CHAPTER1

INTRODUCTION

1.1 INTRODUCTION TO MOBILE APPLICATION DEVELOPMENT

Mobile app development is the act or process by which a mobile app is developed for mobile devices, such as personal digital assistants, enterprise digital assistants or mobile phones. These applications can be pre-installed on phones during manufacturing platforms, or delivered as web applications using server-side or client-side processing (e.g., JavaScript) to provide an "application-like" experience within a Web browser. Application software developers also must consider a long array of screen sizes, hardware specifications, and configurations because of intense competition in mobile software and changes within each of the platforms. Mobile app development has been steadily growing, in revenues and jobs created. A 2013 analyst report estimates there are 529,000 direct app economy jobs within the EU then 28 members (including the UK), 60 percent of which are mobile app developers.

As part of the development process, mobile user interface (UI) design is also essential in the creation of mobile apps. Mobile UI considers constraints, contexts, screen, input, and mobility as outlines for design. The user is often the focus of interaction with their device, and the interface entails components of both hardware and software. User input allows for the users to manipulate a system, and device's output allows the system to indicate the effects of the users' manipulation. Mobile UI design constraints include limited attention and form factors, such as a mobile device's screen size for a user's hand(s). Mobile UI contexts signal cues from user activity, such as location and scheduling that can be shown from user interactions within a mobile app. Overall, mobile UI design's goal is mainly for an understandable, user-friendly interface.

1.2 JAVA

Java is a popular programming language. It is owned by Oracle, and more devices run Java. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). It is an open source secure, fast, free and powerful. As Java is close to C++ and C#, it makes it easy for programmers to switch to Java. Java was developed in the mid-1990s by James A. Gosling, a former computer scientist with Sun Microsystems

1.3 INTRODUCTION TO ANDROID STUDIO

Android Studio is an authority incorporated advancement condition (IDE) for an android application improvement, in the light of the android Studio is planned explicitly for the android improvement accessibility for the download on Windows & Linux, android application being Google's essential development environment for the local android app advancement. Android Studio also offers adaptable Grade-based form framework, the code formats to enable the user to fabricate the regular android app highlights & the rich format editorial manager with the help for the intuitive topic altering & also worked help for the Google Cloud Platform, making it simple to coordinate with the Google Cloud Messaging android app Engine & considerably more android Studio as good as ever interface structure point of view where you can see the interface you are taking a shot at and its related segments. android Studio give away different UI instruments to help you with making designs, executing style subjects, & making realistic.

1.4 INTRODUCTION TO FIREBASE

Firebase is a Backend-as-a-Service which started as a YC11 start-up. It grew up into a next-generation app-development platform on Google Cloud Platform. Firebase is a real-time database that allows storing a list of objects in the form of a tree. We can synchronize data between different devices. It is a software which allows developers to develop Android, IOS, and Web apps. For reporting and fixing app crashes, tracking analytics, creating marketing and product experiments, firebase provides several tools. Firebase has three main services, i.e., a real-time database, user authentication, and hosting. Firebase evolved from Envolve. Envolve is a prior start-up founded by James Tamplin and Andrew Lee in 2011. Firebase Real-time Database was the first product of firebase. It is an API which syncs application data across Android, iOS, and Web devices. It gets stored on Firebase's cloud. Then the firebase real-time database helps the developers to build real-time, collaborative applications. Firebase manages real-time data in the database. So, it easily and quickly exchanges the data to and from the database. Hence, for developing mobile apps such as live streaming, chat messaging, etc., Firebase allows syncing real-time data across all devices - iOS, Android, and Web - without refreshing the screen.

1.5 NECESSITY OF THE PROJECT.

The use of surveillance cameras to consistently monitor surveillance-required places, such as banks, offices, roads, subways, and shopping malls, has become common everywhere to prevent unfortunate incidents, such as robbery, assault, and homicide. There is a huge proliferation in the development of cameras in terms of different sizes, shapes, functionalities, and applications. Modern technology has made possible cameras as small as the size of a human nail while embedding all the features required for the normal operation of a camera, such as Wi-Fi connectivity, audio support, and high definition visual quality. Moreover, they are available in different packages, such as power outlets, USB chargers, and smoke detectors . Such innovations have permitted its illegal use in many private zones, such as hotel rooms , Airbnb , restroom and changing room , which can expose and record the private moments of the victims. Figure 1 gives a typical example, where the camera is placed in a hidden place (dress stand), and the victim is unaware of its placement. This camera (hereafter called spy camera) surreptitiously records the monitored area and streams the data to the remote storage (cloud storage) or local storage .

1.4 APPLICATION AND ADVANTAGES OF SPY CAMERA.

Investigative journalism, Expose to save the day, Stopping Shoplifting and Burglary, The babysitter problem, Proof of crime, Corporate secrets safeguarded .When people think about Spy camera systems, they can drop the idea because of the budget. Just think of the benefit that the cost will give you. Know who comes and goes when you are not home. Find out if your rules are being followed. Spy cameras can help keep a record of thieves on your property. Burglars sometimes approach a house to determine if the owners are home. Many spy cameras are equipped to communicate with you via text notification when activity happens in front of the camera. Potential victims of a burglary need to use a spy camera, so everything can be recorded. When filing a police report, you can refer to the recordings of your spy camera and possibly the police can catch the thief. Meanwhile, in extreme cases, the camera evidence can be used in legal proceedings. Sometimes strangers walk on your property without getting your permission. In cases like this, spy cameras easily capture suspicious behavior of unknown visitors. Seeing that activity allows you to take precautionary measures for property protection.Does your cat act innocent when you simply ask about the claw marks on the sofa? And you don’t know which puppy went in the trash? Now you can monitor your pet's behavior and easily find a solution to fix theproblem.

CHAPTER2

REQUIREMENT SPECIFICATION

2.1 Project Requirements

The package is designed such that users with an Android phone having minimum configuration can also use it. It does not require complex computing.

The App requires a simple daily use android phone in which this app can be installed and then user can run it. For a developer it is required to install android studio or IntelliJ Application software which is used to design the app using Xml and Java. Firebase is used as backend to store the data.

