2016 can be used at no cost.

### Firebase Authentication – Firebase Authentication is a service that can authenticate users using inly client-side code. It supports social login providers Facebook, GitHub, Twitter and Google as well as other service providers like Google Play Games, Apple, Yahoo and Microsoft.

### Firebase Realtime Database – Firebase provides a real-time database and back-end as a service. The service provides application developers an API that allows application data to be synchronized across client’s and stored on Firebase cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript framework such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real-time database can secure their data by using the company server-enforced security rules.

### Firebase Storage – Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality, to be used for storing images, audio, video, or other user-generated content. It is backed by Google Cloud Storage.

### Firebase Performance – Firebase Performance provides insights into an app performance and the latencies the app users experience. [8]

### SOS

### SOS is a Morse code distress signal, used internationally, that was originally established for maritime use. In formal notation SOS is written with an over score line, to indicate that the Morse code equivalents for the individual letters of “SOS” are transmitted as an unbroken sequence of three dots / three dashes / three dots, with no spaces between the letters. In International Morse Code three dots form the letter “S” and three dashes make the letter “O”, so “S O S” became a common way to remember the order of the dots and dashes. Although SOS officially is just a distinctive Morse code sequence that is not an abbreviation for anything, in popular usage it is associated with phrases such as “Save Our Souls” and “Save Our Ship”. Moreover, due to its high-profile use in emergencies, the phrase “SOS” has entered general usage to informally indicate a crisis or the need for action. SOS originated in German government maritime radio regulations adopted effective 1 April 1905. In modern terminology, SOS is a Morse “procedural signal” or “prosing”, used as a start-of-message mark for transmissions requesting assistance when loss of life or catastrophic loss of property is imminent. Other prefixes are used for mechanical breakdowns, requests for medical assistance, and a relayed distress signal originally sent by another station. SOS remained the maritime radio distress signal until 1999, when it was replaced by the Global Maritime Distress and Safety System. SOS is still recognized as a standard distress signal that may be used with any signaling method. It has been used as a visual distress signal, consisting of three short/three long/three short flashes of light, such as from a survival mirror. In some cases, the individual letters “S O S” have been spelled out, for example, stamped in a snowbank or formed out of logs on a beach. The fact that “S O S” can be read right side up as well as upside down is an advantage for visual recognition. [19]

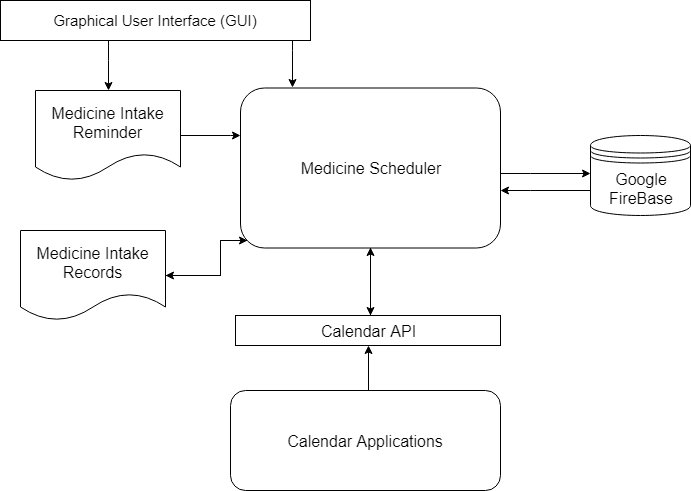
### 3.2 Implementation Details.

The figure 3.3 gives complete knowledge about the whole system. This diagram showcases the various API’s used.

GUI – This is the overall UI of the application. User will interact with this part of the application

only. Reminders – After user logs into the application, he/she will have options to set reminders,

manually input the available stocks etc.



### Figure 3.1.5 System Functional Module

### Create Activity ()

### It is the first activity which will provoke the process of setting the reminder, which will have 2 ways of notifying the users. Either in Toasts or Dialog boxes. Toast is a small view that contains a quick message for the user. This message does not persist since it is available for only a few seconds at most. The toast never collects focus. We will not use a toast for reminding the user. Instead, we use a toast to notify the user when user activity has been saved, so that the user knows something happened. Actually, the final not-so-popular method to grab a user’s attention is to open a dialog window that can instantaneously steal focus from the user’s presently running app and direct it to a dialog window. While this may indeed work as a method for grabbing the user’s attention, the user may get irritated since their app is stealing focus (possibly on a constant basis if the user has a lot of reminders) from his current actions in another application. We will be using the Notification Manager class to handle the alarms for the Task Reminder application.

### Notification Manager

### The Notification Manager class is used to notify a user that an event or events have taken place. These events can be placed in the status bar that is located at the top of the screen. The notification

### items can contain various views and are identified by the icons that we provide. The user can slide

### the screen down to view the notification. The basic implementation of the reminder needs to have these 4 basic things. Title, Body, Reminder Date, Reminder Time. Title being the name of the medicine or name of the reminder (If user has multiple alarms set). Body will contain description of the medicine. We create this file in res/layout directory with appropriate name. This is responsible for all of the alerts, events sent from android system to the screen whether it be from system application or third-party installed application.

### Views

### The Views are important factor in applications since many devices have variable screen resolutions. So, to avoid this will use scroll view as our parent view which creates a scroll bar and allows the view to be scrolled when the contents of the view are too big for the screen device.

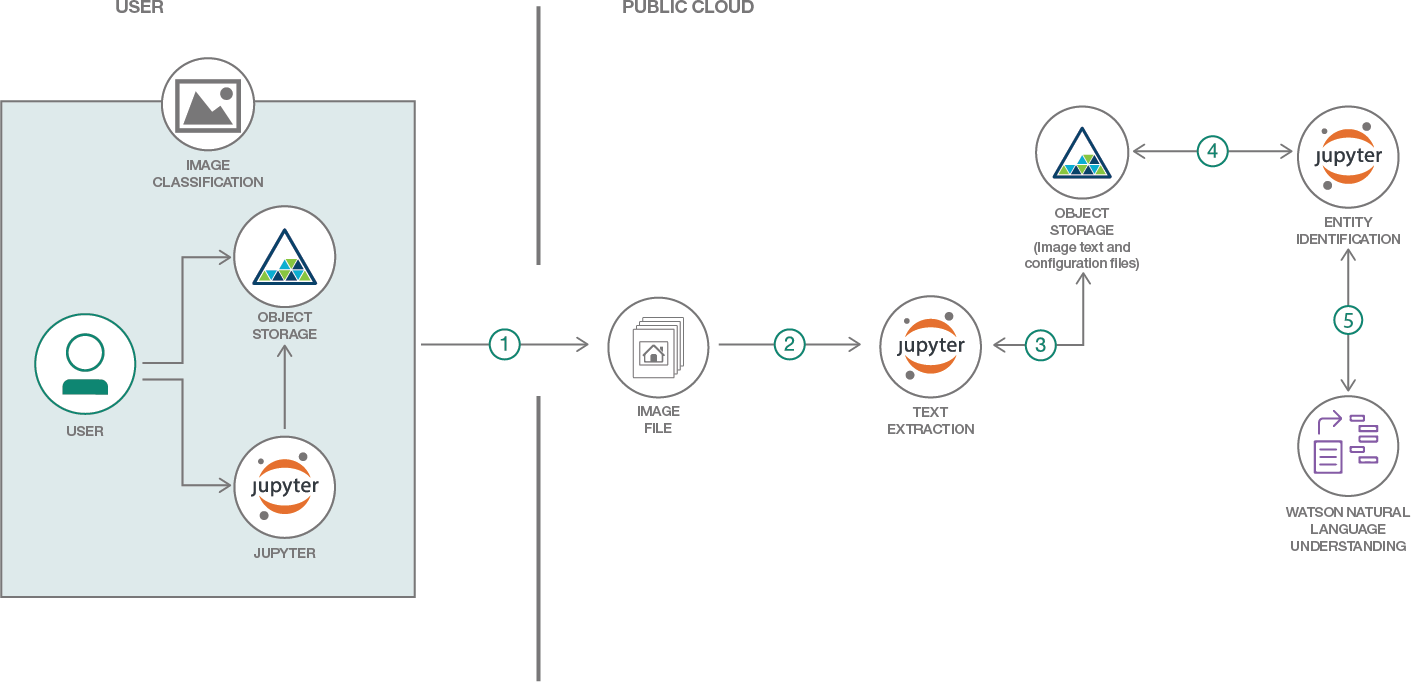
**For Extraction of text from image using tesseract OCR engine and Google Cloud Vision API** State-of-the-art NLP algorithms can extract data from text using deep learning techniques such as healthcare-specific word embeddings, named entity recognition models, and entity resolution models. Such Algorithms use trained models to find relevant words in a body of text. Named entity recognition models work by searching for specific names and grouping these into predefined categories. OCR (Optical Character Recognition) is an algorithm used to extract the text from the given image.

