

**Vision:**

**Mission:**

## **LAB MANUAL**

### **Mobile Computing Lab**

**Name:**

**Class: TE Computer Engineering**

**Sem: VI      Roll No:**

## **CSL603**

### **Mobile Computing Lab**

#### **Lab Objective:**

After successful completion of this course student will be able to:

- 1 To learn the mobile computing tools and software for implementation.
- 2 To understand the security algorithms in mobile networks
- 3 To learn security concepts

#### **Description:**

Design and implementation of any case study/ applications /experiments / mini project based on departmental level courses using modern tools.

#### **Term work:**

The distribution of marks for term work shall be as follows:

Lab/ Experimental Work: 15

Report/ Documentation: 05

Attendance (Theory & Practical):

05

## **Lab Outcome**

At the end of the course, the students will be able to	
<b>1</b>	develop and demonstrate mobile applications using various tools
<b>2</b>	articulate the knowledge of GSM, CDMA & Bluetooth technologies and demonstrate it.
<b>3</b>	Students will be able to carry out simulation of frequency reuse, hidden/exposed terminal problem
<b>4</b>	implement security algorithms for mobile communication network
<b>5</b>	demonstrate simulation and compare the performance of Wireless LAN

Description: The software like Android Studio, J2ME, NS2, NS3 and any other software which is suitable are recommended for performing the practical.

## **List of Experiments**

Sr. No.	Title	Page No.
1	To understand the cellular frequency reuse concept to find the co-channel cells for a particular cell. Design a game-based application on the above concept	4
2	Implementation a Bluetooth network with application as transfer of a file from one device to another.	7
3	To implement a basic function of Code Division Multiple Access (CDMA) to test the orthogonality and autocorrelation of a code to be used for CDMA operation. Write an application based on the above concept.	28

4	To implement Mobile node discovery.	33
5	Implementation of GSM security algorithms (A3/A5/A8)	42
6	Illustration of Hidden Terminal Problem (NS-2)	50
7	Develop an application that uses GUI components.	54
8	Write an application that draws basic graphical primitives on the screen.	61
9	Develop an application that makes use of database.	66
10	Implement an application that creates an alert upon receiving a message.	74

## **Experiment No.: 1**

**Aim:** To understand the cellular frequency reuse concept to find the co-channel cells for a particular cell.

### **Theory:**

In mobile communication systems a slot of a carrier frequency / code in a carrier frequency is a radio resource unit. This radio resource unit is assigned to a user in order to support a call/ session. The number of available such radio resources at a base station thus determines the number of users who can be supported in the call. Since in wireless channels a signal is "broadcast" i.e. received by all entities therefore one a resource is allocated to a user it cannot be reassigned until the user finished the call/ session. Thus, the number of users who can be supported in a wireless system is highly limited.

In order to support a large no. of users within a limited spectrum in a region the concept of frequency re-use is used.

The signal radiated from the transmitter antenna gets attenuated with increasing distance. At a certain distance the signal strength falls below noise threshold and is no longer identifiable.

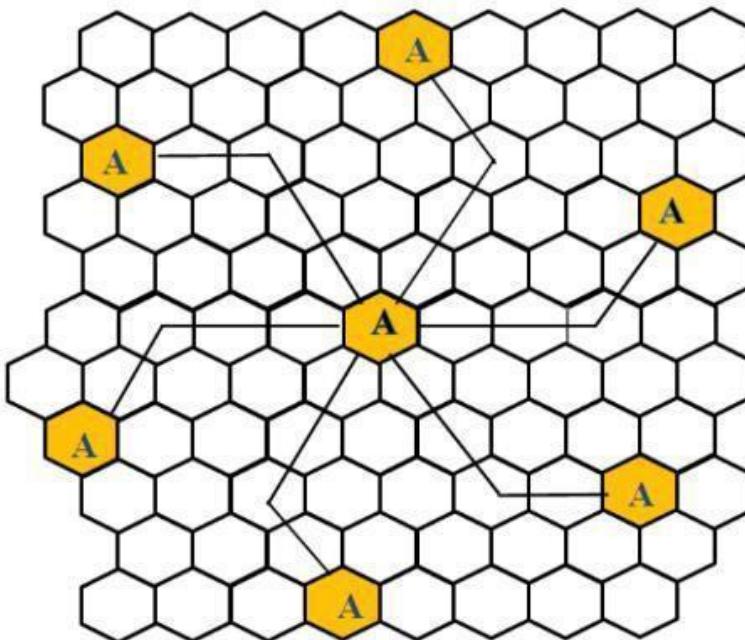
### **Cellular Frequency Reuse:**

Each cellular base station is allocated a group of radio channels to be used within a small geographic area called a cell. Base stations in adjacent cells are assigned channel groups which contain completely different channels than neighboring cells. Base station antennas are designed to achieve the desired coverage within a particular cell. By limiting the coverage area within the boundaries of a cell, the same group of channels may be used to cover different cells that are separated from one another by geographic distances large enough to keep interference levels within tolerable limits. The design process of selecting and allocating channel groups for all cellular base stations within a system is called frequency reuse or frequency planning.

