A MAJOR PROJECT REPORT ON

Ambulance Services

A report Submitted in partial fulfilment of the requirements of the degree

Bachelor of Technology
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

G.JYOSHNA(R170353)

Under the guidance of

Ms. M. HimaBindu
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES,

RK Valley,

AS A PART OF MAJOR PROJECT

May 2023.

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Director, Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head of the Department Mr.N. Satyanandaram for his encouragement, overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey thanks to our guide at college Miss.M HIMABINDHU for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work. I express my profound gratitude to all our friends and family members for their encouragement.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES ,RK VALLEY ,
May 2023.

CERTIFICATE

This is to certify that the report entitled "**Ambulance services**" submitted by G.Jyoshna(R170353) partial fulfilment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by them under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

M.HimaBindu,
Project Internal Guide,
Computer Science and Engineering,
R.K Valley, RGUKT.

N.Satyanandaram,
Head of the Department,
Computer Science and Engineering,
R.K.Valley, RGUKT.

DECLARATION

We are certifying that, I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the student
G. JYOSHNA
R170353

ABSTRACT

The goal of this paper is to find out what are the steps in developing an application, the difficulty of it and future improvements. With this occasion, I had worked on the healthcare project known as EMRI.This project applications are user friendly and serviceable to the public. This Is the application that coordinates every emergency through a toll free number 108/112 which when called in an emergency ensure prompt communication and activation of a response. This project contains frontend,backend,and mobile application parts. I had worked on the mobile application development. The most commonly used tool for mobile application development environment is Android Studio, the all in one developing tool which ensures full control on the application development.

The development of the user friendly application exploited using the Android platform Android is an open-source software stack created for mobile phones and other devices and is built on top of Linux kernel and GNU software. The software stack of the Android runs Java applications using Java core libraries. Each instance of Java application runs on its own Virtual Machine called Dalvik . Android is freely available to ZManufacturer's for customization, there are no fixed hardware and software configurations .However, Android itself supports features. The following listed are some of them which used in the project Uses SQLite, a lightweight relational database, for data storage.

Table of Contents

1.Introduction

Ambulance services is a pioneer in emergency management services in India. EMRI green health services is the largest professional emergency service provider in India today .
IT is a non profit organization providing emergency medical service. The organization was incepted with the objective of delivering comprehensive,speedy,reliable,and quality emergency care services. There are mobile apps in Google Play related to this . Those apps have great features and functionalities to satisfy users. Today 108 is the synonymous with the best in class emergency service and has been acknowledged as the most efficient ,speedy and caring service provider in its category.

Purpose and Features.

The purpose of this application is to the serve the victims who are in an emergency. The first task that is internal scanning of Qr code when we click on image, if the Qr code is equal to particular text then only it will set as the imageview in their application, This is to avoid any improper or illegal activities carrying in the clientside.Not every QR code contain same data, this application Qr code contain some certain data, if it is not equals ,then it may consider as wrong one.

The second one is text and location recognition of the particular image when the image was captured. It is used to recognize where the hospital is located and the text that contain in the image which shows the hospital name like that here text can be recognized in any language.
Next one is creating a stock tab which contain consumable and non consumable items. It gives the availability of the stock in the ambulance.

2
Requirement ANALYSIS

2.1 Requirement Specification

2.1.1Functional requirements

Simple and interactive GUI: System facilitates simple and interactive Graphical User Interface for the user while handling the system. Android is an open sourcing mobile operating system based on Linux which is a completely open and integrated platform for mobile devices.
Database to store and retrieve the data.

2.1.2 Hardware Requirements

Processor: I3

Hardisk: 64GB

RAM: 4GB

2.1.3 Software Requirements.

Android Os: From Android 7

Android Studio: 3

2.2 Technologies used:

Ios and android platforms:

The two most important mobile app platforms are iOS from Apple Inc. and Android from Google. iOS is Apple's proprietary mobile operating system built specifically for iPhones. Android, however, runs on mobile devices manufactured by various OEMs, including Google.

While there are many similarities between the two, however different software development kits (SDKs) are used for different platforms. Apple uses iOS exclusively for their own devices, while Google has made Android available for other companies that meet specific requirements. Developers have built over 1.5 million applications for both platforms to date.

Java

Java is a high level, class based, object oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers *write once, run anywhere.* Android heavily relies on the Java programming language all the sdks required to build for android applications use the standard libraries of Java. Java is used to develop mobile apps, web apps, desktop apps, games and much more. For functionality of the application a developer write the code in java programming language.

XML

XML stands for extensible Markup Language, which is a way of describing data using a text-based document. Because XML is extensible and very flexible, it's used for many different things, including defining the UI layout of Android apps. For designing of the UI part of an mobile applications developers mostly prefer the xml formats. Basically , xml is used for layout designing. All the UI and layout of your app is designed using xml.Unlike Java (which is Back Bone of your app), xml helps you to design your app , how it will look , how components like buttons , text view , etc will be placed and their styling. Apart from these , xml is also used for parsing data either from database or server into your android app.(Xml parsing).

3
SOFTWARE ENVIRONMENT

Android Studio:

Android is a complete set of software for mobile devices such as tablet computers, notebooks, smartphones, electronic book readers, set-top boxes etc.

It contains a **linux-based Operating System**, **middleware** and **key mobile applications**.

It can be thought of as a mobile operating system. But it is not limited to mobile only.

It is currently used in various devices such as mobiles, tablets, televisions etc.

Android Studio is a new and fully integrated development environment, which has been recently launched by Google for the Android operating system. It has been designed to provide new tools for app development and to provide an alternative to Eclipse, currently the most widely used IDE.

When you begin a new project in Android studio, the project's structure will appear with almost all the files held within the SDK directory, this switch to a Gradle based management system offers an even greater flexibility to the build process.