### How does an OCR work in device?

### When a page is scanned, it is typically stored in bitmap file in Tagged image file (TIF) format. When the image is displayed on the screen, we read it. But, to computer it is a series of black and white dots. When the scanner reads the image of the document, it converts the dark elements (text and graphics) on the page to a bitmap, which is a matrix of B&W dots. OCR software reads the bitmap created and averages out the on and off pixels on the page. (affecting the white spaces). In the process, software tries to match each character through pixel-by-pixel comparison to character templates in the memory. Characters that remains unrecognized are processed through Feature Extraction. Finally, the document can be saved either in ASCII file format or other.

### Google Cloud Vision API

Google Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API. It quickly classifies images into thousands of categories, detects individual objects and faces within images, and finds and reads printed words contained within images. The tool first performs a layout analysis on the image to segment the location of the text. After the general location is detected, the OCR module then performs a text recognition analysis on the specified location to generate the text.

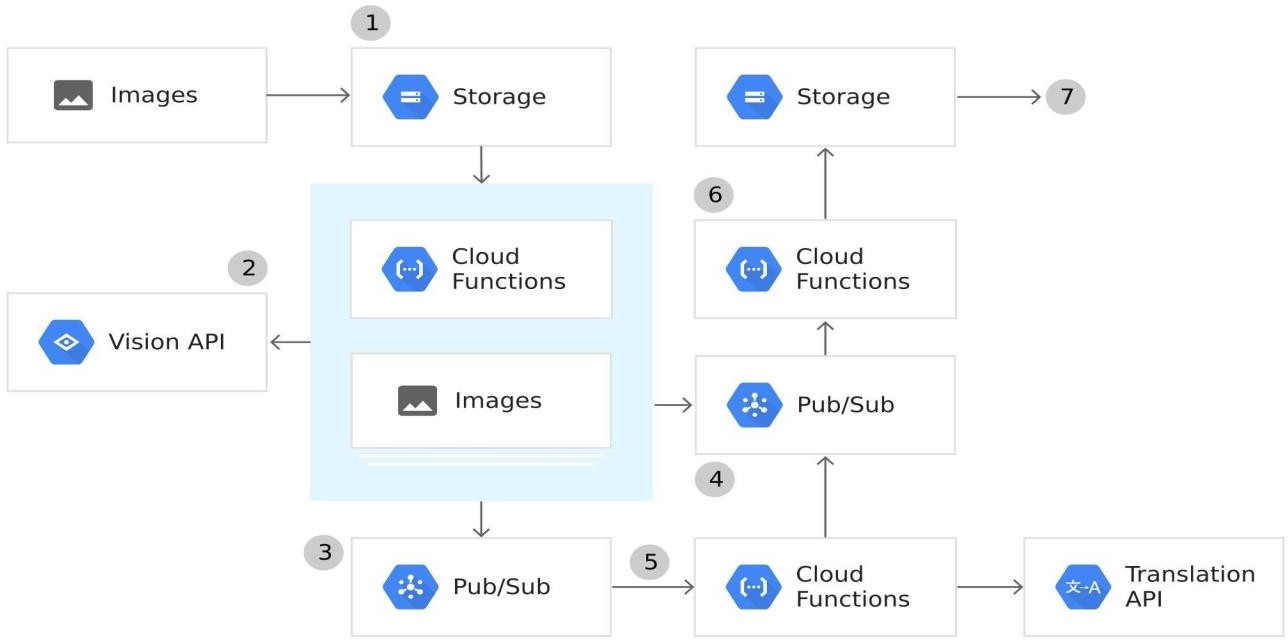


### Fig 3.1.6 Google Vision API [11]

The Vision API can detect and extract text from images. There are two annotation features that support optical character recognition:

TEXT\_DETECTION detects and extracts text from any image. For example, a photograph might contain a street sign or traffic sign. The JSON includes the entire extracted string, as well as individual words, and their bounding boxes.

DOCUMENT\_TEXT\_DETECTION also extracts text from an image, but the response is optimized for dense text and documents. The JSON includes page, block, paragraph, word. [15]



**Figure 3.1.7 Google Vision API diagram 2 [14]**

### Stocks reminder:

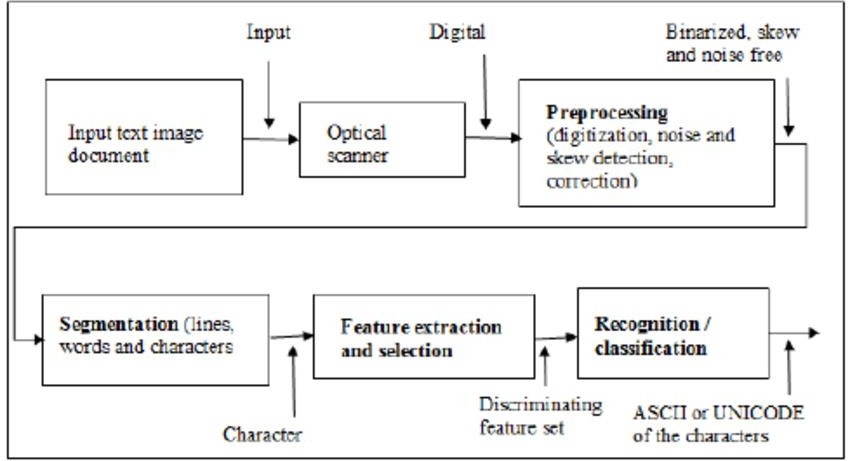
### This system has a stock reminder function which keeps track of the user’s medicine stocks. This module allows user to keep a track of your products inventory & expiry date from back end panel. User can manage inventory status & product expiry data by receiving timely updates on registered email or phone no. When the user’s medicine course is about to complete, there will be notification provoked by the app to remind the user about the remaining stock of the medicine that has left. The count of the medicine will decrease as the user will take the medicine. If the user wants to continue the medicine as prescribed by the doctor, they can set the reminder according to their needs.

### This helps the system to keep track of the patient’s history which is useful for the doctor to perform further analysis. The stock Reminder can also be represented in a graphical manner or statistical manner according to the user’s choice. Optical Scanner-OCR (optical character recognition) is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document.

### Segmentation- In computer vision, image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries in images. [20]

### Word Segmentation- The spacing between the words is used for word segmentation. Generally, in English script, spacing between the words is greater than the spacing between the characters in a word. The spacing between the words is found by taking the Vertical Projection Profile (VPP) of an input text line. Vertical Projection profile is the sum of ON pixels along every column of the image. [21]

Feature Extraction- Feature extraction starts from an initial set of measured data and builds derived values intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is a dimensionality reduction process, where an initial set of raw variables is reduced to more manageable groups for processing, while still accurately and completely describing the original data set. [22]



### Fig: 3.1.8 Text Extraction Diagram [12]

* + 1. **Algorithm**

Algorithm: Geometric

Input : Noise free character image of size *MxN*

Output: Feature vector of size 86

Begin

1.Divide the image into 9 (3x3) equal zones

2.For each zone determine

No. of horizontal lines, vertical lines, right diagonal lines, Left diagonal lines.