2.2 HARDWARE REQUIRMENTS

- ✓ Hard Disk: 1TB 7200 rpm
- ✓ Processor: 10th Generation Intel Core i5
- ✓ RAM: 8 GB
- ✓ Graphics: RADEON / NVIDIA
- ✓ Monitor resolution - A color monitor with a minimum resolution

2.3 SOFTWARE REQUIRMENTS

- ✓ Operating System: Windows 10
- ✓ User Interface: XML
- ✓ Programming Language: Java
- ✓ Application: Android Studio
- ✓ Database: Firebas

CHAPTER 3

SYSTEM DESIGN

3.1 Basic Layout

- Register Page
- Login Page.
- Main Page.
- Result page

The above shown layout is a basic idea of the Application. The Application starts with a Get Started Page that asks user to sign up

- Register Page: In this page user can register name ,email id and password and than registered.
- Login Page: In this page user login entered email id and password.
- Main Page: It defines the different features of the application.
- Result Page: Displays the result.

3.2 Flow Chart

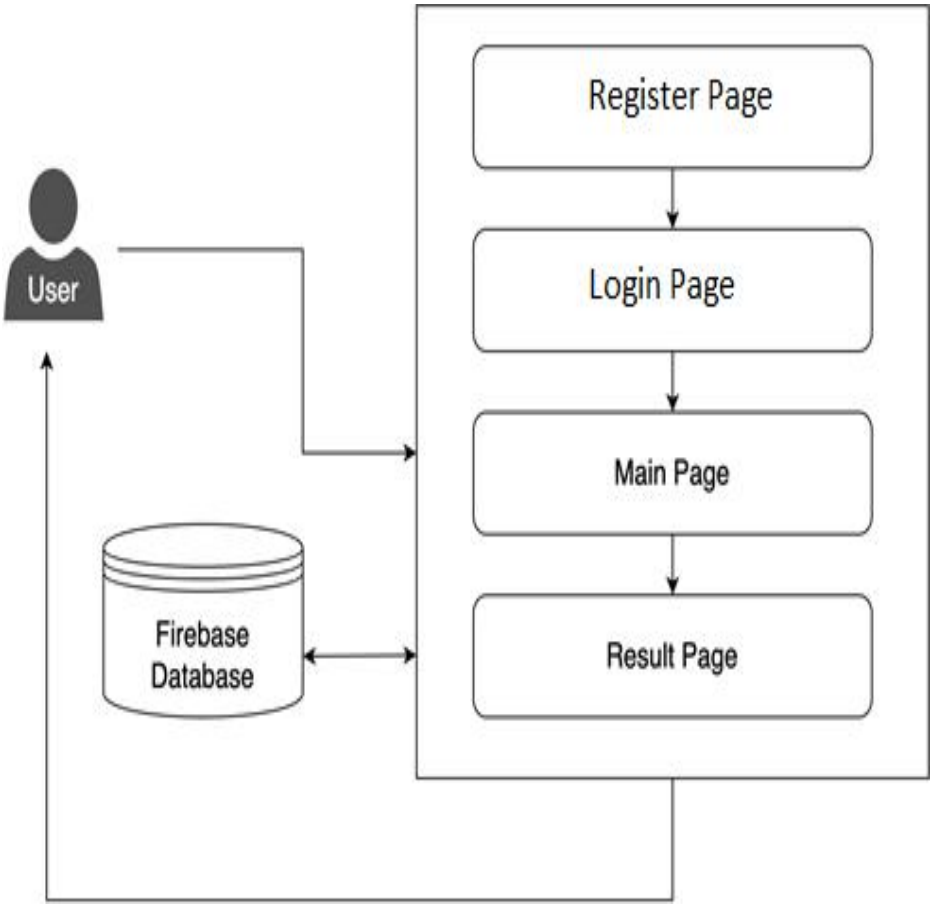

Fig 3.2.1 Flow Diagram

CHAPTER 4

IMPLEMENTATION

4.1 Registration Page

This is the implementation details of the registration page of the program.

```
package com.mad.spycameradetection;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        rfname = findViewById(R.id.rfname); remail =
        findViewById(R.id.remail); rpassword =
        findViewById(R.id.rpassword);rphone =
        findViewById(R.id.rphone);
        rregister = (Button) findViewById(R.id.rregister);rlogin =
        (TextView) findViewById(R.id.rlogin);

        mAuth = FirebaseAuth.getInstance(); rregister.setOnClickListener(new
        View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String email = remail.getText().toString();
                String password = rpassword.getText().toString();

                String fullName = rfname.getText().toString();String
                phone = rphone.getText().toString();

                if (TextUtils.isEmpty(email))
                    { remail.setError("Email is Required.");
                    return;
                }

                if (TextUtils.isEmpty(password))
                    { rpassword.setError("Password is Required.");return;
```

```
            }

            if (password.length() < 6) {
                rpassword.setError("Password Must be Alteast 6 Characters");return;
            }

            // create new user or register new user mAuth.createUserWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {

                @Override
                public void onComplete(@NonNull Task<AuthResult> task)
                {
                    if (task.isSuccessful())
                        { Toast.makeText(getApplicationContext(),"Registration
Successful!",Toast.LENGTH_LONG).show();

                        // if the user created intent to login activity
                        Intent intent = new Intent(Register.this,MainActivity.class);
                        startActivity(intent);
                    }
                    else {
                        // Registration failed Toast.makeText(getApplicationContext(),"Registration
                        Failed!!" + "
Please Try Again Later",Toast.LENGTH_LONG).show();

                    }
                }
            });
        }
    });

    rlogin.setOnClickListener(new View.OnClickListener()
        {@Override
        public void onClick(View v) {
            startActivity(new Intent(getApplicationContext(), LOGIN.class));
        }
    });
```