Android Studio allows you to see any visual changes you make to your app in real-time, and you can also see how it will look on a number of different Android devices, each with different configurations and resolutions, simultaneously.

Android Studio Offers

A robust and straight forward development environment.
An easy way to test performance on other types of device.
Wizards and templates for common elements found in all Android programming.

A full-featured editor with lots of extra tools to speed up the development of your applications. To install Android Studio, it is necessary to have Android's Software Developer Kit (SDK), along with Java Developer Kit (JDK), included in this pack.

Android App Development

Mobile app development is a process for building mobile applications that run on mobile devices. These applications can either be pre-installed or downloaded and installed by the user later. They use the network capabilities of the device to work computing resources remotely. Hence, the mobile app development process requires creating software that can be installed on the device, and enabling backend services for data access through APIs, and testing the application on target devices.

To develop scalable mobile apps, you also need to consider screen sizes, hardware requirements, and many other aspects of the app development process. With an increasing number of jobs in the mobile app development industry, it is essential that the process is well defined and understood by entrepreneurs, startups, and especially developers.

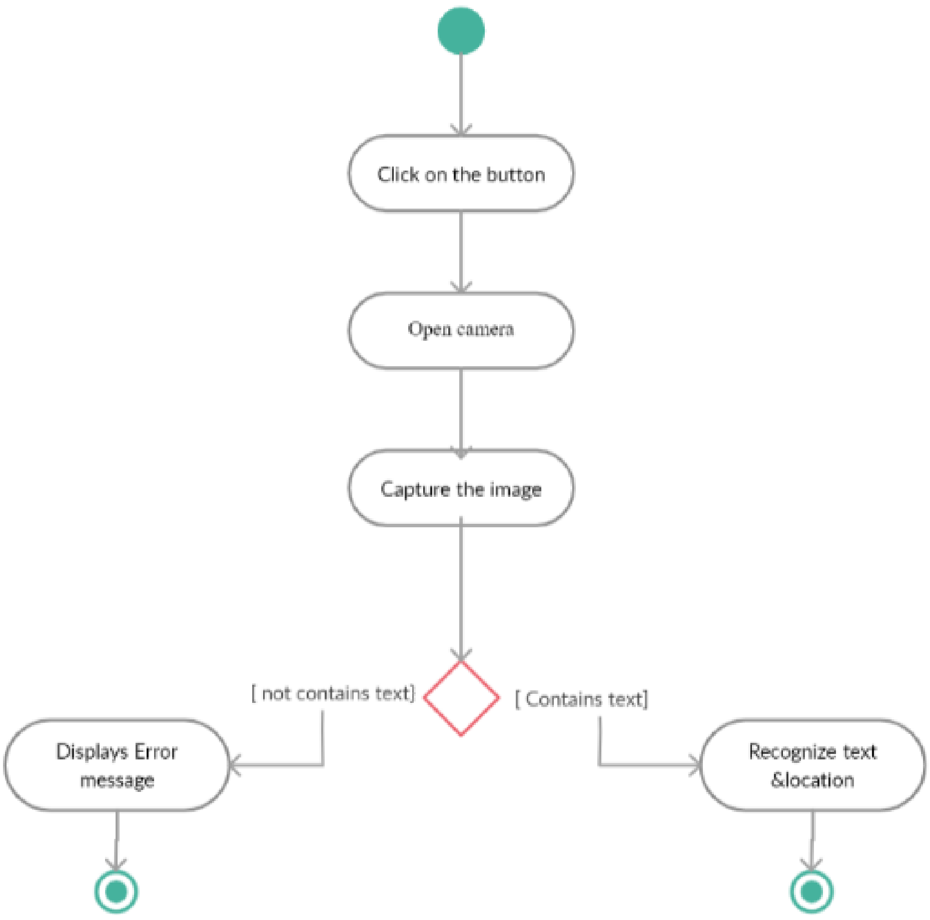
Mobile App Development Lifecycle 2021

There are over 3.5 Billion Software Users worldwide, so there is no doubt that the industry is healthy and thriving. Stats are growing steadily, without any indications of slowing down. And studies show that an average American checks their phone at least once every twelve minutes, and over 10% of these people check their phone about every four minutes. There are some more statistics to keep in mind.

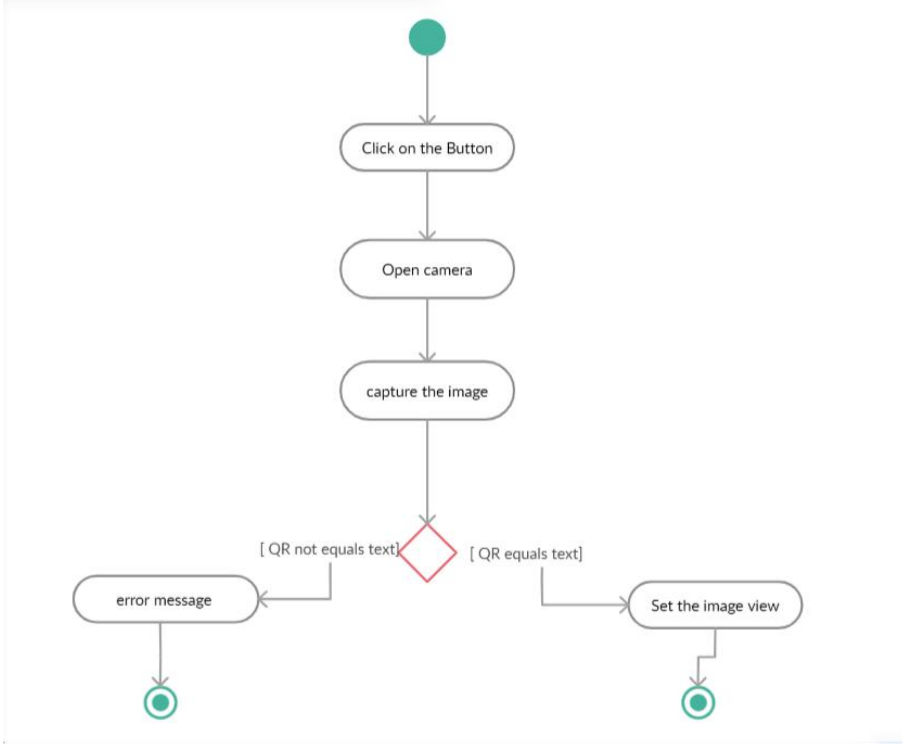
3
SYSTEM DESIGN

UML DIAGRAMS:

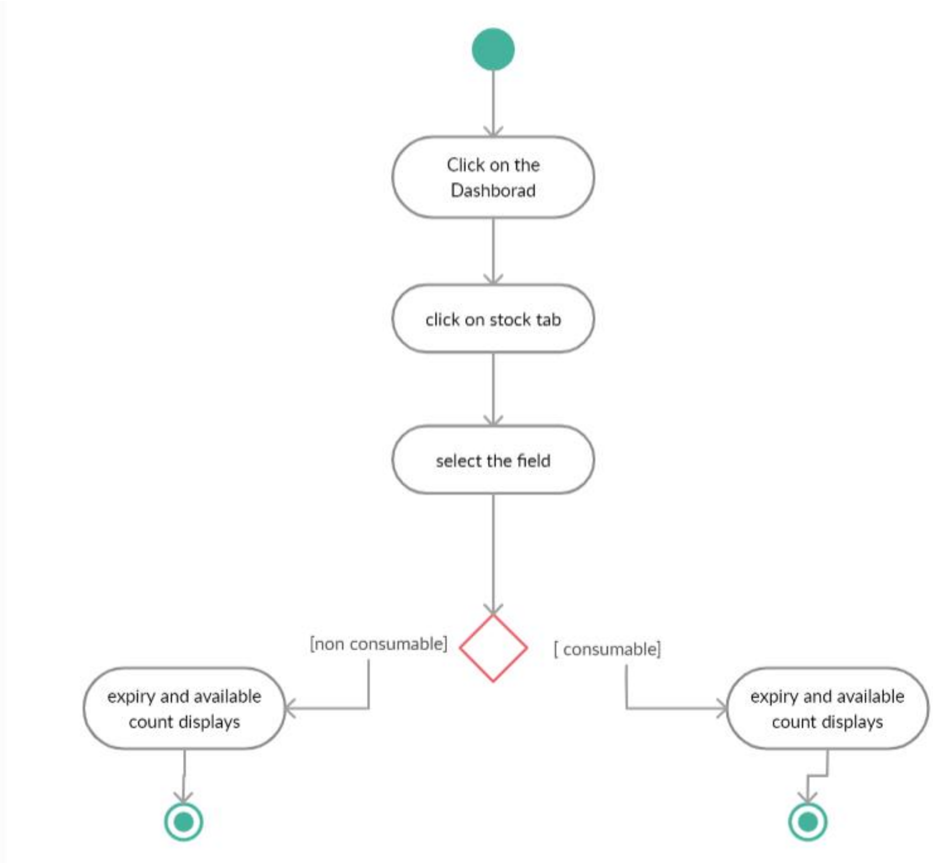
Text recognition when we capture an image.



2.Qr code scanning after image capture



3.Stock tab in ambulance



4
IMPLEMENTATION

4.1 Graphical user interface

The user interface is kept simple and understandable.The user need not take any additional effort to understand the functionality and navigation in the application.The UI designing should be easily understandable and should know where the input is given .Hints are given to help the user in giving the correct input.

4.2 Sample Code & 4.3 Screenshots

1.QR code scanning when we capture an image.

The task is that internal scanning of Qr code when we click on image, if the Qr code is equal to particular text then only it will set as the imageview in their application, This is to avoid any improper or illegal activities carrying in the client side.

Java code

```
package com.task4.myapplication4;
import androidx.activity.result.ActivityResult;
import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.app.Activity;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;
import android.util.*;

import com.google.zxing.ChecksumException;
import com.google.zxing.FormatException;
import com.google.zxing.LuminanceSource;
import com.google.zxing.Reader;
//import com.journeyapps.barcodescanner.ScanContractimport
com.journeyapps.barcodescanner.ScanOptions;
import com.google.zxing.BinaryBitmap;
import com.google.zxing.MultiFormatReader;
import com.google.zxing.NotFoundException;
import com.google.zxing.RGBLuminanceSource;
import com.google.zxing.Result;
import com.google.zxing.common.HybridBinarizer;
import com.google.zxing.oned.MultiFormatOneDReader;

import java.io.FileNotFoundException;
import java.io.InputStream;
import java.net.DatagramPacket;

public class MainActivity extends AppCompatActivity {

    Button b;
    ImageView i;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b = (Button) findViewById(R.id.button);
        i = findViewById(R.id.click_image);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent camera_Intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                someActivityForResultLauncher.launch(camera_Intent);
            }

        });
    }

    ActivityResultLauncher<Intent> someActivityForResultLauncher = registerForActivityResult(
            new ActivityResultContracts.StartActivityForResult(),
            result -> {
                if (result.getResultCode() == RESULT_OK && result.getData() != null) {
                    Bundle bundle = result.getData().getExtras();
                    Bitmap bitmap = (Bitmap) bundle.get("data");

                    String contents = null;

                    int[] intArray = new int[bitmap.getWidth()*bitmap.getHeight()];

                    bitmap.getPixels(intArray, 0, bitmap.getWidth(), 0, 0, bitmap.getWidth(),
bitmap.getHeight());

                    LuminanceSource source = new RGBLuminanceSource(bitmap.getWidth(),
bitmap.getHeight(), intArray);
                    BinaryBitmap bitmap1 = new BinaryBitmap(new HybridBinarizer(source));

                    Reader reader = new MultiFormatReader();
                    Result result1 = null;
                    try {
                        result1 = reader.decode(bitmap1);
                        if(result1!=null && result1.getText().equals("text")){
```

```
                        i.setImageBitmap(bitmap);
                    }
                    else  {
                        Toast.makeText(MainActivity.this, "the content in the qr code is:"
+result1.getText(), Toast.LENGTH_SHORT).show();
                    }

                } catch (NotFoundException e) {
                    Toast.makeText(MainActivity.this, "QR CODE NOT FOUND ",
Toast.LENGTH_SHORT).show();
                    //e.printStackTrace();
                } catch (ChecksumException e) {
                    e.printStackTrace();
                } catch (FormatException e) {
                    e.printStackTrace();
                }


            }


        }
    );
}
```