### **Co-channel Cells:**

A larger cluster size causes the ratio between the cell radius and the distance between co-channel cells to decrease reducing co-channel interference. The value of N is a function of how much interference a mobile or base station can tolerate while maintaining a sufficient quality of communications. Since each hexagonal cell has six equidistant neighbors and the line joining the centers of any cell and each of its neighbors are separated by multiples of 60 degrees, only certain cluster sizes and cell layouts are possible. To connect without gaps between adjacent cells, the geometry of hexagons is such that the numbers of cells per cluster, N, can only have values that satisfy,

$$N = i^2 + ij + j^2$$



Method of locating co-channel cells in a cellular system. In this figure,  $N=19$ (i.e,  $i=3, j=2$ ).

In this example  $N = 19$  (i.e.,  $i = 3, j = 2$ ).

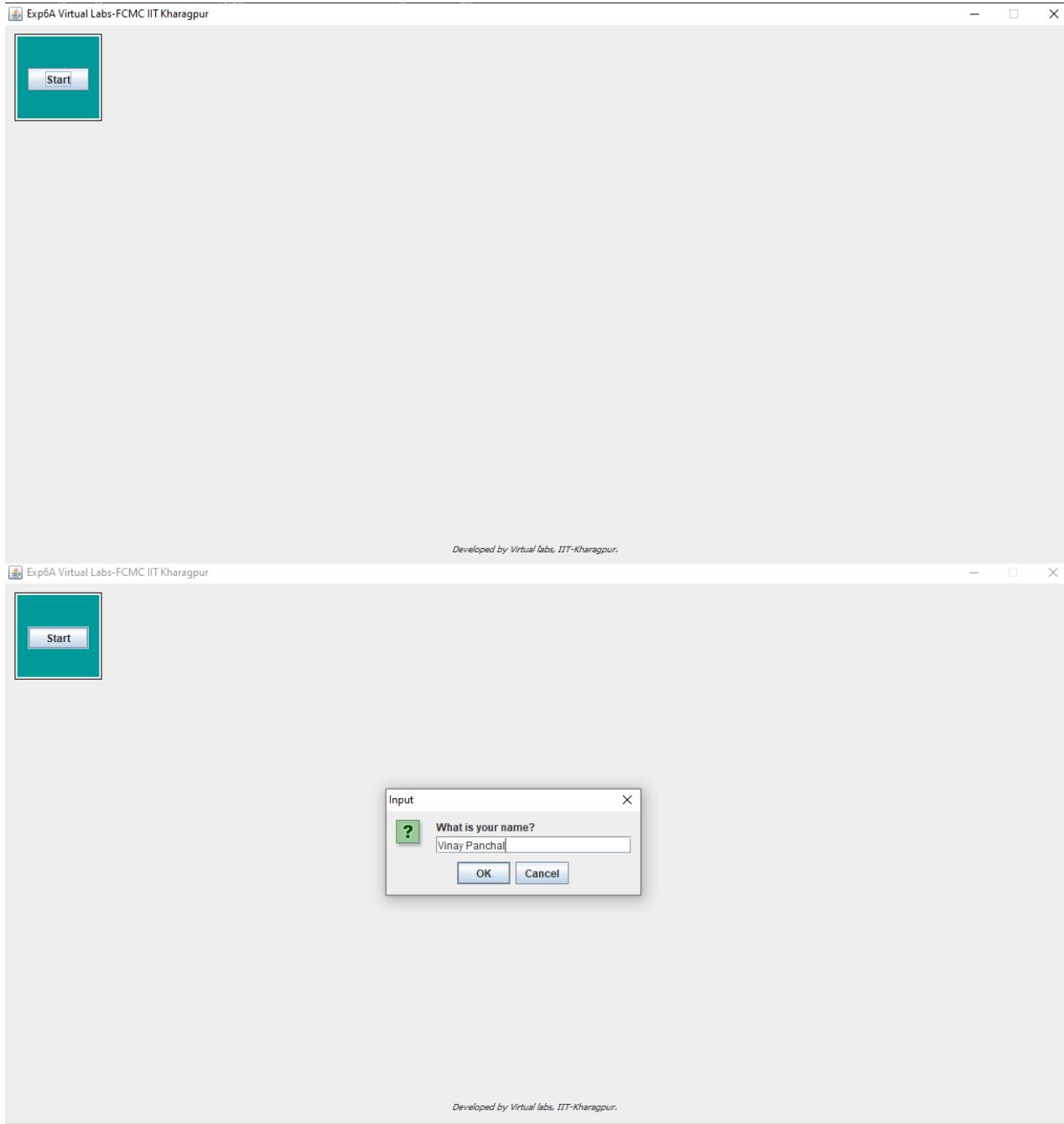
Where,

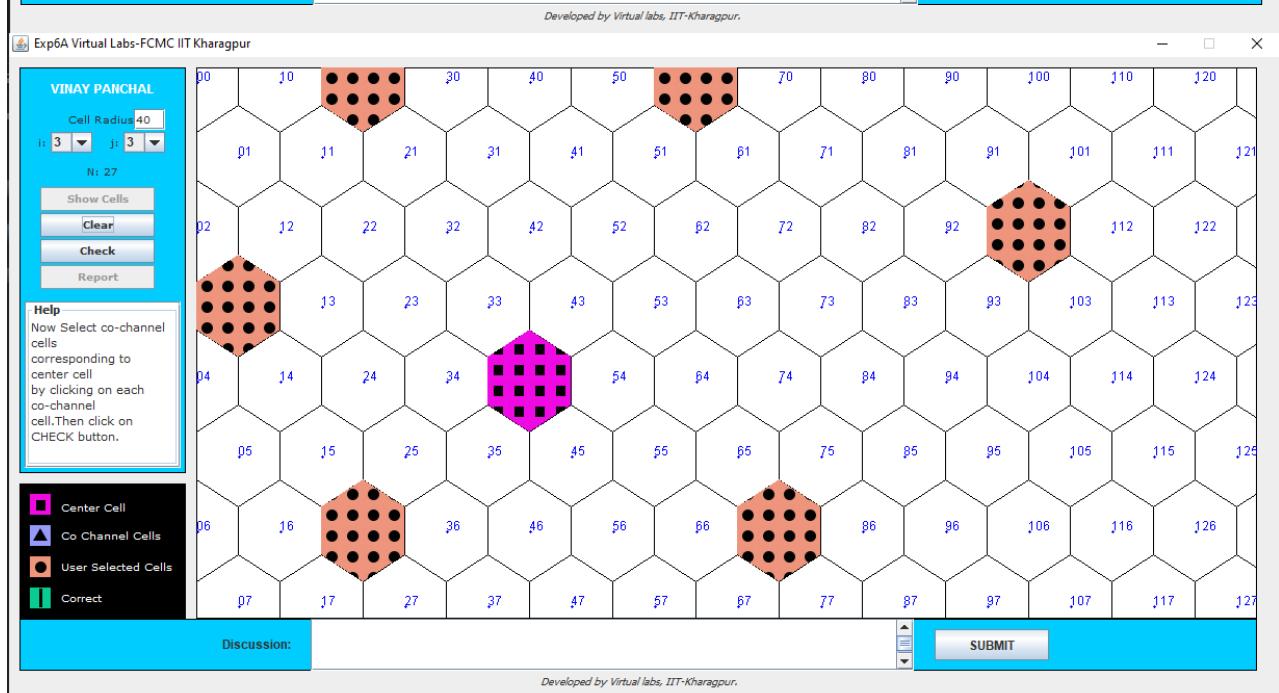
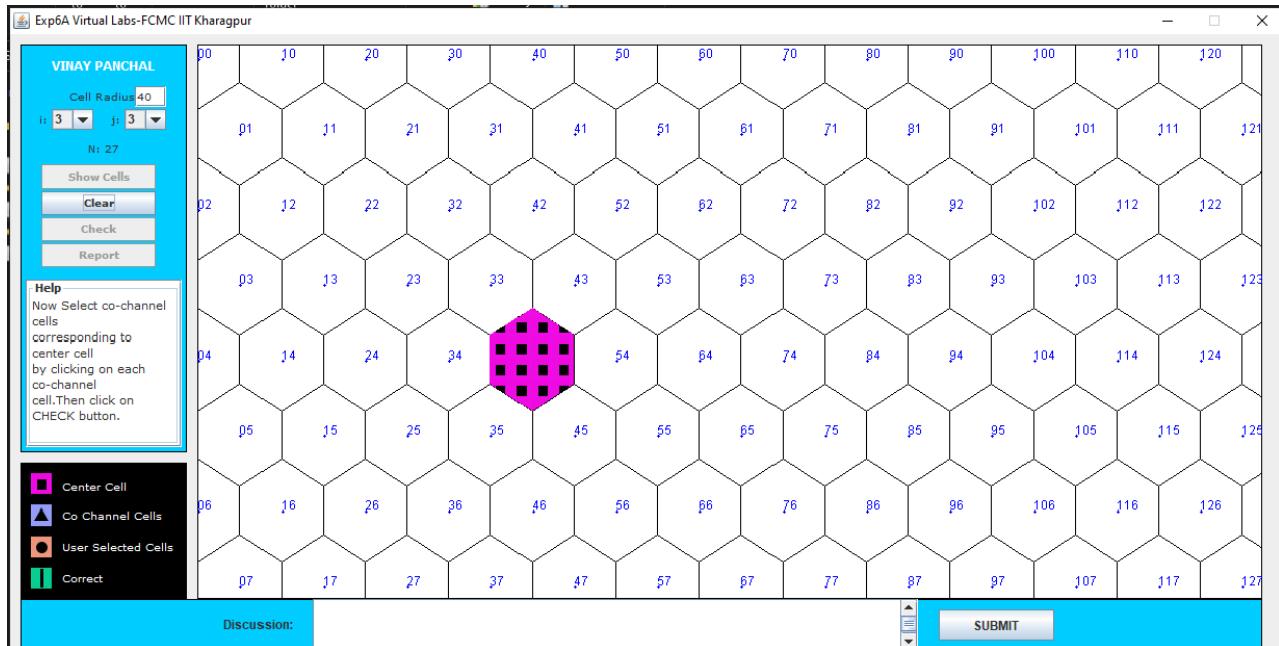
$i$  and  $j$  are non-negative integers.