4.2 Speedometer View

This is the implementation details of the speedometer view of the project.

```
package com.mad.spycameradetection;
public class SpeedometerView extends View {

    private static final String TAG = SpeedometerView.class.getSimpleName();
    public static final double DEFAULT_MAX_SPEED = 100.0;
    public static final double DEFAULT_MAJOR_TICK_STEP = 20.0;
    public static final int DEFAULT_MINOR_TICKS = 1;

    private double maxSpeed = DEFAULT_MAX_SPEED;
    private double speed = 0;
    private int defaultColor = Color.rgb(180, 180, 180);
    private double majorTickStep = DEFAULT_MAJOR_TICK_STEP;
    private int minorTicks = DEFAULT_MINOR_TICKS;
    private LabelConverter labelConverter;

    private List<ColoredRange> ranges = new ArrayList<ColoredRange>();
    private Paint backgroundPaint;
    private Paint backgroundInnerPaint;
    private Paint maskPaint;
    private Paint needlePaint;
    private Paint ticksPaint;
    private Paint txtPaint;
    private Paint colorLinePaint;

   private Bitmap mMask;
    public SpeedometerView(Context context)
        {super(context);
        init();
    }

    public SpeedometerView(Context context, AttributeSet attrs)
        {super(context, attrs);
        TypedArray attributes =
              context.getTheme().obtainStyledAttributes(attrs,
              R.styleable.SpeedometerView(0, 0);
        try {
           setMaxSpeed(attributes.getFloat(R.styleable.SpeedometerView_maxSpeed, (float)
DEFAULT_MAX_SPEED));
           setSpeed(attributes.getFloat(R.styleable.SpeedometerView_speed, 0));
        } finally {
           attributes.recycle(); // recycle() :- read by taping on it where it is used.
        }
```

```
        init();
    }
    public double getMaxSpeed()
        {return maxSpeed;
    }
    public void setMaxSpeed(double maxSpeed)
        {if (maxSpeed <= 0)
            throw new IllegalArgumentException("Non-positive value specified as max speed.");
        this.maxSpeed = maxSpeed;
        invalidate(); // invalidate() :- read by taping on it where it is used.
    }

    public double getSpeed()
        {return speed;
    }

    public void setSpeed(double speed)
        {if (speed < 0)
            throw new IllegalArgumentException("Non-positive value specified as a speed.");
        if (speed > maxSpeed)
            speed = maxSpeed;
        this.speed = speed;
        invalidate();
    }

    @TargetApi(11)
    // ValueAnimator :- read about it by just tapping on it where it is used
    public ValueAnimator setSpeed(double progress, long duration, long startDelay)
        {if (progress <= 0)
            throw new IllegalArgumentException("Non-positive value specified as a speed.");

        if (progress > maxSpeed)
            progress = maxSpeed;

        ValueAnimator va = ValueAnimator.ofObject(new TypeEvaluator<Double>() { // read about
ValueAnimator.ofObject by just tapping on it
            @Override
            public Double evaluate(float fraction, Double startValue, Double endValue)
                {return startValue + fraction*(endValue-startValue);
            }
        }, Double.valueOf(getSpeed()), Double.valueOf(progress));

        va.setDuration(duration); // setDuration() :- read by just tapping on where this is used
        va.setStartDelay(startDelay); // setStartDelay() :- read by just tapping on where this is used
```

```
        va.addUpdateListener(new ValueAnimator.AnimatorUpdateListener() { // addUpdateListener() :-
read by just tapping on where this is used
            public void onAnimationUpdate(ValueAnimator animation)
                {Double value = (Double) animation.getAnimatedValue();
                if (value != null)
                    setSpeed(value);
                Log.d(TAG, "setSpeed(): onAnumationUpdate() -> value = " + value);
            }
        });
        va.start(); // start() :- read by just tapping on where this is used
        return va;
    }

    @TargetApi(11)
    public ValueAnimator setSpeed(double progress, boolean animate)
        {return setSpeed(progress, 1500, 200);
    }

    public int getDefaultColor()
        {return defaultColor;
    }

    public void setDefaultColor(int defaultColor)
        {this.defaultColor = defaultColor;
        invalidate();
    }

    public double getMajorTickStep()
        {return majorTickStep;
    }

    public void setMajorTickStep(double majorTickStep)
        {if (majorTickStep <= 0)
            throw new IllegalArgumentException("Non-positive value specified as a major tick step.");
        this.majorTickStep = majorTickStep;
        invalidate();
    }

    public int getMinorTicks()
        {return minorTicks;
    }

    public void setMinorTicks(int minorTicks)
        {this.minorTicks = minorTicks;
        invalidate();
```

```
    }

    public LabelConverter getLabelConverter()
        {return labelConverter;
    }

    public void setLabelConverter(LabelConverter labelConverter)
        {this.labelConverter = labelConverter;
        invalidate();
    }

    public void clearColoredRanges() {
        ranges.clear(); // clear() :- read by just tapping on where this is used
        invalidate();
    }

    public void addColoredRange(double begin, double end, int color)
        {if (begin >= end)
            throw new IllegalArgumentException("Incorrect number range specified!");
        if (begin < - 5.0/160* maxSpeed)
            begin = - 5.0/160* maxSpeed;
        if (end > maxSpeed * (5.0/160 + 1))
            end = maxSpeed * (5.0/160 + 1);
        ranges.add(new ColoredRange(color, begin, end));
        invalidate();
    }

    @Override
    protected void onDraw(Canvas canvas) { // read about Canvas by just tapping on it where it is used
        super.onDraw(canvas);
        canvas.drawColor(Color.TRANSPARENT); // read about canvas.drawColor() by just tapping on
it where it is used
        drawBackground(canvas);
        drawTicks(canvas);
        drawNeedle(canvas);
    }
) {
    private RectF getOval(Canvas canvas, float factor)
        {RectF oval;
        final int canvasWidth = canvas.getWidth() - getPaddingLeft() - getPaddingRight();
        final int canvasHeight = canvas.getHeight() - getPaddingTop() - getPaddingBottom();

        if (canvasHeight*2 >= canvasWidth) {
            oval = new RectF(0, 0, canvasWidth*factor, canvasWidth*factor);
        } else {
```

```
            oval = new RectF(0, 0, canvasHeight*2*factor, canvasHeight*2*factor);
        }

        // read about oval.offset() by tapping on where it is used
        oval.offset((canvasWidth-oval.width())/2 + getPaddingLeft(), (canvasHeight*2-oval.height())/2 +
getPaddingTop());

        return oval;
    }

    private RectF getOval(float w, float h)
        {RectF oval;
        final float canvasWidth = w - getPaddingLeft() - getPaddingRight();
        final float canvasHeight = h - getPaddingTop() - getPaddingBottom();
        if (canvasHeight*2 >= canvasWidth) {
            oval = new RectF(0, 0, canvasWidth, canvasWidth);
        } else {
            oval = new RectF(0, 0, canvasHeight*2, canvasHeight*2);
        }
        return oval;
    }

    private void drawBackground(Canvas canvas)
        {RectF oval = getOval(canvas, 1);
        canvas.drawArc(oval, 180, 180, true, backgroundPaint);

        RectF innerOval = getOval(canvas, 0.9f);
        canvas.drawArc(innerOval, 180, 180, true, backgroundInnerPaint);

        //read about Bitmap.createScaledBitmap() by tapping on it
        Bitmap mask = Bitmap.createScaledBitmap(mMask, (int)(oval.width()*1.1),
(int)(oval.height()*1.1)/2, true);
        canvas.drawBitmap(mask, oval.centerX() - oval.width()*1.1f/2, oval.centerY()-
oval.width()*1.1f/2, maskPaint);
    }
    @SuppressWarnings("NewApi")
    private void init() {
        if (Build.VERSION.SDK_INT >= 11 && !isInEditMode())
            {setLayerType(View.LAYER_TYPE_HARDWARE, null);
        }

        backgroundPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
        backgroundPaint.setStyle(Paint.Style.FILL);
        backgroundPaint.setColor(Color.rgb(127, 127, 127));
```

```
        backgroundInnerPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
        backgroundInnerPaint.setStyle(Paint.Style.FILL);
        backgroundInnerPaint.setColor(Color.rgb(150, 150, 150));

        txtPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
        txtPaint.setColor(Color.WHITE);
        txtPaint.setTextSize(18);
        txtPaint.setTextAlign(Paint.Align.CENTER);

        mMask = BitmapFactory.decodeResource(getResources(), R.drawable.spot_mask);
        mMask = Bitmap.createBitmap(mMask, 0, 0, mMask.getWidth(), mMask.getHeight()/2);

        maskPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
        maskPaint.setDither(true); // read by just tapping on where this is used

        ticksPaint = new Paint(Paint.ANTI_ALIAS_FLAG);
        ticksPaint.setStrokeWidth(3.0f);
        ticksPaint.setStyle(Paint.Style.STROKE);
        ticksPaint.setColor(defaultColor);

        colorLinePaint = new Paint(Paint.ANTI_ALIAS_FLAG);
        colorLinePaint.setStyle(Paint.Style.STROKE);
        colorLinePaint.setStrokeWidth(5); // read by just tapping on where this is used
        colorLinePaint.setColor(defaultColor);

        needlePaint = new Paint(Paint.ANTI_ALIAS_FLAG);
        needlePaint.setStrokeWidth(5);
        needlePaint.setStyle(Paint.Style.STROKE);
        needlePaint.setColor(Color.argb(200, 255, 0, 0));
    }
    public static interface LabelConverter {

        String getLabelFor(double progress, double maxProgress);

        }
        }
}
```