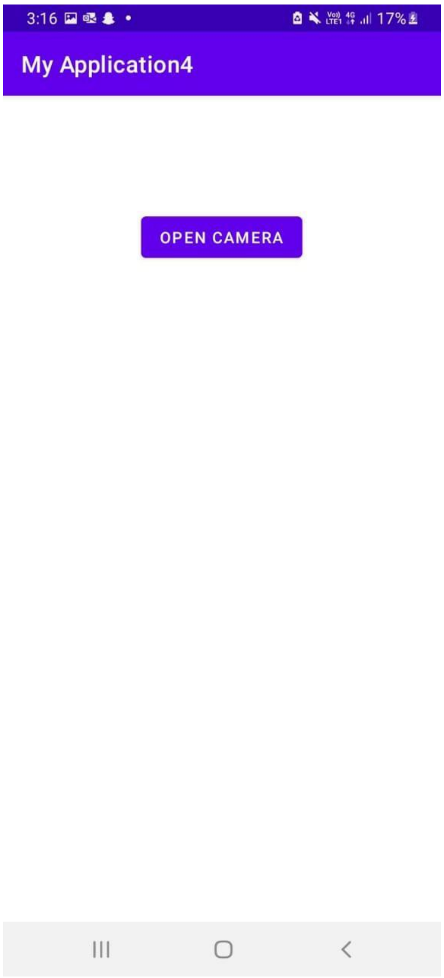
XML CODE:

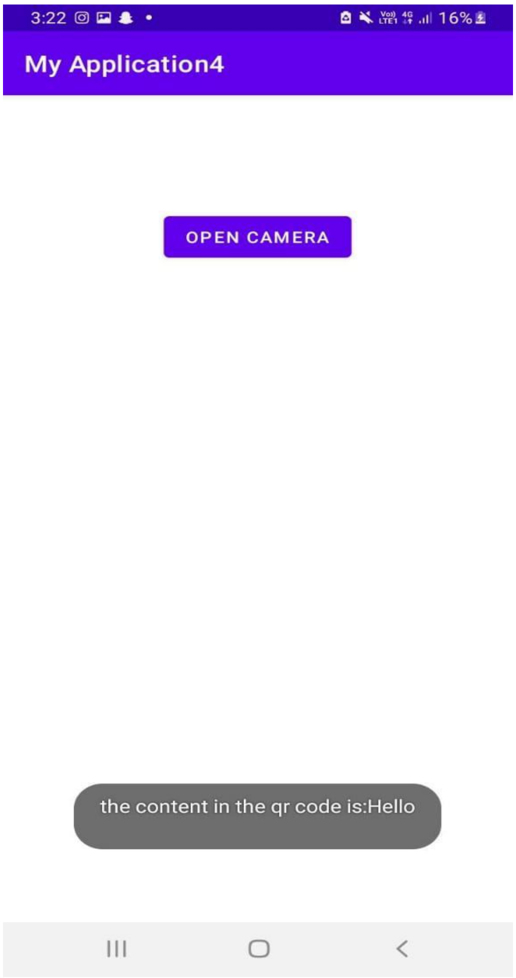
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    tools:context=".MainActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="open camera"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:id="@+id/button"/>
    <ImageView
        android:id="@+id/click_image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="150dp"
        android:layout_marginBottom="10dp"
        />
</RelativeLayout
```

DEPENDENCIES

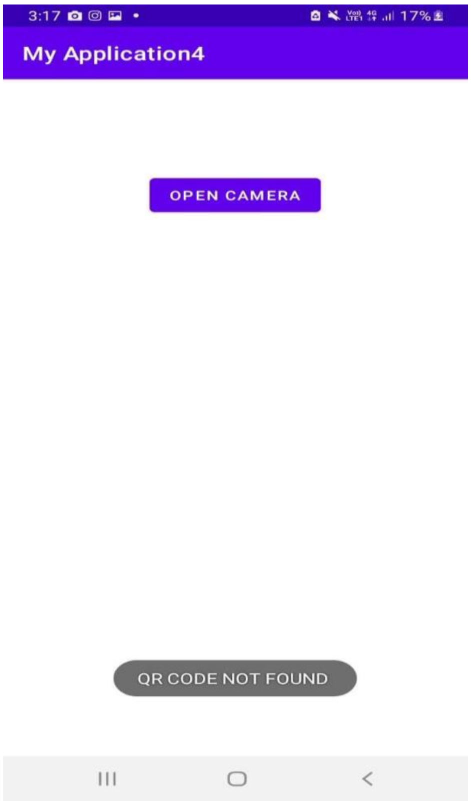
```
dependencies {
    implementation 'androidx.appcompat:appcompat:1.6.0'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'

    implementation 'com.journeyapps:zxing-android-embedded:4.3.0'
}
```

SCREENSHOTS:





the content in the qr code is:Hello



2.Text and location recognition when we capture an image.

The task is text and location recognition of the particular image when   the image was captured. It is used to recognize where the hospital is located and the text that contain in the image which shows the hospital name like that here text can berecognized in any language.