Normalized Length of all the above values (4 values)

Normalized area of the skeleton

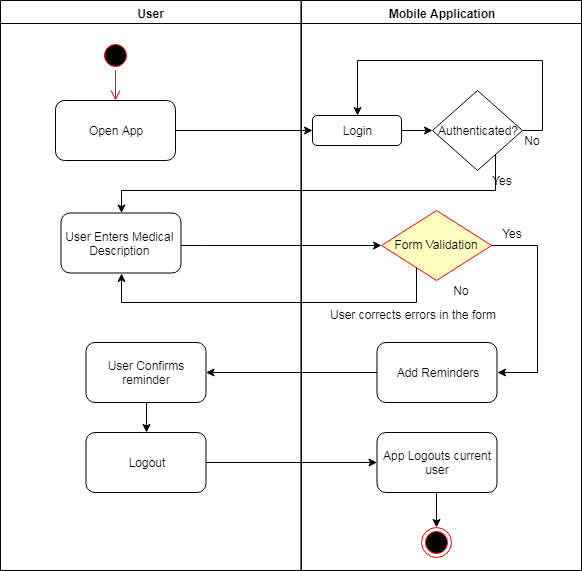
3.Compute the global parameters of Euler number, eccentricity, extent, convex and concave curve orientations.

4.Return extracted features i.e. Feature Vector. End;

In general Google Cloud Vision API support many types of image analysis techniques. Be it optical character recognition, landmark detection or a simple logo detection. Cloud vision API does it very accurately. Also, the best part about this API is that; its cross platform and is available via API access, which results in very light weight applications. Also, since this technology is comparatively new, there is a lot of scope for improvement. And while that’s happening behind the scenes we don’t need to worry about the changing code-base as basically it’s just an API call. Moving on let’s have a look at the supported Android image recognition techniques.

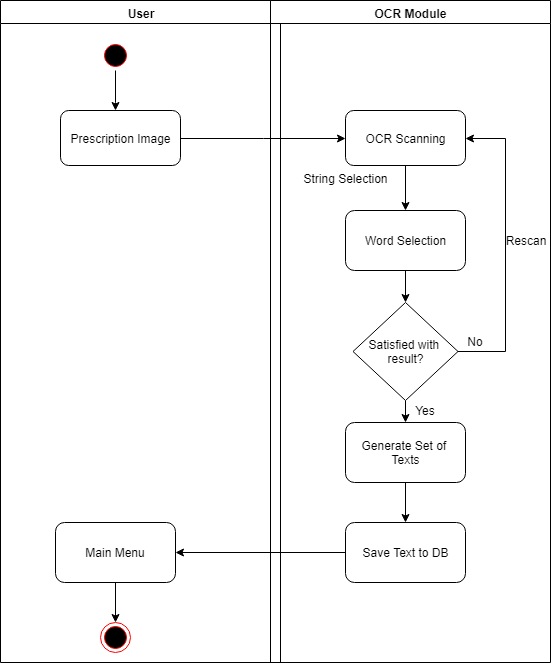
### Activity Diagram

This diagram shows the general process involved in setting a simple reminder in android. User first logs in using his mobile number, if user authentication is successful then the user will be logged into the application where user can enter medical description into form after which user will be asked to set a reminder and after confirming the reminder will be set.



### Figure 3.2.1 Activity diagram of reminder

First the prescription(printed) is uploaded for the process of extracting text from image. Then in a further step the prescription in scanned by an OCR scanner, OCR scanner recognizes the image and start reading and analyzing its pattern for the process of text extraction. After the selection of text is done the further step is to validate the result that we have got in the form of text. Decision box will check if the text is accurate then proceed or else rescan the image for accurate data. The text will be extracted from the image and save it to the database.



### Figure 3.2.2 Activity Diagram for Text Extraction

* + 1. **Hardware and Software Specification**

The experiment setup is carried out on a computer system which has different hardware and software specifications as given in Table 3.2. and Table 3.3respectively.

Table 3.2 Hardware details

|  |  |
| --- | --- |
| Processor | Intel Xeon Platinum /Intel Core i7 /Ryzen 5 1600 |
| HDD | Western Digital Black / Seagate 180 GB |
| RAM | 8 GB GDDR4 |

Table 3.3 Software details

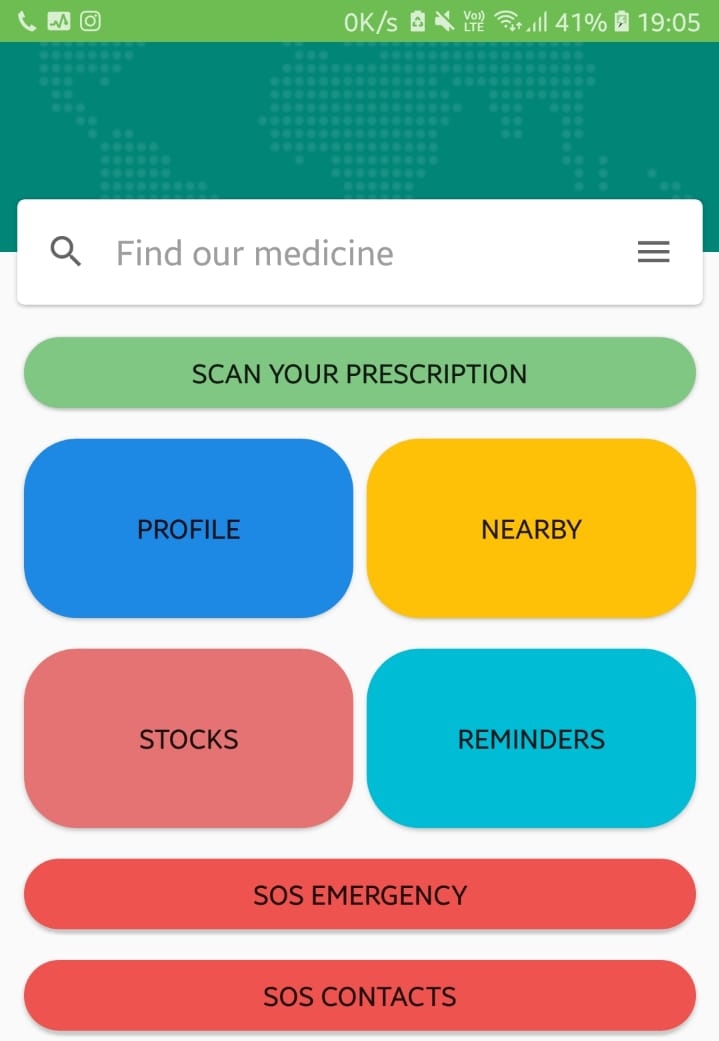
|  |  |
| --- | --- |
| Operating System | Windows 10/8/7 |
| Programming Language | Java, Android Studio |
| Database | Google Firebase API |

**Chapter 4**

**Testing and Result Discussion**

### Input Details and Screen Shots

In our application the input details consist of a user creating an account to login, user confirmation, add the number of medications prescribed by the doctor and scanning the prescription, nearby hospitals and search for the medicine.