CHAPTER 5

RESULT

5.1 Registration and login Page

Registration only happens the first time you access the system. It is a way to check your credentials. Every time after your initial registration, you will log on to the system using the email and password you created

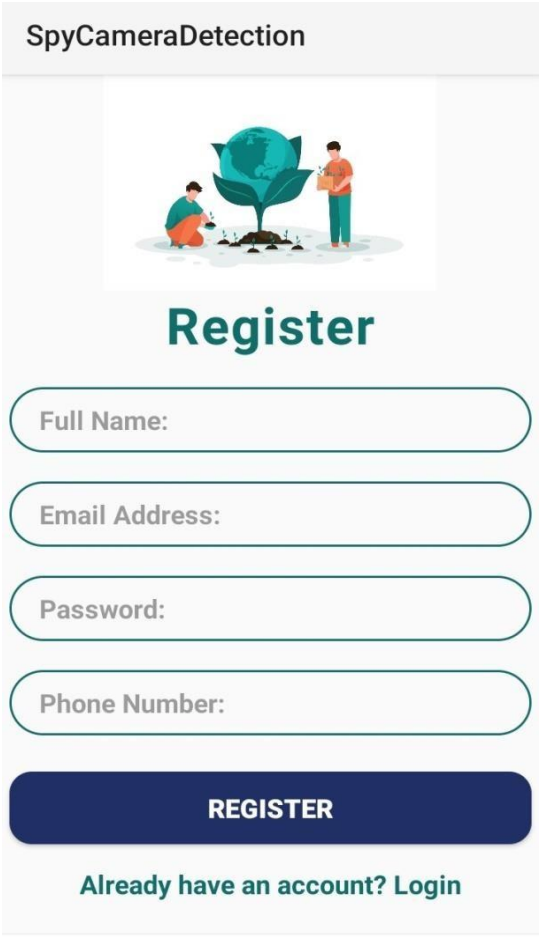

Fig 5.1.1:Registration Page

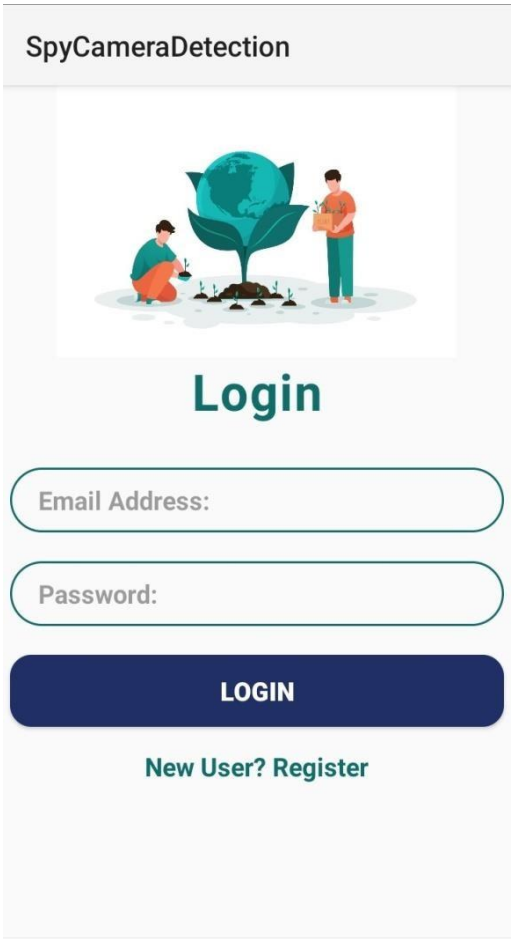

Fig 5.1.2:Login Page

5.2 Main Page