```
package com.task.textrecognization.
import android.Manifest;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.PopupMenu;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.result.ActivityResult;
import androidx.activity.result.ActivityResultCallback;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.android.material.imageview.ShapeableImageView;
import com.google.mlkit.vision.common.InputImage;
import com.google.mlkit.vision.text.Text;
import com.google.mlkit.vision.text.TextRecognition;
import com.google.mlkit.vision.text.TextRecognizer;
import com.google.mlkit.vision.text.devanagari.DevanagariTextRecognizerOptions;
import com.google.mlkit.vision.text.latin.TextRecognizerOptions;

import java.io.IOException;
import java.util.List;
import java.util.Locale;

public class MainActivity extends AppCompatActivity implements LocationListener {

    Button b1;
    TextView tv, t1;
    EditText E1, E2;
    ShapeableImageView imageIv;
    private Uri imageUri = null;
    private static final int CAMERA_REQUEST_CODE = 102;
    private static final int STORAGE_REQUEST_CODE = 103;
    private String[] cameraPermissions;
    private String[] StoragePermissions;
    LocationManager locationManager;
    private ProgressDialog pd;
    private TextRecognizer textRecognizer;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1 = findViewById(R.id.button);
        tv = findViewById(R.id.textview1);
        t1 = findViewById(R.id.textview2);
//        t1.setOnClickListener(new View.OnClickListener() {
```

```
//            @Override
//            public void onClick(View v) {
//                getLocation();
//            }
//        });
        E1 = findViewById(R.id.edittext);
        E2 = findViewById(R.id.edittext2);
        imageIv = findViewById(R.id.imageIv);
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(MainActivity.this, new String[]{

Manifest.permission.ACCESS_FINE_LOCATION,Manifest.permission.ACCESS_COARSE_LOC
ATION
            }, 100);

            cameraPermissions = new String[]{android.Manifest.permission.CAMERA,
android.Manifest.permission.WRITE_EXTERNAL_STORAGE};
            StoragePermissions = new
String[]{android.Manifest.permission.WRITE_EXTERNAL_STORAGE};
            textRecognizer = TextRecognition.getClient(new
DevanagariTextRecognizerOptions.Builder().build());
            b1.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    showInputImageDialog();
                }

            });
        }
    }

    @SuppressLint("MissingPermission")
    private void getLocation() {
        try {
            locationManager = (LocationManager)
getApplicationContext().getSystemService(LOCATION_SERVICE);
            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 5000, 5,
MainActivity.this);
            locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0,
MainActivity.this);

            } catch (Exception e) {
                e.printStackTrace();
            }

        }

        private void recognisedtext () {
```

```
        try {
            InputImage inputimage = InputImage.fromFilePath(this, imageUri);
            Task<Text> text = textRecognizer.process(inputimage)
                    .addOnSuccessListener(new OnSuccessListener<Text>() {
                        @Override
                        public void onSuccess(Text text) {

                            String recognizedtext = text.getText();
                            E1.setText(recognizedtext);
                            //Toast.makeText(MainActivity.this, "the stored text is "+recognizedtext,
Toast.LENGTH_SHORT).show();
                        }
                    })
                    .addOnFailureListener(new OnFailureListener() {
                        @Override
                        public void onFailure(@NonNull Exception e) {

                            Toast.makeText(MainActivity.this, "failed to recognized the data due  to" +
e.getMessage(), Toast.LENGTH_SHORT).show();
                        }
                    });

        } catch (Exception e) {

            Toast.makeText(this, "failed to load the data" + e.getMessage(),
Toast.LENGTH_SHORT).show();

        }
    }
    private void showInputImageDialog () {
        PopupMenu popmenu = new PopupMenu(this, b1);
        popmenu.getMenu().add(Menu.NONE, 1, 1, "CAMERA");
//       popmenu.getMenu().add(Menu.NONE,2,2,"GALLERY");
        popmenu.show();
        popmenu.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                int id = item.getItemId();
                if (id == 1) {
                    if (checkCameraPermissions()) {
                        pickImageCamera();
                    } else {
                        requestCameraPermissions();
                    }
                }
                return true;
            }
        });
    }

    private void pickImageCamera () {
        ContentValues values = new ContentValues();
```

```
          values.put(MediaStore.Images.Media.TITLE, "SAMPLE TITLE");
          values.put(MediaStore.Images.Media.DESCRIPTION, "SAMPLE DES");
          imageUri =
getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values);
          Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
          intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);
          cameraActivityResultLauncher.launch(intent);
       }
       private ActivityResultLauncher<Intent> cameraActivityResultLauncher =
registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new
ActivityResultCallback<ActivityResult>() {
                @Override
                public void onActivityResult(ActivityResult result) {
                   if (result.getResultCode() == Activity.RESULT_OK) {
                      imageIv.setImageURI(imageUri);
                      recognisedtext();
                      getLocation();
                   } else {
                      Toast.makeText(MainActivity.this, "canceled", Toast.LENGTH_SHORT).show();
                   }

                }
             }
       );
       private boolean checkStoragePermission () {
          boolean result = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
(PackageManager.PERMISSION_GRANTED);
          return result;
       }
       private void requestStoragePermission () {
          ActivityCompat.requestPermissions(this, StoragePermissions,
STORAGE_REQUEST_CODE);
       }
       private boolean checkCameraPermissions () {
          boolean cameraResult = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
(PackageManager.PERMISSION_GRANTED);
          boolean StorageResult = ContextCompat.checkSelfPermission(this,
android.Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
(PackageManager.PERMISSION_GRANTED);
          return cameraResult && StorageResult;
       }
       private void requestCameraPermissions () {
          ActivityCompat.requestPermissions(this, cameraPermissions,
CAMERA_REQUEST_CODE);
       }

       @Override
       public void onRequestPermissionsResult ( int requestCode, @NonNull String[] permissions,
       @NonNull int[] grantResults){
          super.onRequestPermissionsResult(requestCode, permissions, grantResults);
```