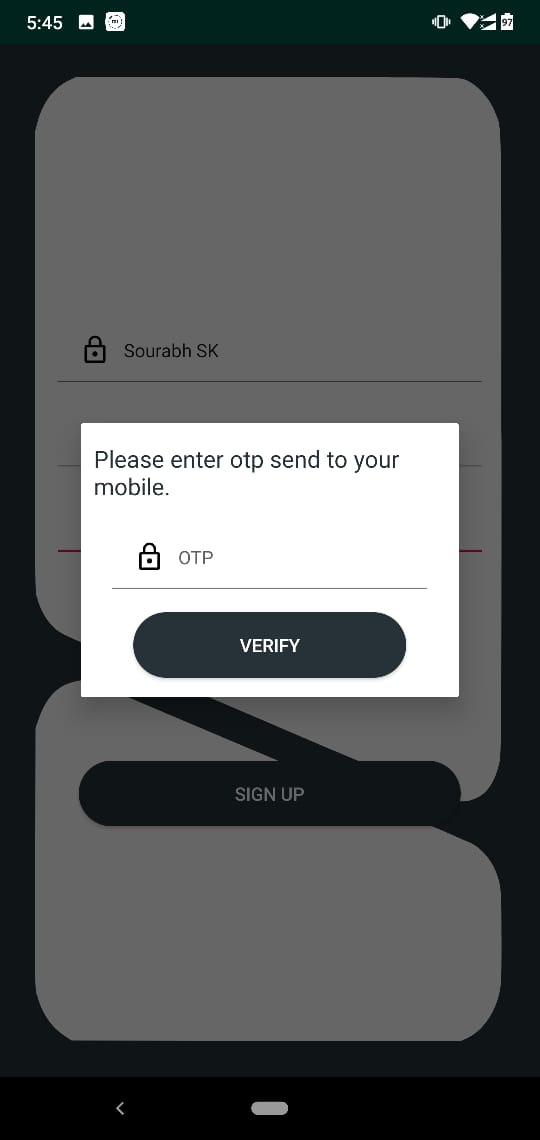


### Fig 4.1.1 Home UI

Fig.4.1.1 depicts the front end of the application. The front end consists of 8 tabs namely a search box, Scan your Prescription, Profile, Nearby, Stocks, Reminder, SOS Emergency, SOS Contact.

**Login Details and Confirmation**

A login is a set of credentials used to authenticate a user. Most often, these consists of a username and password. However, a login may include other information, such as PIN number, email-id, passcode or passphrase. Our application requires a username, email-id and a phone number. Once the user provides with his/her phone number an OTP (One Time Password) is sent to the registered phone number to confirm the registration of the user.

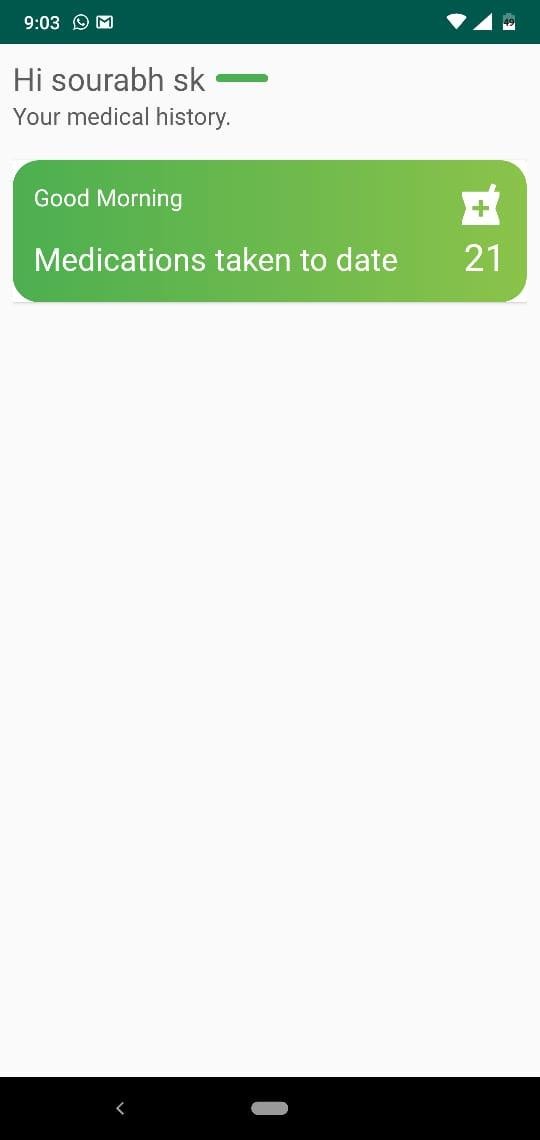




**Fig. 4.1.2 Login Details Fig. 4.1.3 OTP Confirmation**

**Profile**

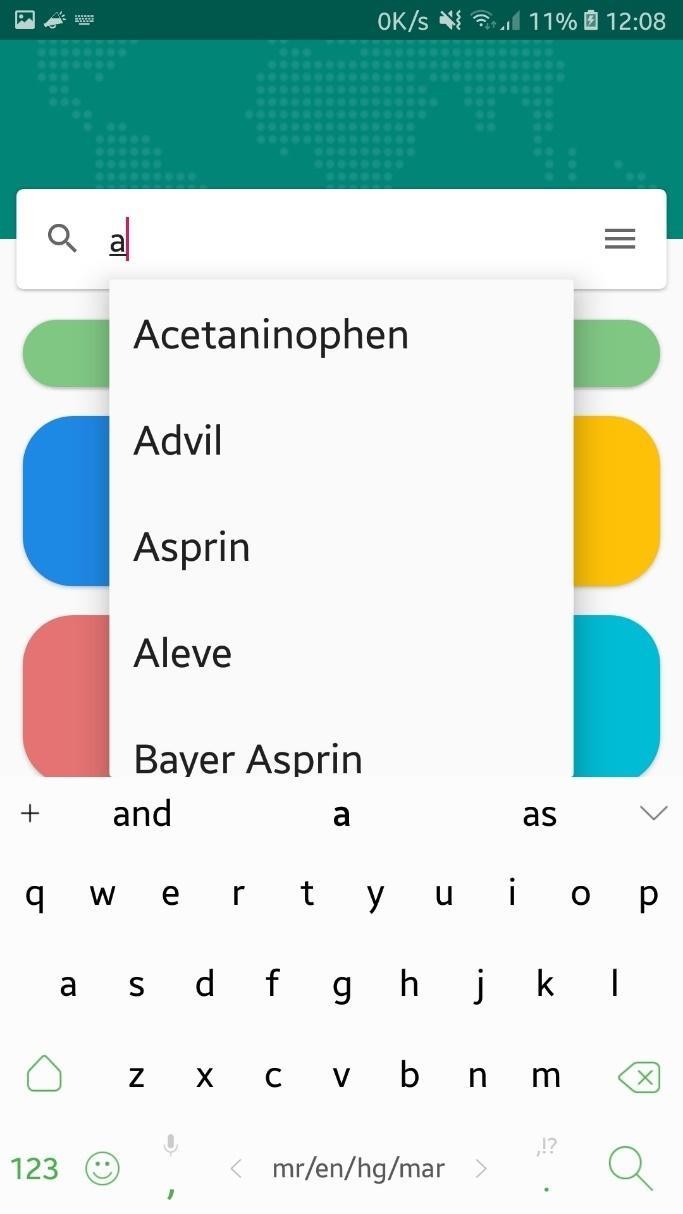
A user profile is a visual display of personal data associated with a specific user, or a customized desktop environment. A profile refers therefore to the explicit digital representation of a person’s identity. A user profile can also be considered as the computer representation of a user model. A user model is a (data) structure that is used to capture certain characteristics about an individual user, and the process of obtaining the user profile is called user modelling or profiling. A profile can be used to store the description of the characteristics of a person. This information can be exploited by systems taking into account the person’s characteristics and preferences. Here in this application users name, age, date of registration and number of medications are stored.



### Fig 4.1.4 User Profile

**Search Bar**

The search bar is a UI component that’s controlled by the Android system. When activated by the user, the search dialog appears at the top of the activity. The Android system controls all events in the search dialog. When the user submits a query, the system delivers the query to the activity that you specify to handle searches. The dialog can also provide search suggestions while the user types. In this application the user will search for his required medicine.