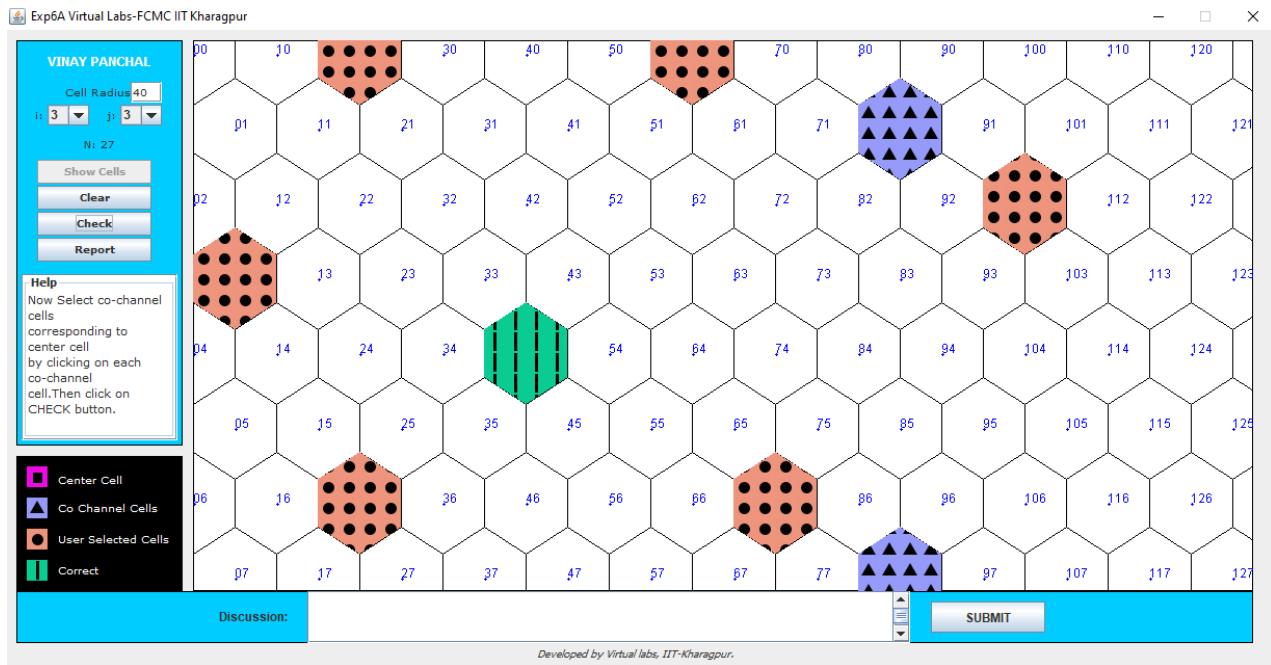
To find the nearest co-channel neighbours of a particular cell,

- a. move  $i$  cells along any chain of hexagons then,
- b. turn 60 degrees counter-clockwise and move  $j$  cells.

## Output:







## Conclusion:

Thus, we have performed the frequency reuse experiment wherein we select a hexagon with particular frequency and then select other hexagons where the after which frequency can be used using above formula, we performed the experiment properly.

## **Experiment No.: 2**

**Aim:** To implement a Bluetooth network with application as transfer of a file from one device to another.

### **Theory:**

Bluetooth is a wireless technology standard used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in the industrial, scientific and medical radio bands, from 2.402 GHz to 2.480 GHz, and building personal area networks (PANs). It was originally conceived as a wireless alternative to RS-232 data cables.

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 35,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE standardized Bluetooth as IEEE 802.15.1, but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks. A manufacturer must meet Bluetooth SIG standards to market it as a Bluetooth device.

Transfer of words between two phones using Bluetooth is done below.

### **Code:**

#### **Main\_Activity.java:**

<https://github.com/vinaynpp/mcc>

```
package com.example.bluetooth_communication; import android.app.Dialog; import android.bluetooth.BluetoothAdapter; import android.content.Intent; import android.content.pm.PackageManager; import android.content.pm.ResolveInfo; import android.net.Uri; import android.os.Environment; import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle; import android.view.Menu; import android.view.MenuItem; import android.view.View; import android.widget.AdapterView; import android.widget.ArrayAdapter; import android.widget.Button;
```

```
import android.widget.EditText;  
import android.widget.ListView;
```

```
import android.widget.TextView;
import android.widget.Toast;

import java.io.File; import
java.util.ArrayList; import
java.util.List;

public class MainActivity extends AppCompatActivity {
    //Create Objects-----
    Button buttonopenDailog, buttonUp, send;
    TextView textFolder;
    EditText dataPath;
    static final int CUSTOM_DIALOG_ID = 0;
    ListView dialog_listView;
    File root, fileroot, curFolder;
    private List<String> fileList = new ArrayList<String>();
    private static final int DISCOVER_DURATION = 300;
    private static final int REQUEST_BLU = 1;
    BluetoothAdapter btAdatper = BluetoothAdapter.getDefaultAdapter();
    //-----

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dataPath=(EditText)findViewById(R.id.filePath);
        buttonopenDailog= (Button) findViewById(R.id.opendialog);
        send=(Button)findViewById(R.id.sendBtooth);
        buttonopenDailog.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                dataPath.setText("");
                showDialog(CUSTOM_DIALOG_ID);
            }
        });
        root = new File(Environment.getExternalStorageDirectory().getAbsolutePath());
        curFolder = root;
        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                sendViaBluetooth();
            }
        });
    }
}
```

```

@Override
protected Dialog onCreateDialog(int id) {
    Dialog dialog = null;
    switch (id) {
        case CUSTOM_DIALOG_ID:
            dialog = new Dialog(MainActivity.this);
            dialog.setContentView(R.layout.dailoglayout);
            dialog.setTitle("File Selector");
            dialog.setCancelable(true);
            dialog.setCanceledOnTouchOutside(true);
            textViewFolder = (TextView) dialog.findViewById(R.id.folder);
            buttonUp = (Button) dialog.findViewById(R.id.up);
            buttonUp.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    ListDir(curFolder.getParentFile());
                }
            });
            dialog.ListView = (ListView) dialog.findViewById(R.id.dialoglist);
            dialog.ListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
                @Override
                public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
                    File selected = new File(fileList.get(position));
                    if (selected.isDirectory()) {
                        ListDir(selected);
                    } else if (selected.isFile()) {
                        getSelectedFile(selected);
                    } else {
                        dismissDialog(CUSTOM_DIALOG_ID);
                    }
                }
            });
            break;
    }
    return dialog;
}

@Override
protected void onPrepareDialog(int id, Dialog dialog) {
    super.onPrepareDialog(id, dialog);
    switch (id) {
        case CUSTOM_DIALOG_ID:
            ListDir(curFolder);
            break;
    }
}