```
            switch (requestCode) {
                case CAMERA_REQUEST_CODE: {
                    if (grantResults.length > 0) {
                        boolean cameraAccepted = grantResults[0] ==
PackageManager.PERMISSION_GRANTED;
                        boolean StorageAccepted = grantResults[1] ==
PackageManager.PERMISSION_GRANTED;
                        if (cameraAccepted && StorageAccepted) {
                            pickImageCamera();
                        } else {
                            Toast.makeText(this, "camera and storage permission required",
Toast.LENGTH_SHORT).show();
                        }
                    } else {
                        Toast.makeText(this, "cancelled", Toast.LENGTH_SHORT).show();
                    }
                }
                break;
            }
        }

        @Override
        public void onLocationChanged (@NonNull Location location){
            //Toast.makeText(this, ""+location.getLatitude()+","+location.getLongitude(),
Toast.LENGTH_SHORT).show();
            try {
                Geocoder geocoder = new Geocoder(MainActivity.this, Locale.getDefault());
                String latlong = location.getLatitude() + "," + location.getLongitude();
                //Toast.makeText(this, ""+location.getLatitude()+","+location.getLongitude(),
Toast.LENGTH_SHORT).show();
                List<Address> addresses = geocoder.getFromLocation(location.getLatitude(),
location.getLongitude(), 1);
                String address = addresses.get(0).getAddressLine(0);
                E2.setText(latlong + address);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        @Override
        public void onLocationChanged (@NonNull List < Location > locations) {
            LocationListener.super.onLocationChanged(locations);
        }

        @Override
        public void onFlushComplete ( int requestCode){
            LocationListener.super.onFlushComplete(requestCode);
        }

        @Override
        public void onProviderEnabled (@NonNull String provider){
            LocationListener.super.onProviderEnabled(provider);
        }

        @Override
        public void onProviderDisabled (@NonNull String provider){
            LocationListener.super.onProviderDisabled(provider);
        }
    }
```

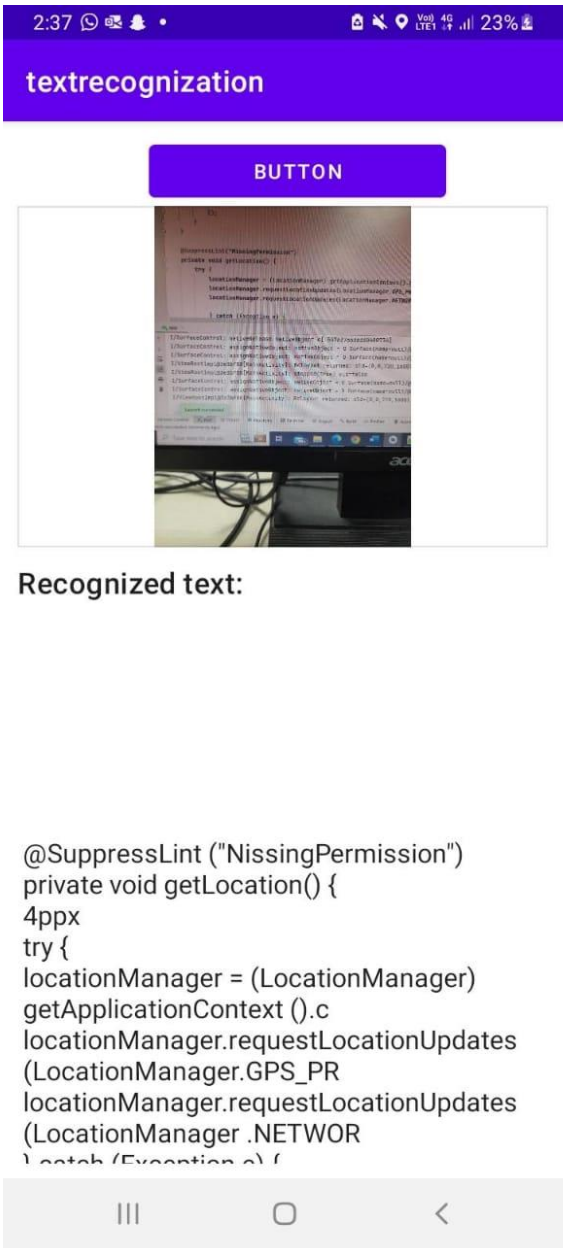
XML CODE:

```
<?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:padding="10dp"
    tools:context=".MainActivity"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
            android:id="@+id/button"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="90dp"
            android:layout_marginEnd="70dp"
            android:layout_weight="1"
            android:text="Button" />

    </LinearLayout>
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <com.google.android.material.imageview.ShapeableImageView
                android:id="@+id/imageIv"
                android:layout_width="match_parent"
                android:layout_height="234dp"
                android:adjustViewBounds="true"
                app:strokeWidth="2dp" />

            <TextView
                style="@style/TextAppearance.MaterialComponents.Headline6"
                android:id="@+id/textview1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:text="Recognized text:" />
            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/edittext"
                android:text="text"
                android:layout_marginTop="150dp"

            />
            <TextView
                style="@style/TextAppearance.MaterialComponents.Headline6"
                android:id="@+id/textview2"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="10dp"
                android:text="location:" />
            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:id="@+id/edittext2"
                android:text="address"
                android:layout_marginTop="10dp"/>