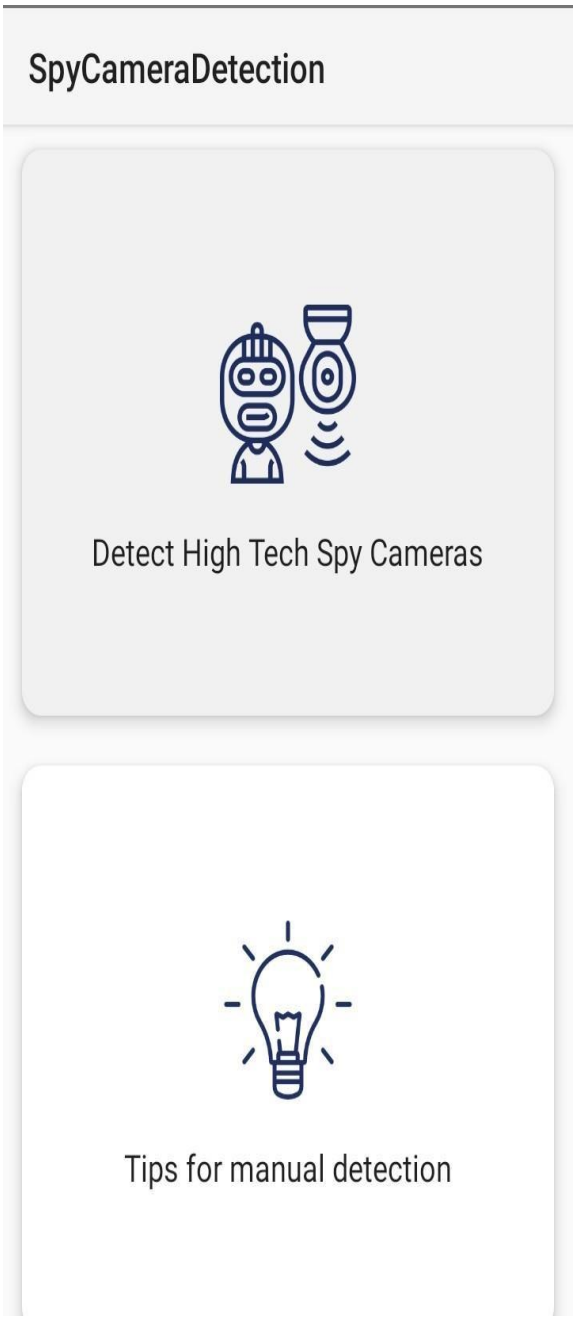

Fig 5.2.1: Main Page

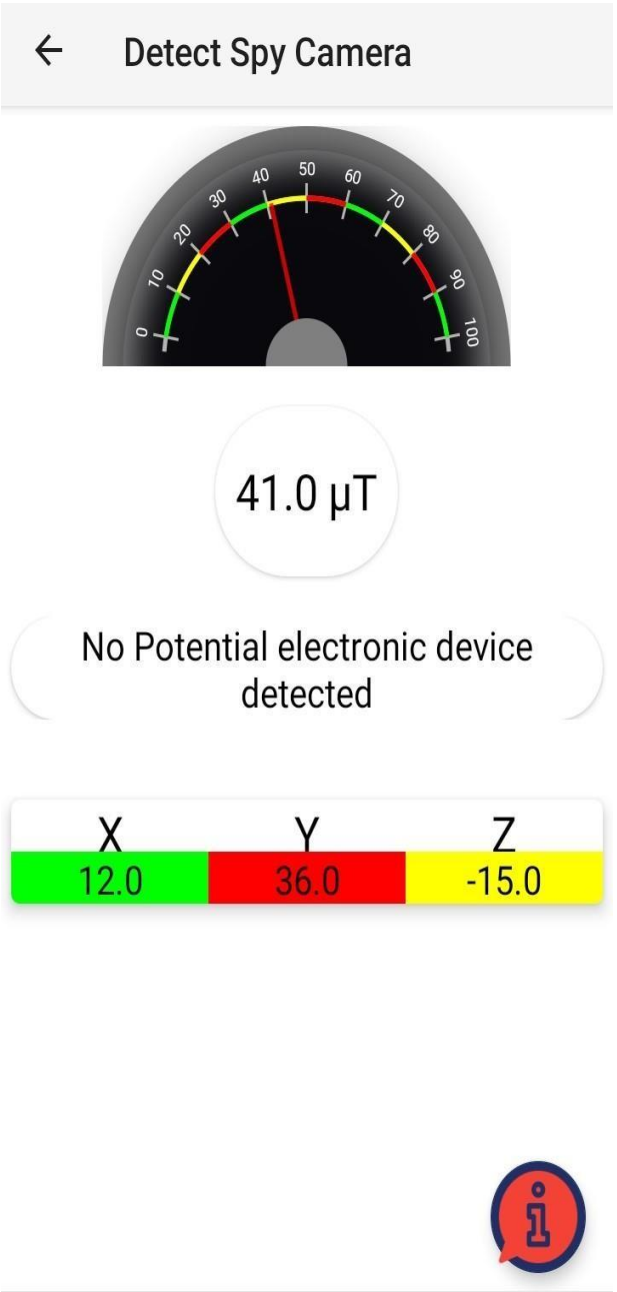

Fig 5.2.2: High Tech Spy camera

5.3 Features

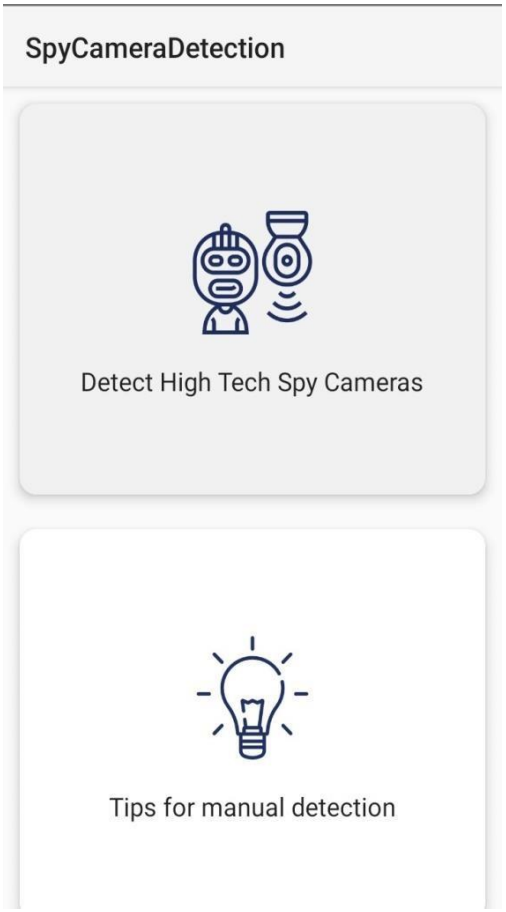

Fig 5.3.1: Main Page

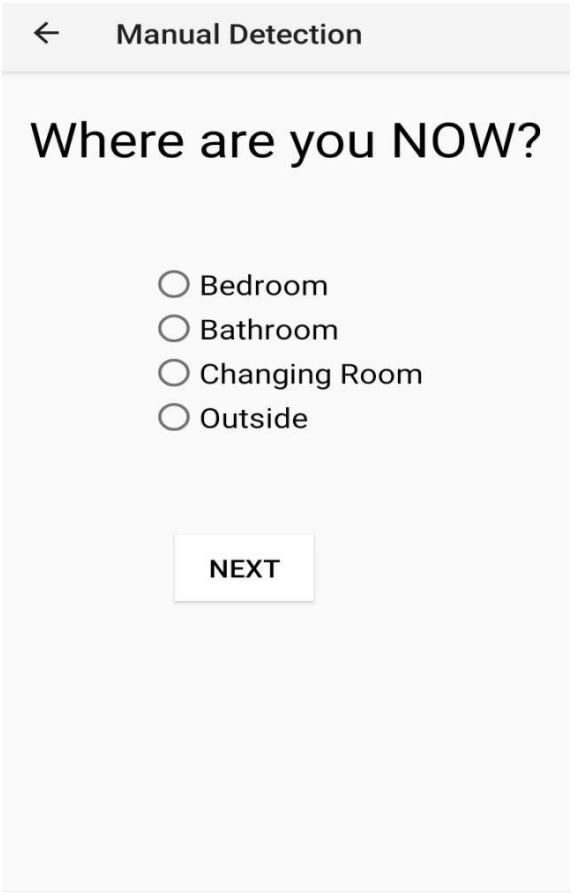

Fig 5.3.2: High Tech Spy camera features

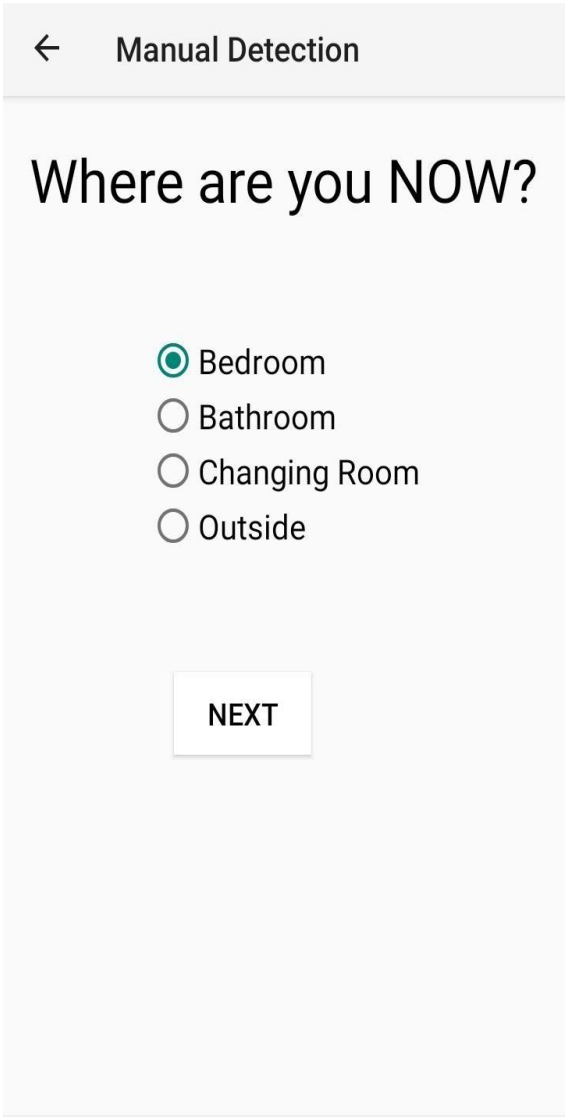

Fig 5.3.3: Bedroom Page

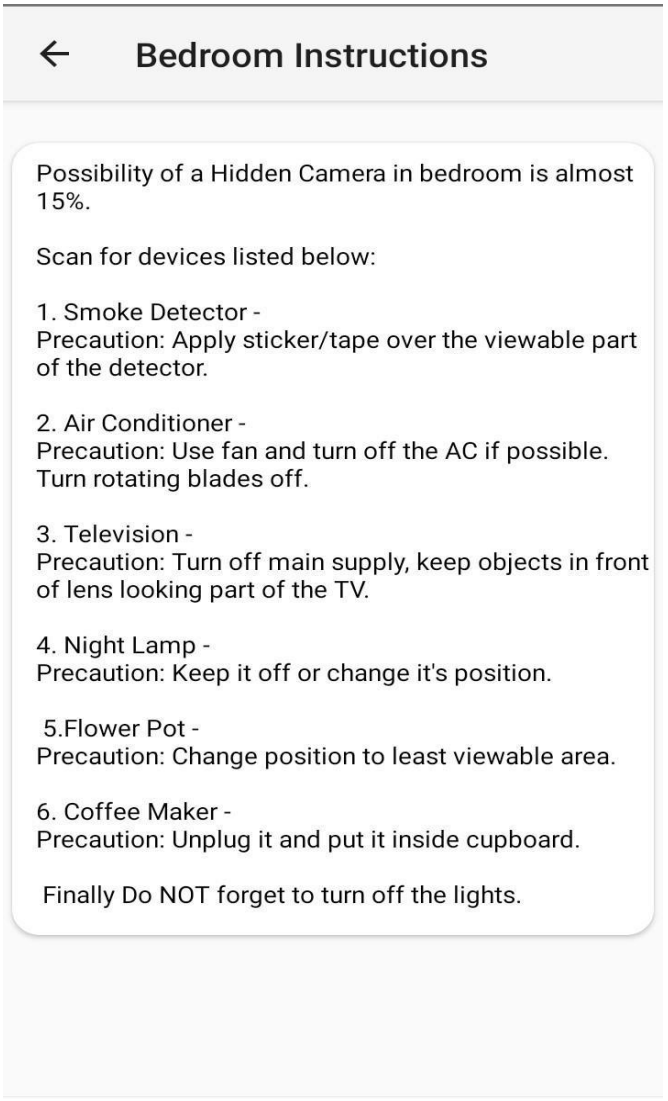

Fig 5.3.4 Bedroom Instruction