### Fig 4.1.5 Medicine Search Bar

**Scan your Prescription**

The Text Recognition API recognizes text in any Latin-based language. Text Recognition can automate tedious data entry for credit cards, receipts, and business cards, as well as help, organize photos, translate documents, or increase accessibility. Here the user can scan the prescription given to him by the doctor, which makes it easy for the user to feed data rather than manually typing it.

### 

### 

**4.1.6 Prescription added Confirmation 4.1.7 Text Extraction**

**Nearby**

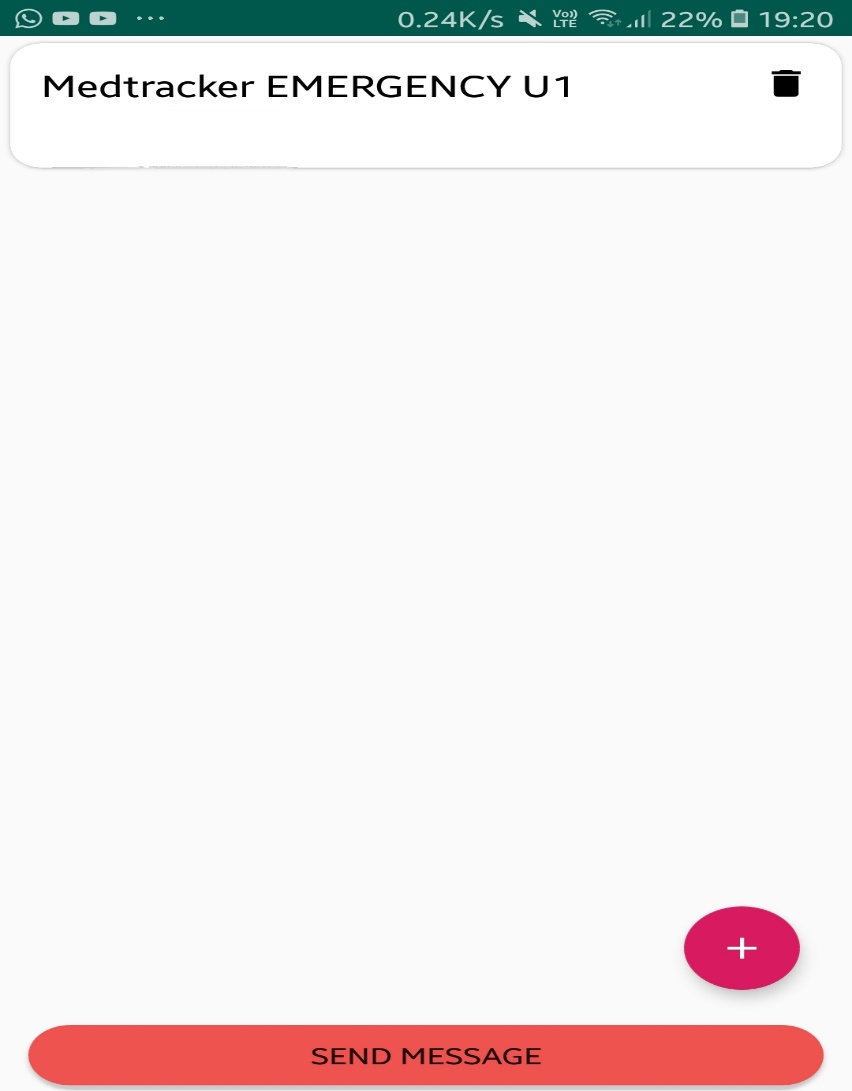
This tab will show the nearby medical store and hospital to user wherever the user is located. If the user is travelling from one location to another then this option will help the user to locate the nearby hospitals and medical store.



### Fig 4.1.8 Nearby Hospital Location

**SOS Contact List**

To use the SOS Contact tab the user first needs to store the contact of the person with whom he wants to contact during an emergency. The user can add N number of contacts he wants to add and if the user wishes to delete any of the stored contact the user can delete as per his needs.

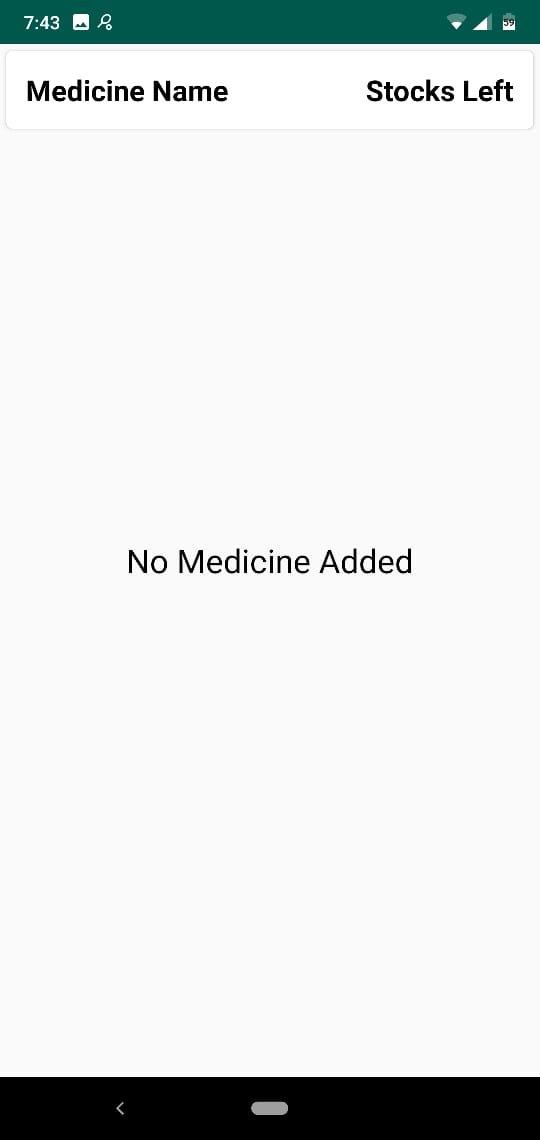
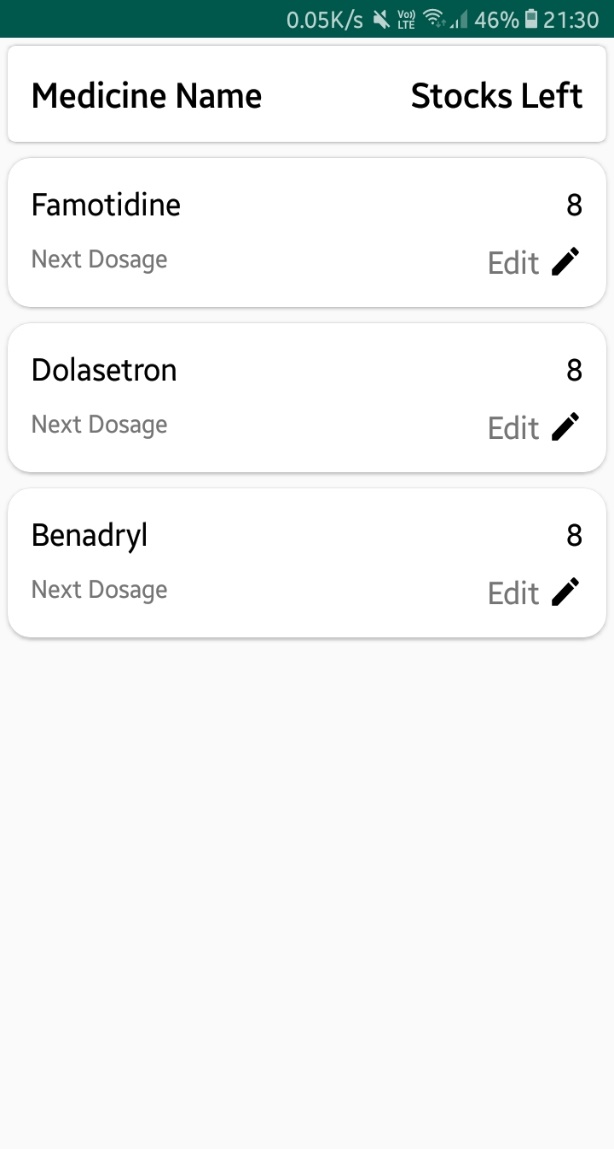
**Fig. 4.1.9 Stored Contact List**

* 1. **Output Details and Screen Shots**

The output details consist of the remining stocks, it gives a reminder to the users for medicine intake, SOS Emergency and SOS Message.

