

# Medicine Information Mobile Application Using Tablet Image Analysis Using Android Studio

M.Sakthiumamaheswari<sup>1</sup>, Parimala Suresh Congovi<sup>2</sup>, Madheswari Kanmani<sup>3</sup>

*Department of Computer science Engineering, SSN College of Engineering, Chennai, India.*

## Abstract

The mobile application aims in providing medicine information using image analysis of the tablet. Pharmacies have wide range of medicine for a specific cause. It becomes very important for common people to buy the right medicine and be aware of its usage. The developed mobile application is developed in order to provide people with the necessary information. The mobile application takes the input as either the name of the medicine or an image of the backside of the tablet. The image is processed by the built-in library present in android studio and the text is extracted from the image. The complete information of that particular medicine is provided to the user. The developed work includes the following features:

1. Provide alternate medicine information.
2. Provide with text to voice conversion.
3. Provide daily reminder.

**Keywords:** Android Studio; text extraction; text-to-voice conversion; Smartphone;

## INTRODUCTION

Medicine for a very long time played an important role in saving human life. With the increasing population, the number of diseases and the medicines to cure it has increased exponentially. Even for common illness like headache, cough, etc. many medicine are present in the pharmacy. In India, people buy medicine from the pharmacy most of the time without any doctor prescription. In such case, there are chances that people

buy medicine without much knowledge of its usage. Apart from it, people would not have knowledge of the other alternate medicine present (even if they are more effective with cheaper cost) . Our mobile application helps to overcome these difficulties with flexible, user-friendly, understandable GUI to provide people with the necessary information regarding the medicines they eat.

There also arises situation where people tend to forget their pills. This breaks the cycle of intake. In order to avoid this, our application has a reminder facility to notify the user when its time to take pills.

Our application also has the feature of text-to-voice conversion. This helps senior citizens as they may have difficulty reading the information from the screen of the smartphone.

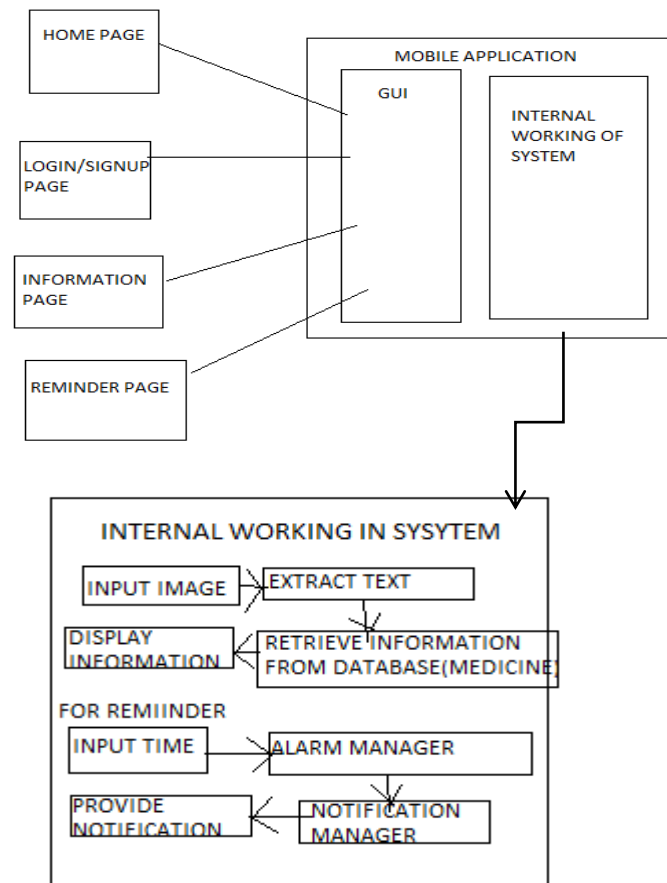
## SURVEY OF EXISITING WORK

1MG mobile application has been developed with some of these features namely, providing medicine information based on user input, proving reminders. Many such applications are available in Google Play Store. None of this application has a flexibility of including image analysis of tablet and provides text-to-voice conversion.

## DEVELOPED WORK

The system (mobile application) includes a GUI to display or get information from the user. The information obtained from the user is used to make further decision.

Following is the architecture diagram of the mobile application.