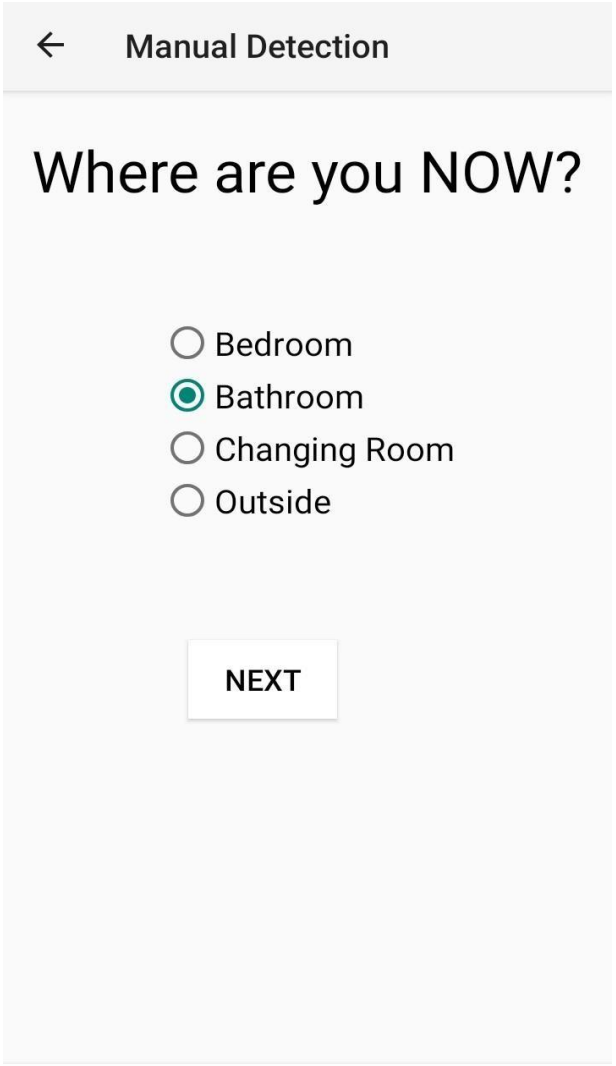

Fig 5.3.5: Bathroom Page

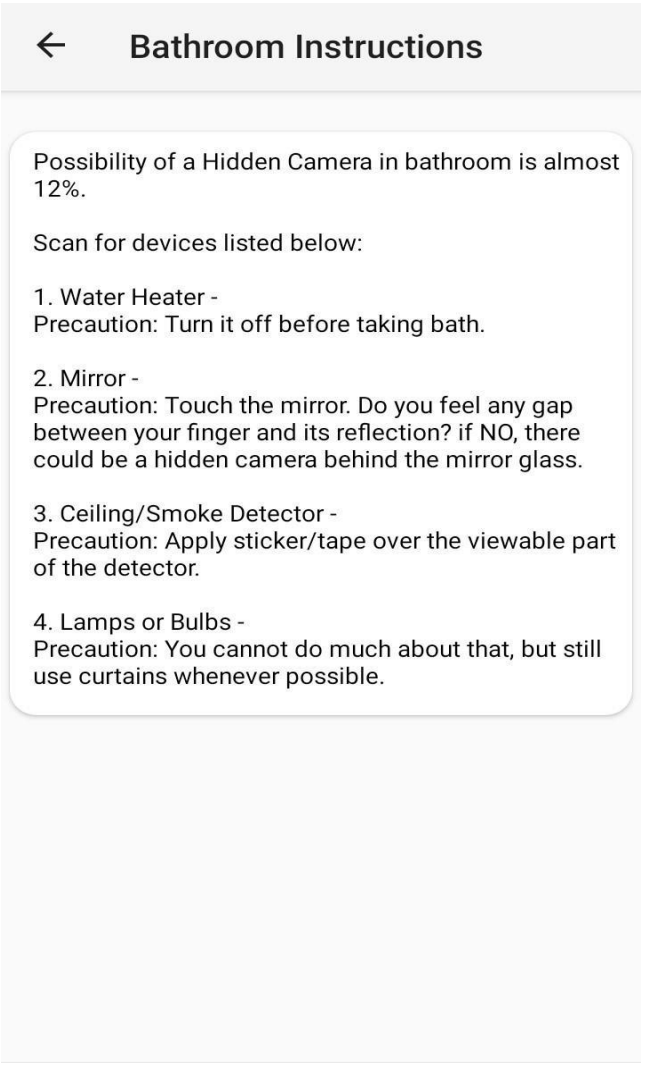

Fig 5.3.6 Bathroom Instruction

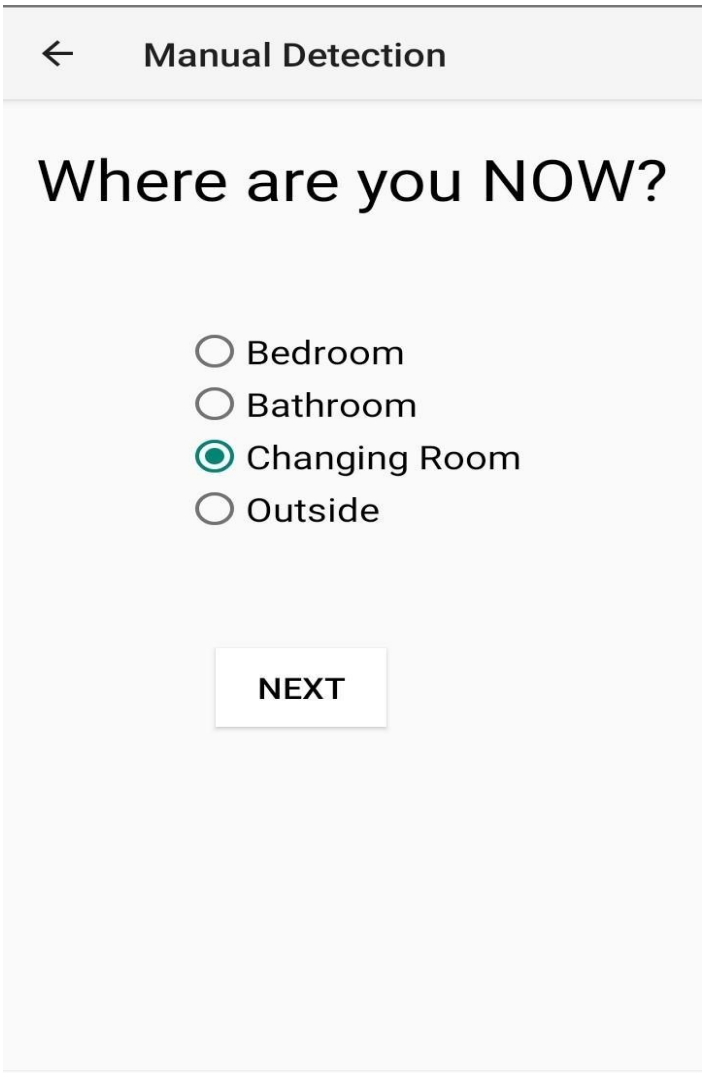

Fig 5.3.7: Changing Room Page

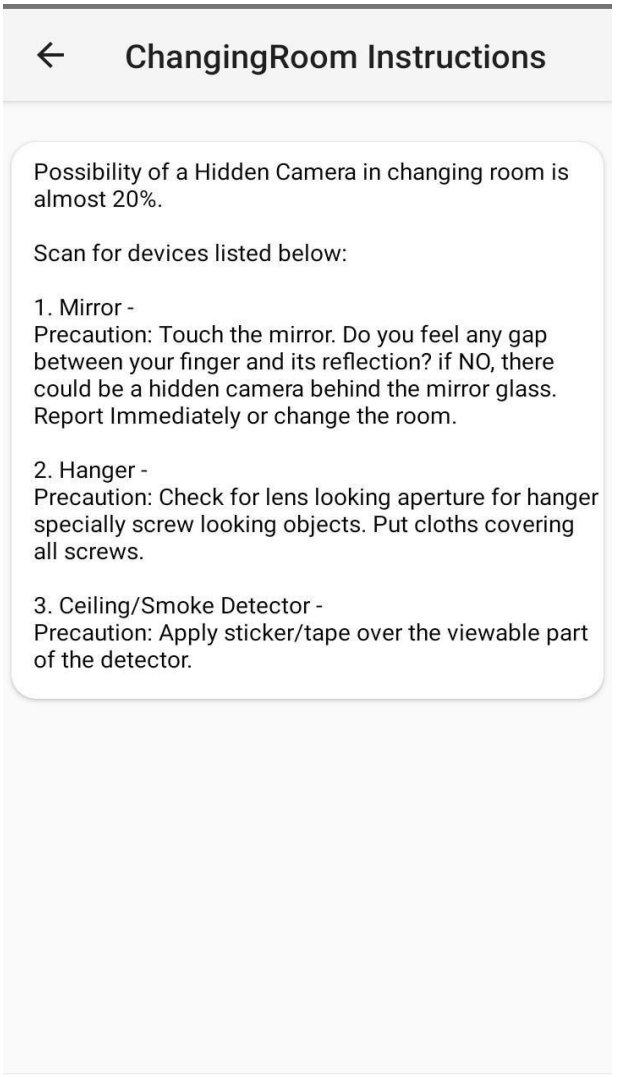

Fig 5.3.8:Changing Room Instruction

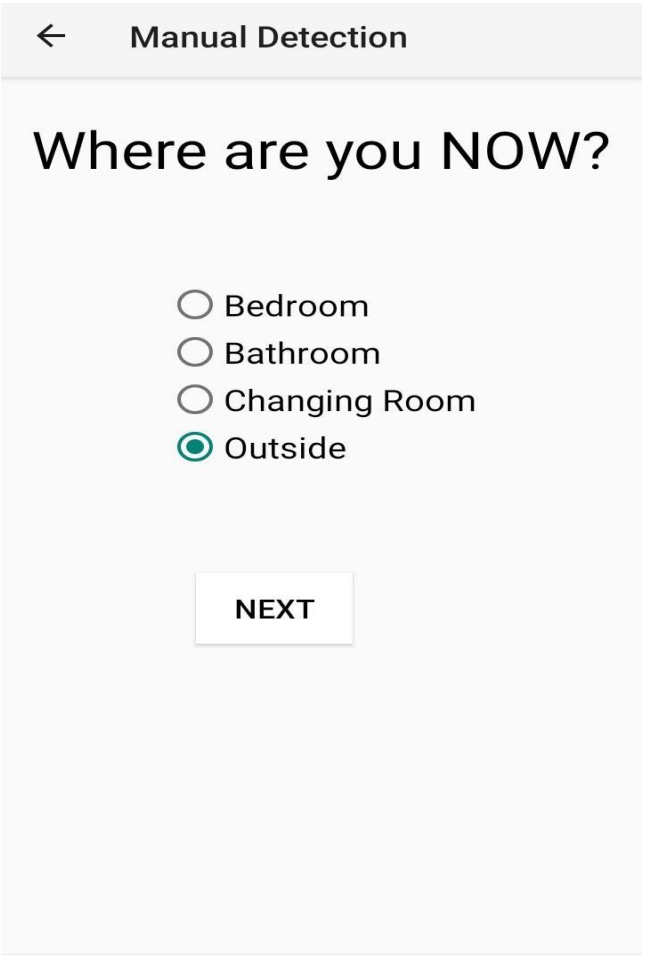

Fig 5.3.9:Outside Page

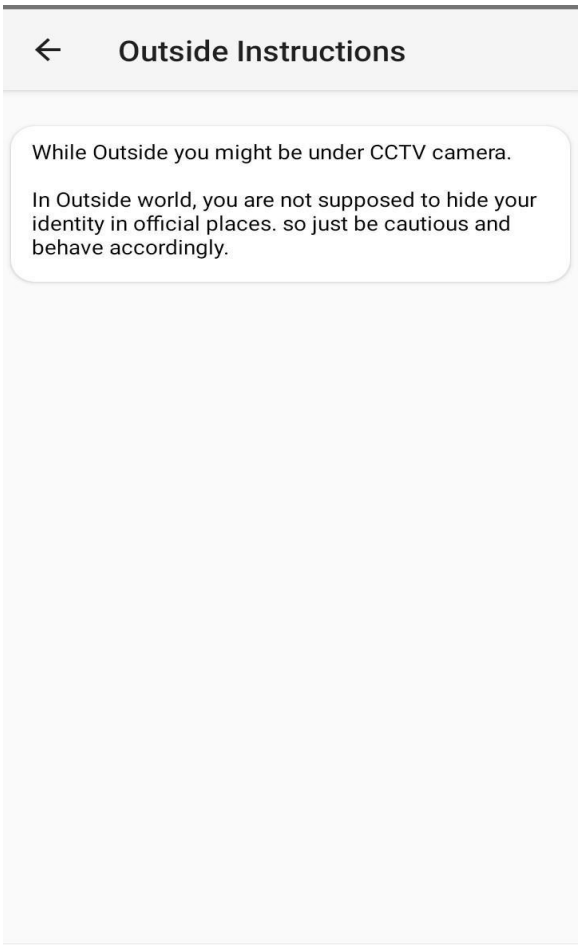

Fig 5.3.10:Outside Instruction

CHAPTER 6

CONCLUSION

Hidden camera detection would detect spy cameras in trial rooms, theatres and many other public places where it is prohibited with immediate report to the authorities after the detection. Manually checking their presence is almost impossible, the proposed system is an efficient method for camera detection because no matter how small the lenses are, they can be easily detected. In confidential meeting rooms, even after heavy checking people manage to skip the security and take cameras inside, record the conversations and use them for illegal purposes. So this application would be of great help for camera restricted areas.

REFERENCES

1. https://developer.android.com/studio/intro
2. www.w3schools.com
3. www.tutorialspoint.com
4. https://www.ijarcce.com/upload/2016/april-16/IJARCCE%20150.pdf