**Stocks**

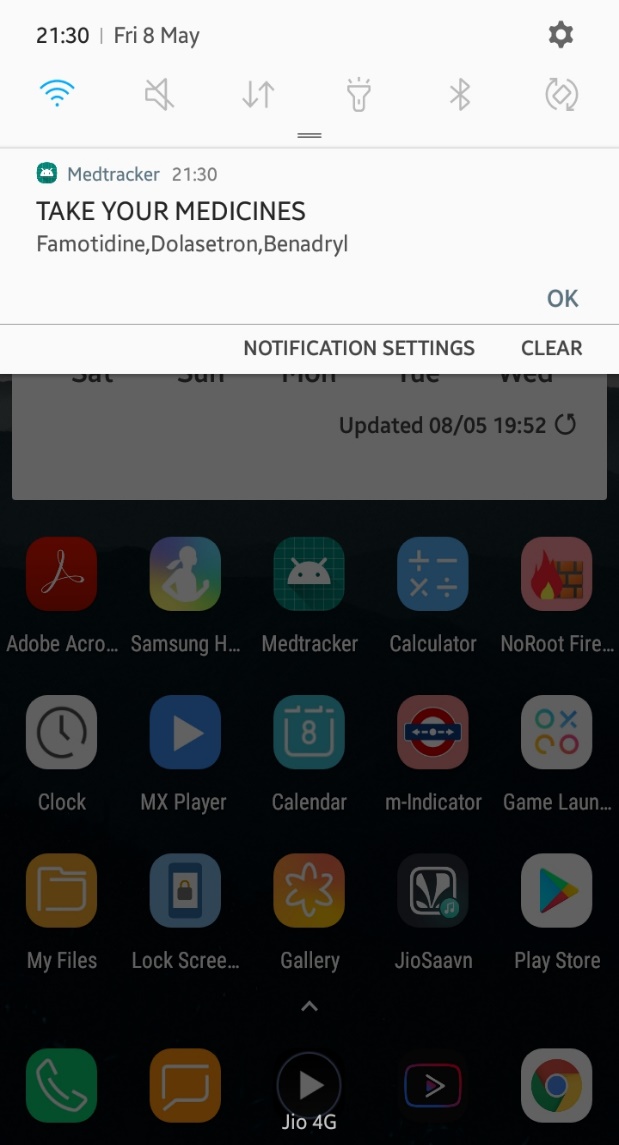
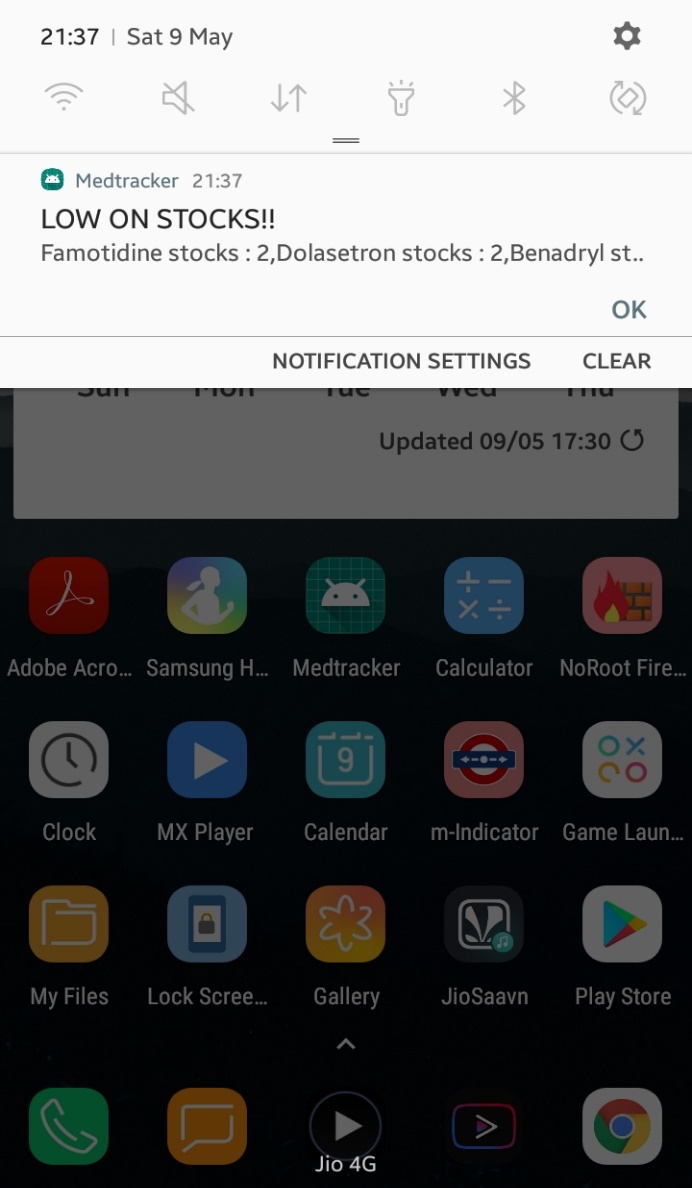
The stock tab is of great help to the user, resulting in storing the number of medicine available and does remind the user when it is running out of stock.



### Fig 4.1.10 Medicine Stock Screen Fig 4.1.11 Stock Reminder

**Reminder**

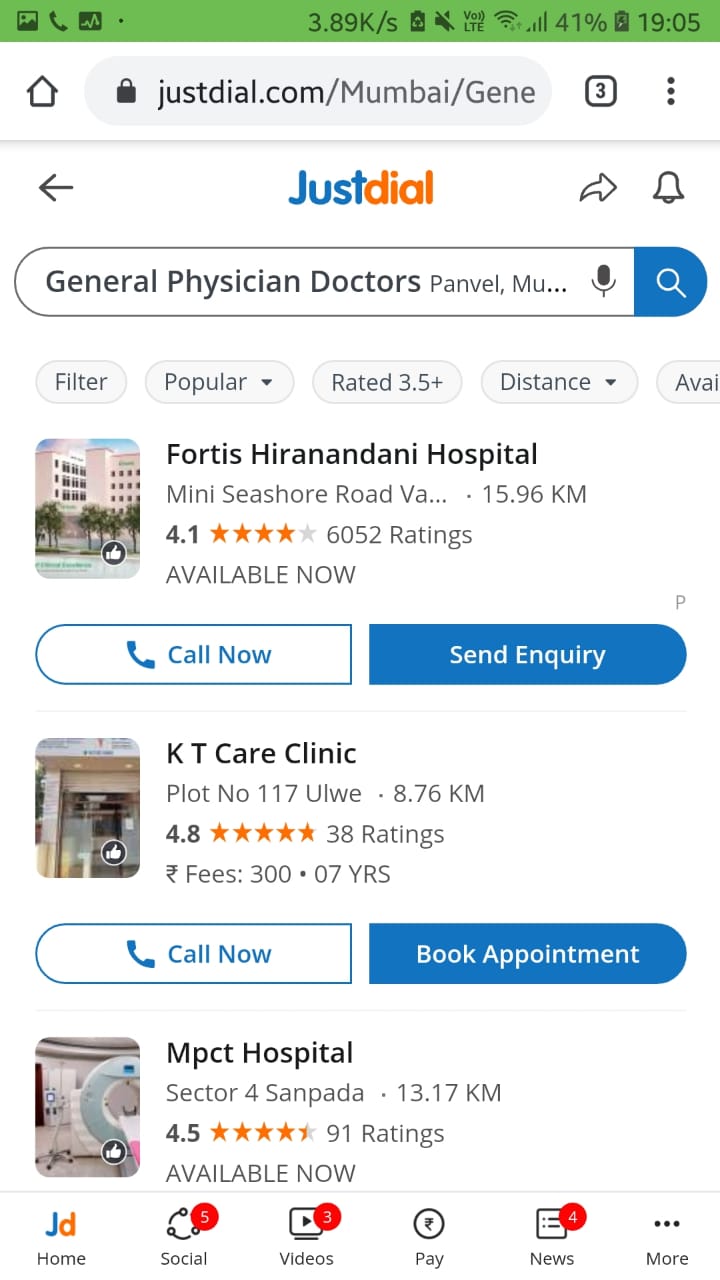
A reminder tab is very useful for the people who are very busy with their work schedule and also helpful for the senior citizens who often forget to take their medicine on time. Reminder tab is basically an alarm system which will ring according to the user’s medication time. The user has to manually fill the medication as prescribed by the doctor. There’s another reminder which will remind the user when the medicine is low on stock so as the user gets to know that he is running out of stocks.