**WORKING:**

- The application consists of three forms.
- In the first form, the user either logs in if he is an existing user or signs up if he is a new user.
- If the user is a new user, he enters his details namely Name, Email id, Password and Mobile Number.
- The application then sends a One Time Password[1] to the registered mobile number with the help of FIREBASE to verify the user’s mobile number.
- After the verification of the user’s mobile number, the user is navigated to a tabbed pane consisting of three tabs namely HOME, MEDICINE DETAILS and REMINDER.
- If the user is an existing user, he enters his username and password, and is authenticated.
- On successful authentication, he is navigated to the HOME page.
- The authentication is done with the help of the user details stored in the SQLite Database[2].
- In the medicine details tab, the user is provided with a choice to input the medicine name.
- He can either enter the medicine name or upload the medicine’s picture (backside of the tablet).
- If the user uploads the image then the name of the medicine is extracted from the image by using text extraction from image interface provided by the Google vision[3].
- On clicking the submit button, the user is provided with the medicine details like the medicine cost, Description about the medicine, Side effects of the medicine and its Alternatives.
- On clicking on the alternate medicine ,details about the alternate medicine is displayed.
- The user is also provided with the audio about the medicine which is rendered to the user with the help of Text-To-Speech class in Studio.
- In the last tab, user enters the time (in 24 hour format)if he wants to be reminded to take up his medicine.
- The notification is provided to the user at that time using the Alarm Manager and Notification Manager [4] classes in studio.

## RESULT



**Figure 1:** Signup Page

If the user does not exist, then he/she is shown with this page. It contains text fields to input the name, phone number, password and mail id. On clicking the **submit** button, an OTP is generated which automatically checks the phone number and redirects to the UI shown in Fig.3

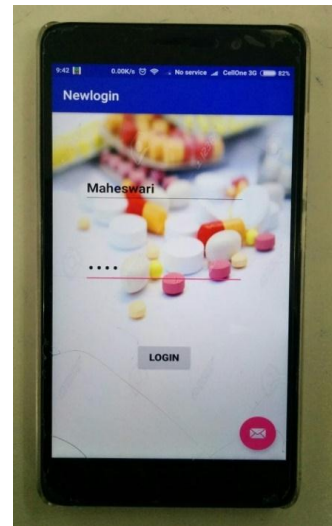
Code for OTP implementation is:

```
//Importing necessary headers
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthProvider;

//Instantiating the objects
PhoneAuthProvider.ForceResendingToken mResendToken;
PhoneAuthProvider.OnVerificationStateChangedCallbacks mCallbacks;

//To obtain phone number and send OTP to it.
PhoneAuthProvider.getInstance().
verifyPhoneNumber(
    "+91"+MobileNumber.getText().toString(), // Phone number to verify
    60, // Timeout duration
    TimeUnit.SECONDS, // Unit of timeout
    Signup.this, // Activity (for callback binding)
    mCallbacks); // onVerificationStateChangedCallbacks
```

```
//Verifies the OTP input with the generated OTP
otpbutton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        PhoneAuthCredential credential =
        PhoneAuthProvider.getCredential(mVerificationId,
        otpText.getText().toString());
        signInWithPhoneAuthCredential(credential);
    }
});
```



**Figure 2:** Login Page

If the user already exists, he moves to the Login page shown. It contains fields to input the username and password. The given credentials are verified by the app (string comparison with the information provided by the user and the information in the database) and redirects to the UI shown in Fig.3

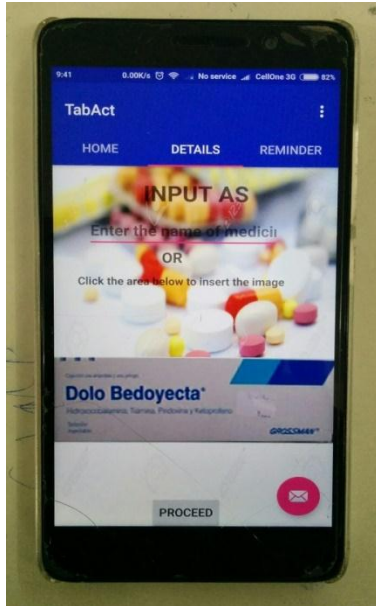


**Figure 3:** Tabbed Page  
**(HOME, DETAILS, REMINDER)**

This is a Tabbed Activity containing the tabs

- HOME
- DETAILS
- REMINDER

Tabbed Activity helps user to move through the different tabs easily.



**Figure 4:** Input from user as image/text

The input from the user is obtained either as a text or as an image. If the input is an image, an intent to the gallery is made. The code to intent to gallery is:

```
Intent gallery = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
if(text==null)
startActivityForResult(gallery, RESULT_LOAD_IMAGE);
```

Text recognition is done on the loaded image to extract the text. This is done with the help of the TextRecogniserclass present in android.

```
TextRecognizertextRecognizer = new
TextRecognizer.Builder(getActivity()).build();
```

It uses a Frame object to build the image using Bitmap. Bitmap helps in image manipulation.

```
Frame frame = new Frame.Builder().setBitmap(bitmap).build();
Bitmap bitmap = BitmapFactory.decodeStream(imageStream);
```

SparseArray object is used to store the values detected from the text recognizer.

```
SparseArray<TextBlock> items = textRecognizer.detect(frame);
```

String builder object is created to obtain the values from the SparseFloatArray and set the respective text fields.