```

}

```

}

public void getselectedFile(File f){
    dataPath.setText(f.getAbsolutePath());
    fileList.clear();
    dismissDialog(CUSTOM_DIALOG_ID);
}

public void ListDir(File f) {
    if (f.equals(root)) {
        buttonUp.setEnabled(false);
    } else {
        buttonUp.setEnabled(true);
    }
    curFolder = f;
    textFolder.setText(f.getAbsolutePath());
    dataPath.setText(f.getAbsolutePath());
    File[] files = f.listFiles();
    fileList.clear();

    for (File file : files) {
        fileList.add(file.getPath());
    }
    ArrayAdapter<String> directoryList = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, fileList);
    dialog_ListView.setAdapter(directoryList);
}

//exit to application-----
public void exit(View V) {
    btAdatper.disable();
    Toast.makeText(this,"*** Now Bluetooth is off... Thanks.
***",Toast.LENGTH_LONG).show();
    finish(); }

//Method for send file via bluetooth-----
public void sendViaBluetooth() {
    if(!dataPath.equals(null)){
        if(btAdatper == null) {
            Toast.makeText(this, "Device not support bluetooth", Toast.LENGTH_LONG).show();
        } else {
            enableBluetooth();
        }
    }
}

```

**}else{**

```

        Toast.makeText(this, "Please select a file.", Toast.LENGTH_LONG).show();
    }
}

public void enableBluetooth() {
    Intent discoveryIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
    discoveryIntent.putExtra(BluetoothAdapter.EXTRA_DISCOVERABLE_DURATION,
DISCOVER_DURATION);
    startActivityForResult(discoveryIntent, REQUEST_BLU);
}

//Override method for sending data via bluetooth availability-----
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode == DISCOVER_DURATION && requestCode == REQUEST_BLU) {
        Intent i = new Intent();
        i.setAction(Intent.ACTION_S
END); i.setType("/*");
        File file = new File(dataPath.getText().toString());

        i.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(file));

        PackageManager pm = getPackageManager();
        List<ResolveInfo> list = pm.queryIntentActivities(i, 0);
        if (list.size() > 0) {
            String packageName = null;
            String className = null;
            boolean found = false;

            for (ResolveInfo info : list) {
                packageName = info.activityInfo.packageName;
                if (packageName.equals("com.android.bluetooth")) {
                    className = info.activityInfo.name;
                    found = true;
                    break;
                }
            }
        }
        //CHECK BLUETOOTH available or not-----
        if (!found) {
            Toast.makeText(this, "Bluetooth not been found", Toast.LENGTH_LONG).show();
        } else {
            i.setClassName(packageName,
className); startActivity(i);
        }
    }
}