**Fig 4.1.12 Reminder Notification Fig 4.1.13 Low Stock Reminder**

**SOS Emergency**

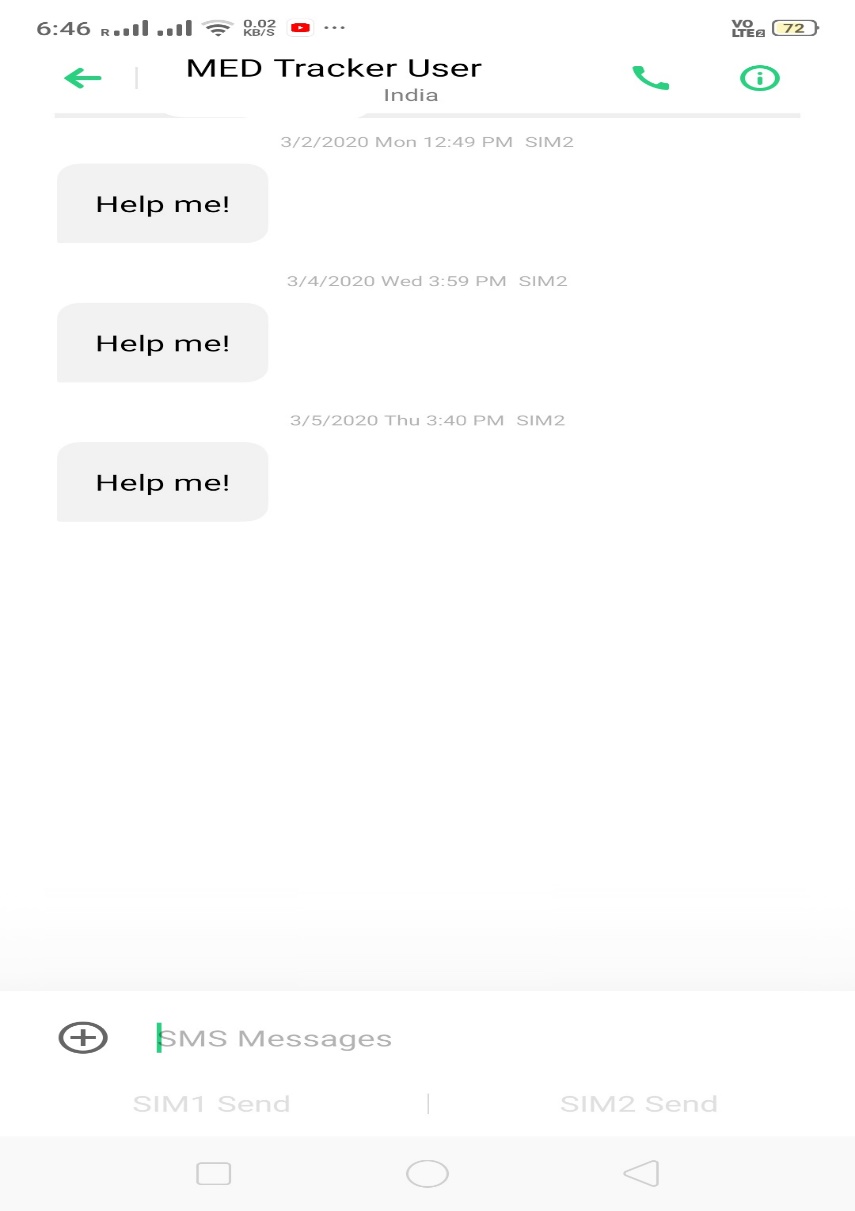
SOS Emergency is of great help when the user is in medical emergency. We have linked our application with Justdial. Justdial is a company that provides local search for different services in India over the phone and online. On clicking the SOS Emergency tab, the users are provided with the list of nearby hospitals and medical stores. As the application is linked with Justdial the users are provided with every information regarding the hospital including its contact number and address.

******

**Fig. 4.1.14 SOS Emergency (nearby hospitals)**

**SOS Messages**

On clicking the SOS Contact tab, a text message “HELP ME” will be sent to the number of stored contacts. This notifies the other person of the user being in some emergency and requires urgent help.



**Fig. 4.1.15 SOS Emergency Messages**

**Chapter 5**

**Summary and Future Scope**

* 1. **Summary**

Currently there are many medication reminder systems which are operable manually. Due to manual work, the available system becomes more time consuming. So, in the given work, an attempt has been made to implement fully automatic medication reminder system based on handwritten character recognition. This is an innovative system for any user mostly targeting the aged population as a medical helper. The user or anyone on behalf of the user can enter the medicine reminders such as tablet name, quantity and when it should be taken with a reminder. The reminder helps the user to look into the details of tablets that he or she has to take. The System also helps the user to click picture of medicines or make use of OCR where the system will make use of the camera and print the text on the screen and the user has to tap the text to save them. The System uses Mobile Vision API for Implementing OCR. This can also be a medium of not typing the name of a medicine while using OCR instead saving a lot of time. This system can be useful for a person where he has many tablets to intake. There are many advantages of using this application such as the system can be operated only if the user is a registered thus securing data, the system will vibrate when it will give reminder notification as well gives a detail about the reminder. A mobile-phone based automated medication reminder system shows promise in improving medication adherence.

**5.2 Future Scope**

In today’s market there are many medicine intake reminder systems and mobile applications available. These systems and mobile applications have been made on various different platforms. Talking about the future scope we can implement the application with an OCR Scanner which can extract the number of times the medicine has to be taken whether in the morning, afternoon or at night and also the details of the medicine such as the manufacturing date, expiry date, manufacturing company.

## References

[1]R. Sunder Reddy, C. Niharika, Rohit Sharma, Bharat Kumar. “An Intelligent Patient Medicine Reminder System”, 2017.

[2] Bhadane Ashwini, Bhuse Ishwari, Pawar Pallavi., “An Android based Medication Reminder System based on OCR using ANN”, August 2017.

[3] Mayuresh Waykole, Vatsalya Prakash, Himanshu Singh., “Ardu Med – Smart Medicine Reminder for Old People”, May 2016.

[4] Akshay Pandey, Rahul Kumar, Vinay Yadav., “Smartphone Based Medicine Intake Reminder Using GCM”, October 2016.