        </LinearLayout>
    </ScrollView>
</LinearLayout>
```

Dependencies:

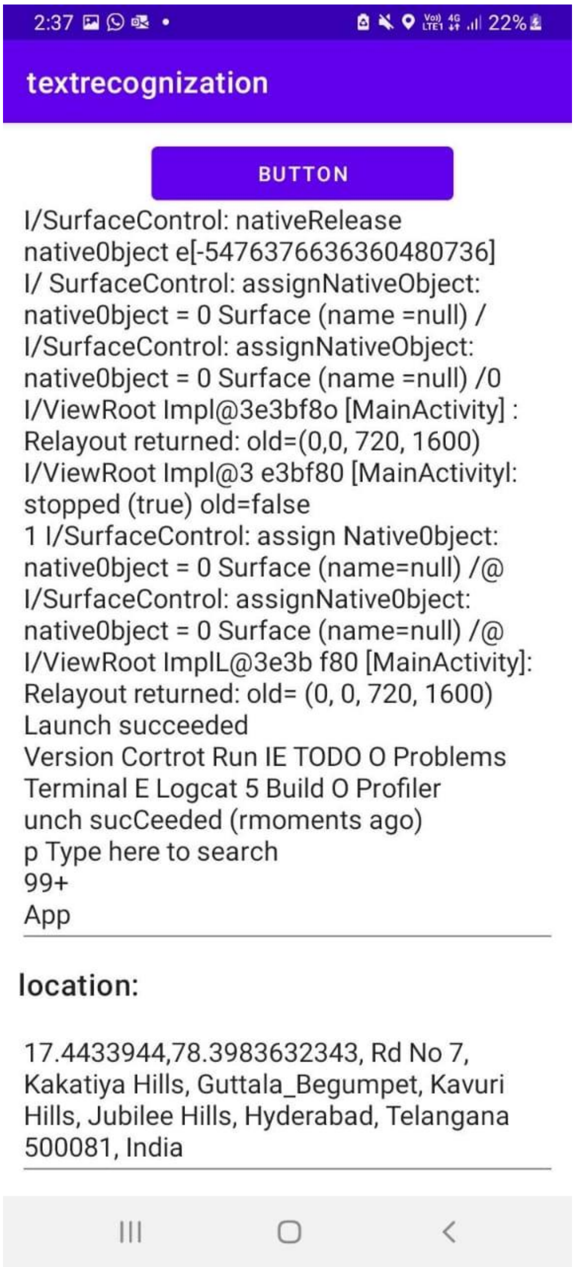
```
dependencies {

    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.8.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    implementation 'com.google.android.gms:play-services-mlkit-text-recognition:18.0.2'
    implementation 'com.google.mlkit:text-recognition-devanagari:16.0.0-beta6'
    implementation "com.google.android.gms:play-services-location:18.0.0"
}
```

Screenshots:



Recognized text:

```
@SuppressLint ("NissingPermission")
private void getLocation() {
4ppx
try {
locationManager = (LocationManager)
getApplicationContext ().c
locationManager.requestLocationUpdates
(LocationManager.GPS_PR
locationManager.requestLocationUpdates
(LocationManager .NETWOR
} catch (Exception e) {
```