```



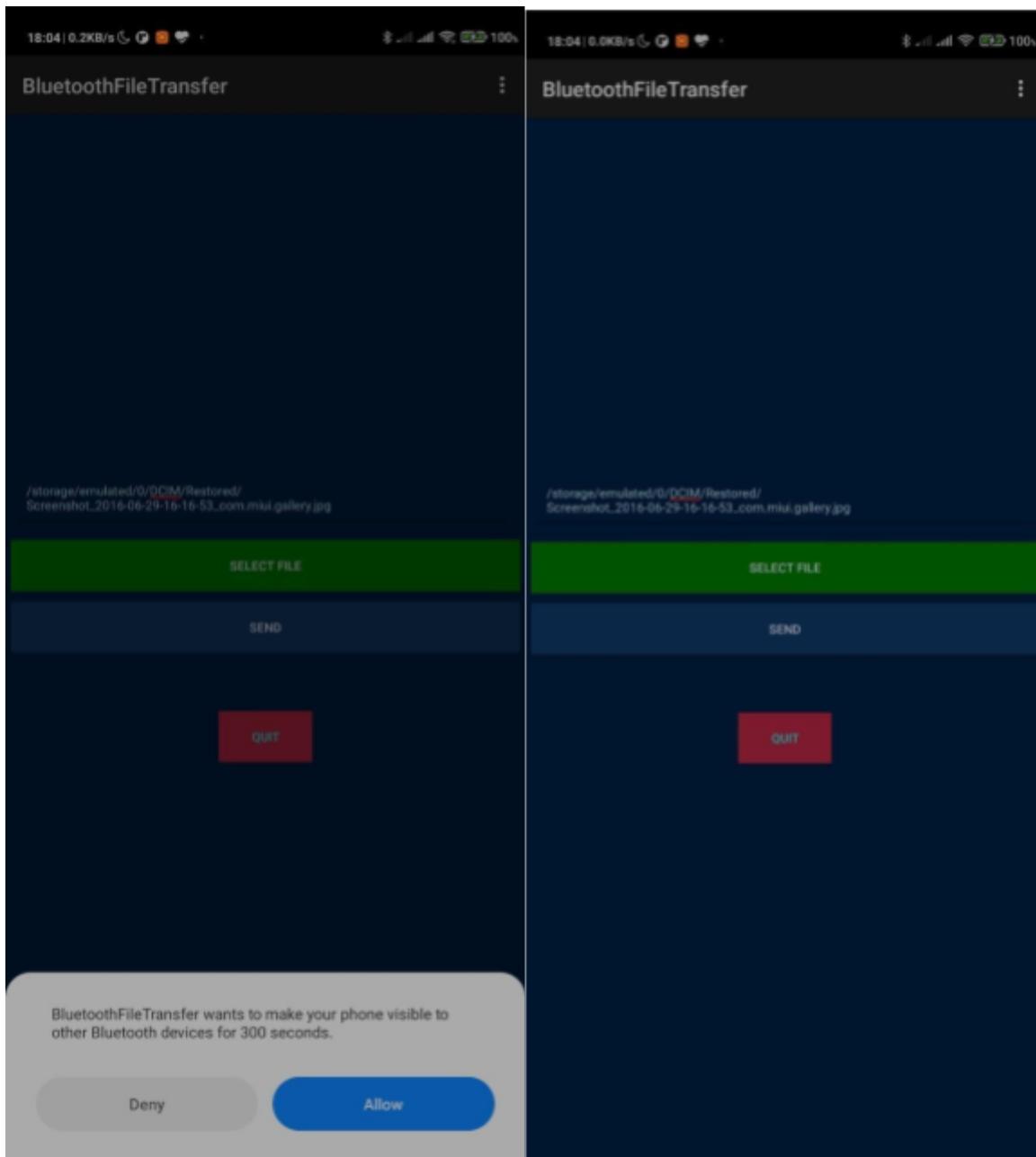
```
        }
    } else {
        Toast.makeText(this, "Bluetooth is cancelled", Toast.LENGTH_LONG).show();
    }
}

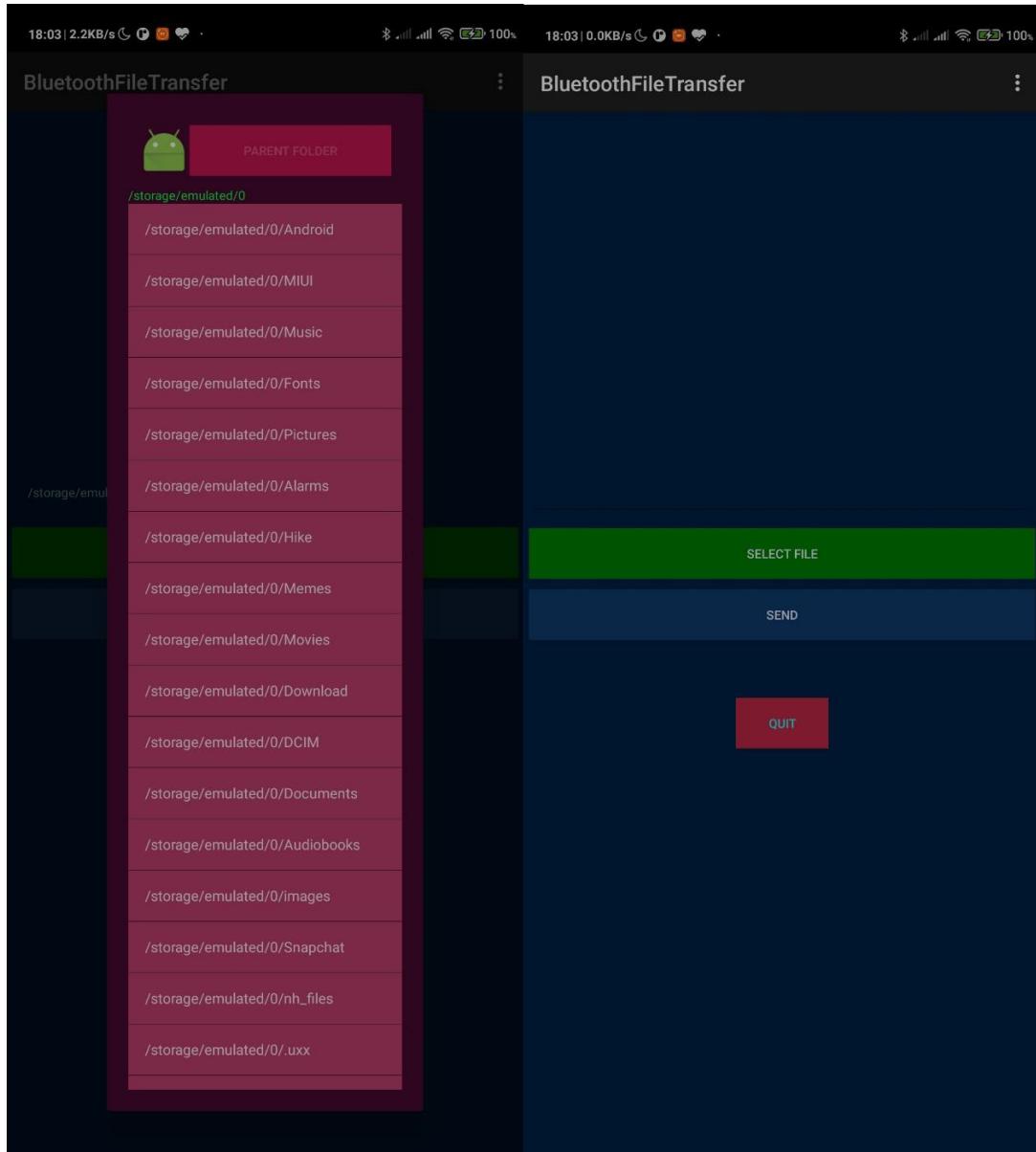
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        Toast.makeText(this, "*****\nDeveloper: Santosh Kumar Singh\nContact:\nsuperssingh@gmail.com\n*****", Toast.LENGTH_LONG).show();
        return true;
    }
    return super.onOptionsItemSelected(item);
}
}
```