[5] MD. Aby Sayeed Mondol, Ifat Afrin Emi, John Stankovic., “MedRem: An Interactive Medication Reminder and Tracking System on Wrist Devices.”, September 2016.

[6] Shivani Sharma, Katyayni Tyagi, Pooja Shishodia., “Salubrity- A medicine reminder application using Android”, May 2018.

[7] Android Developers. (2015, October.) Platform Architecture [Online]. Available: <https://developer.android.com/guide/platform>. [Accessed April 1, 2019].

## [8] Waqas, (2019, December). How to use Firebase Database in Android. Available: <https://firebasetutorials.com/firebase-database/>. [Accessed April , 2020]

[9] Swarup Kumar Raj, Kakoli De. “chapter 17 Electronic Resource Management and Digitization”, IGI Global, 2020.

[10] Sara Robinson. (2017, June.) Exploring the Cloud Vision API. Available: <https://medium.com/@srobtweets/exploring-the-cloud-vision-api-1af9bcf080b8>. [Accessed May 5, 2020]

[11] Bartosz, Powerful Image Analysis with Google Cloud Vision and Python. Available: <https://www.smashingmagazine.com/2019/01/powerful-image-analysis-google-cloud-vision-python/>.  [Accessed April 28, 2020]

[12] Swamy Das, Ram Mohan. (2015, September) Evaluation of Neural Based Feature Extraction Methods for Printed Telugu OCR System. Available: <https://www.researchgate.net/figure/Block-diagram-of-a-typical-OCR-System_fig1_289675117>. [Accessed May 3, 2020]

[13] Hongwei Zhang, Changsong Liu, Cheng Yang (2011, September). An Improved Scene Text Extraction Method Using Conditional Random Field and Optical Character Recognition.

Available: <https://ieeexplore.ieee.org/abstract/document/6065403/authors#authors>. [Accessed May1, 2020]

[14] Google Cloud Documentations. (2020, May) Optical Character Recognition (OCR) Tutorial.  Available: <https://cloud.google.com/functions/docs/tutorials/ocr?hl=zh-tw>. [Accessed May 3, 2020]

[15] Ashraff Hathibelagal. (2017, June). How to Use the Google Cloud Vision API in Android Apps. Available: <https://code.tutsplus.com/tutorials/how-to-use-the-google-cloud-vision-api-in-android-apps--cms-29009>  [Accessed April 20, 2020]

[16] Leonardo Pirro (2018, February). Quick Look at Google Cloud Vision API on Android. Available:<https://medium.com/iquii/quick-look-at-google-cloud-vision-api-on-android-66d87d08943>.  [Accessed April 20, 2020]

[17] Nandand. OCR Implementation in Android. (2018, February). Available: <https://www.talentica.com/blogs/ocr-implementation-in-android/>. [Accessed April 10, 2020].

[18] Civil Simplified (2016, June). Global Positioning System. Available: <https://www.civilsimplified.com/resources>. [Accessed May 18, 2020].

[19] Wikipedia (2020, April). SOS. Available: <https://en.wikipedia.org/wiki/SOS>. [Accessed May 18, 2020].

[20] Gupta Mehul (2014). Text-based image Segmentation Methodology. Available: <https://www.sciencedirect.com/science/article/pii/S2212017314000954>. [Accessed May 18, 2020].

[21] Archana A. Shinde. “Text Pre-Processing and Text Segmentation for OCR”, January 2012.

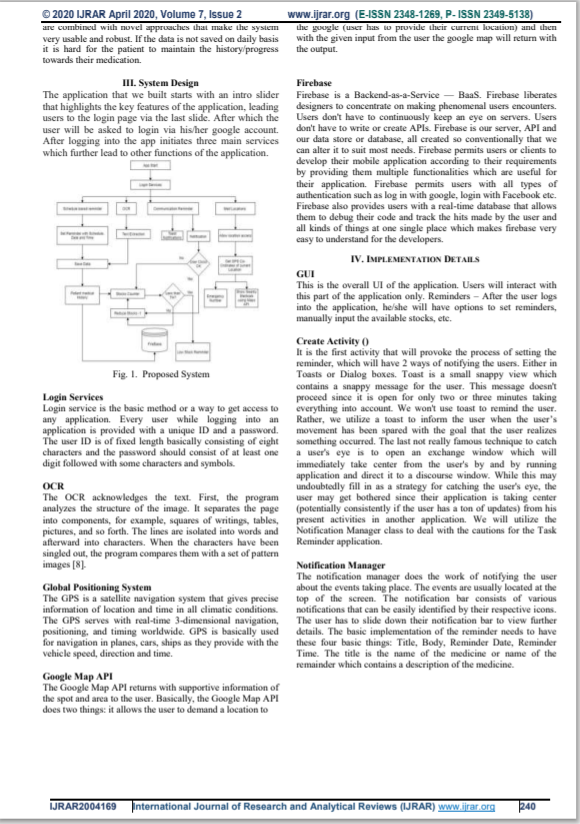
[22] Wikipedia (2020, February). Feature Extraction. Available: <https://en.wikipedia.org/wiki/Feature_extraction>. [Accessed May 18, 2020].

[23] Micro Pyramid. (2017, October) Extract text with OCR for all image types in python using tesseract. Available: [https://medium.com/@MicroPyramid/extract-text-with-ocr-for-all-image-](https://medium.com/%40MicroPyramid/extract-text-with-ocr-for-all-image-) types-in-python-using-pytesseract-ec3c53e5fc3a. [Accessed March 30, 2019].

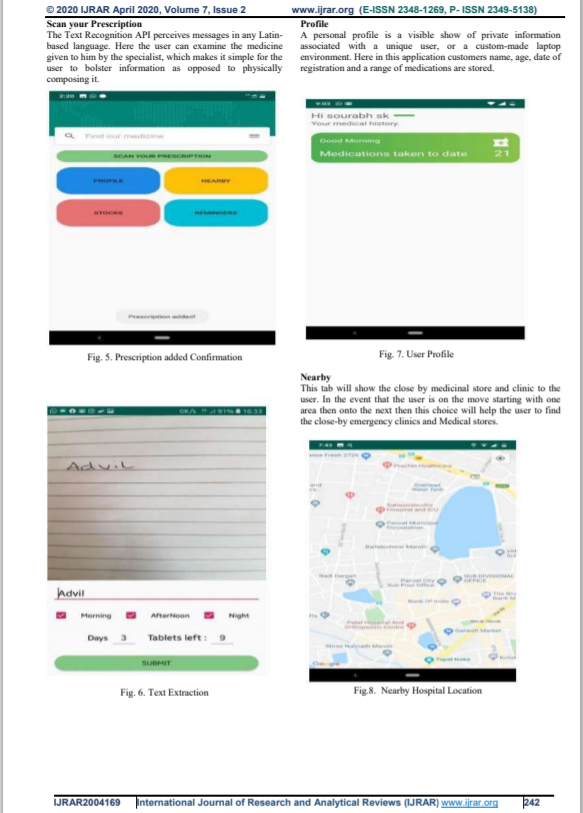
**Published Papers / Certificates**

Deokar, S., Khandke, S., Ture, S. and Lawand, D., 2020. MED-TRACKER: ANDROID APP FOR MEDICINE. *International Journal of Research and Analytical Reviews (IJRAR)*, 7(2), pp.239-245.









## 

## 

## 

## 

## 

## 



## Acknowledgement

We would like to thank our mentor, Prof. Deepti Lawand for her guidance and unwavering support throughout the project and the semester. We would like to thank our HOD, Dr. Satishkumar Varma for his encouragement and motivation to learn and implement projects of sorts. Lastly, we would like to thank our principal, Dr. Sandeep Joshi for providing us opportunities to explore our domain and for motivating us to do better.

Shrutika Deokar

Sourabh Khandke

Shubham Ture