**Figure 5:** Medicine Details Page along with text-to-voice conversion

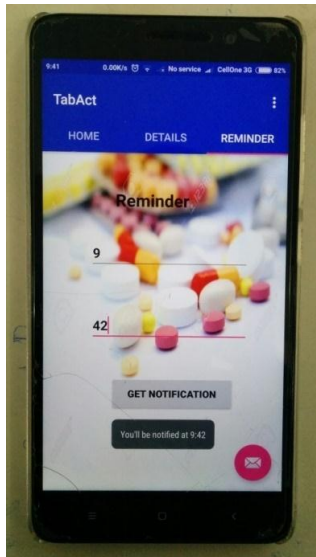
The details of the required medicines is displayed to the user. If the user requires an assistance to get an audio feed of the same, text-to-voice conversion is made when

**GET AUDIO**

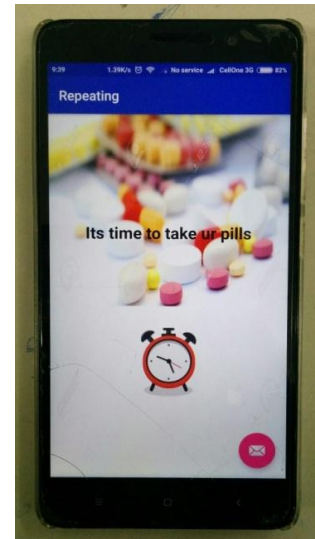
Button is pressed.

```
t1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
String toSpeak = ed1.getText().toString();
Toast.makeText(getApplicationContext(),
toSpeak, Toast.LENGTH_SHORT).show();
t1.speak(toSpeak, TextToSpeech.QUEUE_FLUSH, null);
}
});
```

t1 refers to the id of the textview



**Figure 6:** Reminder Page



**Figure:** Reminder

It consists of text fields for obtaining the Time to provide reminder. Its a 24 hr clock. The values are set in the Calendar object which is used by the AlarmManager to provide alarm.

*AlarmManager am=*

```
(AlarmManager) getActivity().getSystemService(Context.ALARM_SERVICE);
```

```
am.setRepeating(AlarmManager.RTC_WAKEUP,cal.getTimeInMillis(),AlarmManager.INTERVAL
```

```
_DAY,pi);
```

*//pi refers to the pending intent object .it also requires an intent object( intent)*

*PendingIntent pi=*

```
PendingIntent.getBroadcast(getActivity(),100,intent,PendingIntent.FLAG_UPDATE_CURRENT);
```

*//This intents to Notification\_receiver class to generate Notification to the user*

```
Intent intent =new Intent(getActivity(),Notification_receiver.class);
```

```
Calendar cal= Calendar.getInstance();
```

```
cal.set(Calendar.HOUR_OF_DAY,hr);
```

```
cal.set(Calendar.MINUTE,min1);
```

```
cal.set(Calendar.SECOND,0);
```

Notification is built using NotificationManager class:

*NotificationManager*

```
nm=(NotificationManager)context.getSystemService(Context.NOTIFICATION_SERVICE);
```

```
NotificationCompat.Builder builder=new
```

```
NotificationCompat.Builder(context)
```

```
.setContentIntent(pi)
```

```
.setSmallIcon(android.R.drawable.arrow_up_float)
```

```
.setTitle("Medico")
```

```
.setText("Hurry
```

```
Up!!!").setAutoCancel(true)
```

```
.setSound(Settings.System.DEFAULT_NOTIFICATION_URI)
```

```
.setVibrate(vibrate);
```

*//notify displays the notification to the user*

```
nm.notify(100,builder.build());
```

The above images show the complete functionalities of the mobile application.

## CONCLUSION

The developed mobile application provides the facilities of displaying the medicine details, providing timely reminders.

It also has a text-to-voice conversion to cope with the difficult of reading the text from the mobile screen. This

application will be useful for a large section of the population in making them aware of the medicine they take.

## REFERENCES

- [1] One time password. [online]. Available:<https://youtu.be/Uo7XlWm7KvE>
- [2] SQLite Database for user authentication, medicine details. [online]. Available: <https://youtu.be/NT1qxmQH1eM>
- [3] Image Processing. [online]. Available: <https://youtu.be/GQPyXkbikXw>
- [4] Notification Manager. [online]. Available: <https://youtu.be/xoTKpstv9f0>