```
I/SurfaceControl: nativeRelease
nativeObject e[-5476376636O480736]
I/ SurfaceControl: assignNativeObject:
nativeObject = 0 Surface (name =null) /
I/SurfaceControl: assignNativeObject:
nativeObject = 0 Surface (name =null) /0
I/ViewRoot Impl@3e3bf8o [MainActivity] :
Relayout returned: old=(0,0, 720, 1600)
I/ViewRoot Impl@3 e3bf80 [MainActivityl:
stopped (true) old=false
1 I/SurfaceControl: assign NativeObject:
nativeObject = 0 Surface (name=null) /@
I/SurfaceControl: assignNativeObject:
nativeObject = 0 Surface (name=null) /@
I/ViewRoot ImplL@3e3b f80 [MainActivity]:
Relayout returned: old= (0, 0, 720, 1600)
Launch succeeded
Version Cortrot Run IE TODO O Problems
Terminal E Logcat 5 Build O Profiler
unch sucCeeded (rmoments ago)
p Type here to search
99+
App
```

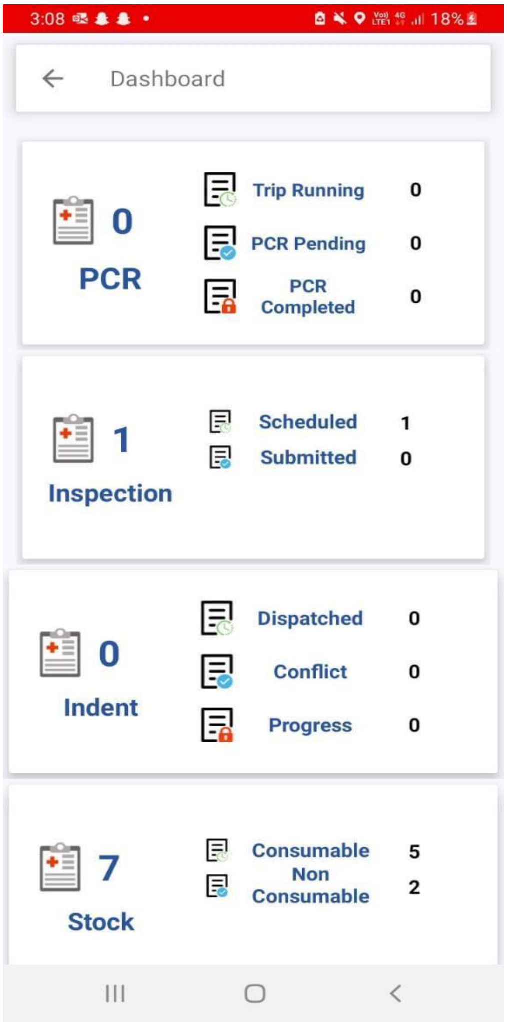
location:

17.4433944,78.3983632343, Rd No 7, Kakatiya Hills, Guttala_Begumpet, Kavuri Hills, Jubilee Hills, Hyderabad, Telangana 500081, India

3. Creating a stock tab which contain consumable and non consumable items.

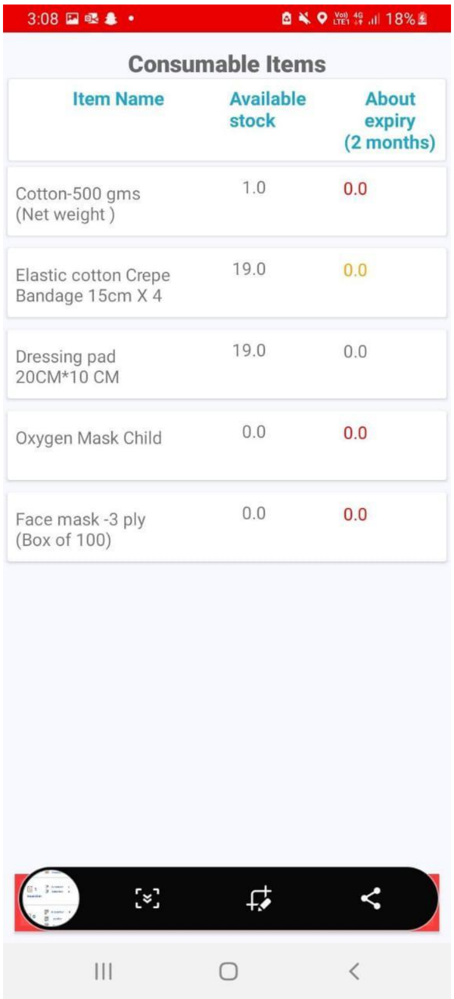
It gives the availability of the stock in the ambulance .It is used to know the expired items count in the ambulance. If the expired count less than the minimum stock it indicates in red colour.

Screenshoots:



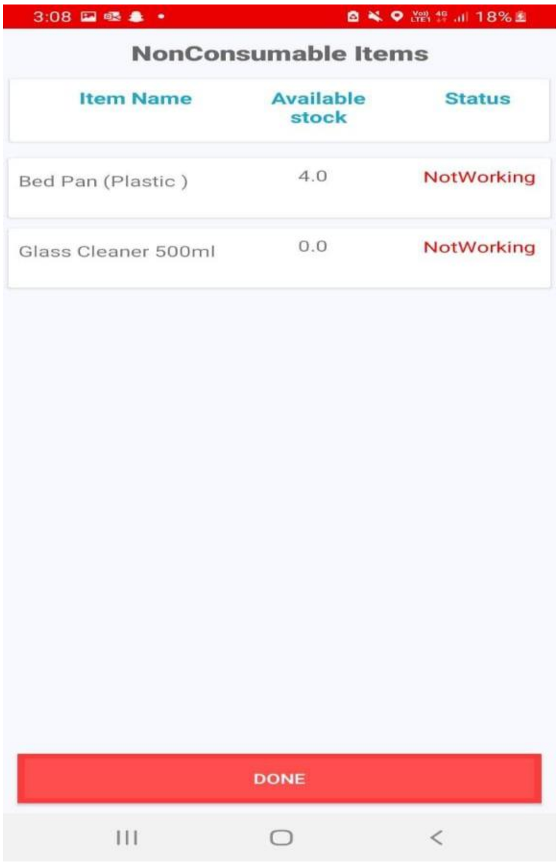
Dashboard

0 PCR	Trip Running	0
	PCR Pending	0
	PCR Completed	0
1 Inspection	Scheduled	1
	Submitted	0
0 Indent	Dispatched	0
	Conflict	0
	Progress	0
7 Stock	Consumable	5
	Non Consumable	2



Consumable Items

Item Name	Available stock	About expiry (2 months)
Cotton-500 gms (Net weight)	1.0	0.0
Elastic cotton Crepe Bandage 15cm X 4	19.0	0.0
Dressing pad 20CM*10 CM	19.0	0.0
Oxygen Mask Child	0.0	0.0
Face mask -3 ply (Box of 100)	0.0	0.0



NonConsumable Items

Item Name	Available stock	Status
Bed Pan (Plastic)	4.0	NotWorking
Glass Cleaner 500ml	0.0	NotWorking

6.Testing

System tests are designed to validate a fully developed system to assure that it meets its requirements. The test cases are therefore designed solely based on the SRS document.
(OR)

System testing is nothing but how the customer is going to start using your application and checking whether everything is up to the mark and meeting the needs of the customer

6.1 Unit Testing:

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. Reduces Defects in the Newly developed features or reduces bugs when changing the existing functionality.

Improves design and allows better refactoring of code. Unit Tests, when integrated with build gives the quality of the build as well. It is the first level of functional testing. Below are the test cases on the individual modules of the designed website. The functionality of each module has been checked by the developer of the module.

6.2 Integration Testing:

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units. Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The goal of integration testing is to check the correctness of communication among all the modules. It includes four types of approaches.

A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.
Integration testing includes various approaches like

1. Bigbang approach

2. Top down approach

3.Bottom up approach

4.Sandwiched Integration testing.

7
Conclusion and References

7.1 Conclusion

This project main goal is to the serve the people. It is the world's largest Integrated emergency response services(ERS) provider serving 40% of India's population across 10 states. As a not -for-profit professional organization operating in the public private partnership mode,EMRI handles,medical,Police and fire emergencies through the 108 service"

This is the free service delivered through state of art emergency call centres and has over 2000 ambulances across 10 states of India.
This tasks i.e adding a stock tab, which is useful to the ambulance driver for knowing the available stock either consumable stock or non-consumable stock in the ambulance. This feature is also useful to recognize the expiry items count in the ambulance. And Qr code scanning is useful for finding any improper activities that is carried out. Text and location recognition when capturing an image is useful to the user to fetch the exact location where he capture the image and he get the text in the image completely which contain different languages. This QR code and text recognition are one of the very useful features in the project.

7.2 References

For adding this features(tasks) in the project, I had referred the following

https://www.geeksforgeeks.org/android-tutorial/

https://developer.android.com/

https://www.w3schools.blog/android-tutorial