## Output:





## Conclusion:

Thus, we have performed successfully the experiment of transferring data between two mobile phone using Bluetooth network and after that have checked and it performed.

## Experiment No.: 3

**Aim:** To implement a basic function of Code Division Multiple Access (CDMA) to test the orthogonality and autocorrelation of a code to be used for CDMA operation. Write an application based on the above concept.

### Theory:

Code-division multiple access (CDMA) is a channel access method used by various radio communication technologies. CDMA is an example of multiple access, where several transmitters can send information simultaneously over a single communication channel. This allows several users to share a band of frequencies (see bandwidth). To permit this without undue interference between the users, CDMA employs spread spectrum technology and a special coding scheme (where each transmitter is assigned a code).

CDMA is used as the access method in many mobile phone standards. IS-95, also called "cdmaOne", and its 3G evolution CDMA2000, are often simply referred to as "CDMA", but UMTS, the 3G standard used by GSM carriers, also uses "wideband CDMA", or W-CDMA, as well as TD-CDMA and TD-SCDMA, as its radio technologies.

The intended 4G successor to CDMA2000 was UMB (Ultra Mobile Broadband); however, in November 2008, Qualcomm announced it was ending development of the technology, favoring LTE instead

#### CDMA Orthogonality:

Techniques generally used are direct sequence spread spectrum modulation (DS-CDMA), frequency hopping or mixed CDMA detection (JD-CDMA). Here, a signal is generated which extends over a wide bandwidth. A code called spreading code is used to perform this action. Using a group of codes, which are orthogonal to each other, it is possible to select a signal with a given code in the presence of many other signals with different orthogonal codes.

#### CDMA Autocorrelation:

Autocorrelation of the sequence, it determines the ability to synchronize and lock the spreading code for the received signal.

### Code:

```
import numpy as np
c1=[1,1,1,1]
c2=[1,-1,1,-1]
c3=[1,1,-1,-1]
```

<https://github.com/vinaynpp/mcc>

```

c4=[1,-1,-1,1]
rc=[]

print("Enter the data bits :")

d1=int(input("Enter D1 :"))
d2=int(input("Enter D2 :"))
d3=int(input("Enter D3 :"))
d4=int(input("Enter D4 :"))

r1=np.multiply(c1,d1)
r2=np.multiply(c2,d2)
r3=np.multiply(c3,d3)
r4=np.multiply(c4,d4)
resultant_channel=r1+r2+r3+r4;
print("Resultant Channel",resultant_channel)
Channel=int(input("Enter the station to listen for C1=1 ,C2=2, C3=3 C4=4 :"))

if Channel==1:
    rc=c1
elif Channel==2:
    rc=c2
elif Channel==3:
    rc=c3
elif Channel==4:
    rc=c4
inner_product = np.multiply(resultant_channel,rc)

print("Inner Product",inner_product)
res1=sum(inner_product)

data = res1/len(inner_product)
print("Data bit that was sent",data)
https://github.com/vinaynpp/mcc

```

## Output:

```
.. Enter the data bits :  
Enter D1 :23  
Enter D2 :5  
Enter D3 :456  
Enter D4 :56  
Resultant Channel [ 540  418 -484 -382]  
Enter the station to listen for C1=1 ,C2=2, C3=3 C4=4 : 1  
Inner Product [ 540  418 -484 -382]  
Data bit that was sent 23.0
```

### **Conclusion:**

Thus, we have studied the CDMA code to test autocorrelation and orthogonality of codes and executed the same using the java code as above and got proper output for it.



### Experiment No.:

**4 Aim:** To implement Mobile node discovery

#### **Theory:**

The mobile node is an end system or device such as a cell phone, PDA (Personal Digital assistant), or laptop whose software enables network roaming capabilities. A mobile node is an Internet-connected device whose location and point of attachment to the Internet may frequently be changed. This kind of node is often a cellular telephone or handheld or laptop computer, although a mobile node can also be a router. Special support is required to maintain Internet connections for a mobile node as it moves from one network or subnet to another, because traditional Internet routing assumes a device will always have the same IP address. Therefore, using standard routing procedures, a mobile user would have to change the device's IP address each time they connected through another network or subnet.

Since mobility and ease of connection are crucial considerations for mobile device users, organizations that want to promote mobile communications are putting a great deal of effort into making mobile connection and uncomplicated for the user. The Internet Engineering Task Force (IETF) Mobile IP working group has developed several standards or proposed standards to address these needs, including Mobile IP and later enhancements, Mobile IP version 6 (MIPv6) and Hierarchical Mobile IP version 6 (HMIPv6).

#### **Code:**

<https://github.com/vinaynpp/mcc>  
package BtNodeDiscovery;

```
import java.io.IOException;
import javax.bluetooth.*;

import javax.bluetooth.DiscoveryListener; import javax.microedition.lcdui.*; import javax.microedition.midlet.*;

public class discover_device extends MIDlet implements CommandListener,DiscoveryListener {

    private final List deviceList;
    private final Command Exit,Refresh; private String deviceName; private DiscoveryAgent agent; private Alert dialog;
    public discover_device()
    {
        deviceList = new List("List of Devices",List.IMPLICIT);
        Exit= new Command("Exit",Command.EXIT, 0);
        Refresh = new
        Command("Refresh",Command.SCREEN, 1);

        deviceList.addCommand(Exit); deviceList.addCommand(Refresh);
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
deviceList.setCommandListener(this); Display.getDisplay(this).setCurrent(deviceList);
}

public void startApp() {
try {
deviceList.deleteAll();

LocalDevice local = LocalDevice.getLocalDevice(); local.setDiscoverable(DiscoveryAgent.GIAC); deviceName =
local.getFriendlyName();

agent = local.getDiscoveryAgent(); }
catch (BluetoothStateException ex) { ex.printStackTrace(); }
try {
agent.startInquiry(DiscoveryAgent.GIAC, this);

}
catch (BluetoothStateException ex) { ex.printStackTrace(); }

}

public void pauseApp() {
}
public void destroyApp(boolean unconditional) {
}

public void commandAction(Command c, Displayable
d) {
if(c==Exit)
{
this.destroyApp(true);

notifyDestroyed();
}
if(c==Refresh){
this.startApp();
}
}

public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) { String deviceaddress = null;
try {
deviceaddress =

btDevice.getBluetoothAddress();//btDevice. getFriendl

} catch (Exception ex) { ex.printStackTrace();
}
deviceList.insert(0, deviceaddress , null);
}
```



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

```
public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {  
    throw new UnsupportedOperationException("Not supported yet.");  
}  
public void serviceSearchCompleted(int transID, int respCode) {  
    throw new UnsupportedOperationException("Not supported yet.");  
}  
  
public void inquiryCompleted(int discType) { Alert dialog = null;  
if (discType !=  
DiscoveryListener.INQUIRY_COMPLETED) {  
  
dialog = new Alert("Bluetooth Error","The inquiry failed to complete normally",null,AlertType.ERROR);  
}  
else {  
  
dialog = new Alert("Inquiry Completed","The inquiry completed normally", null,AlertType.INFO);  
}  
dialog.setTimeout(500);  
  
Display.getDisplay(this).setCurrent(dialog);  
}  
  
package BtNodeDiscovery;  
  
  
  
import javax.microedition.midlet.*;  
  
import javax.microedition.lcdui.*;  
  
import javax.microedition.io.*;  
  
import javax.bluetooth.*;  
  
import java.util.*;  
/**  
 * @author VYD  
 */  
public class Blue extends MIDlet implements  
    CommandListener, DiscoveryListener {  
  
private List activeDevices;  
private Command select, exit;  
private Display display;  
private LocalDevice local = null;  
private DiscoveryAgent agent = null;  
private Vector devicesFound = null;
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
private ServiceRecord[] servicesFound = null;
private String connectionURL = null;

public void startApp() {

    display = Display.getDisplay(this);

    activeDevices = new List("Active Devices",
        List.IMPLICIT); select = new Command("Search Again",
        Command.OK, 0);

    exit = new Command("Exit",
        Command.EXIT, 0);

    activeDevices.addCommand(exit);

    activeDevices.setCommandListener(this);
    try {

        local = LocalDevice.getLocalDevice();
    } catch (Exception e) {
    }

    doDeviceDiscovery();

    display.setCurrent(activeDevices);

}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {
    notifyDestroyed();
}

public void commandAction(Command cmd, Displayable disp) {

    if (cmd == select && disp == activeDevices) {

        activeDevices.deleteAll();

        doDeviceDiscovery();

    }

    if (cmd == exit) {

        destroyApp(false);
    }
}
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
public void inquiryCompleted(int param)
{ try {
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
switch (param) {  
    case DiscoveryListener.INQUIRY_COMPLETED:  
        if (devicesFound.size() > 0) {  
            activeDevices.addCommand(select);  
            activeDevices.setSelectCommand(select);  
        } else {  
            activeDevices.append("No Devices Found", null);  
        }  
        break;  
    }  
}  
}  
public void serviceSearchCompleted(int transID, int respCode) {  
}  
public void servicesDiscovered(int transID, ServiceRecord[] serviceRecord) {  
}  
public void deviceDiscovered(RemoteDevice remoteDevice, DeviceClass deviceClass) {  
    String str = null;  
    try {  
        str = remoteDevice.getBluetoothAddress() + " - ";  
        str += remoteDevice.getFriendlyName(true);  
    } catch (Exception e) {  
    }  
    activeDevices.append(str, null);  
    devicesFound.addElement(remoteDevice);  
    try {  
        if (!agent.startInquiry(DiscoveryAgent.GIAC, this)) {  
        }  
    } catch (BluetoothStateException e) {  
    }  
//    TODO Auto-generated catch block
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

e.printStackTrace();



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

```
        }  
    }  
  
private void doDeviceDiscovery() {  
  
    try {  
        local = LocalDevice.getLocalDevice();  
        agent = local.getDiscoveryAgent();  
        devicesFound = new Vector();  
    } catch (Exception e) {  
    }  
}  
}https://github.com/vinaynpp/mcc
```

**Output:**



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC



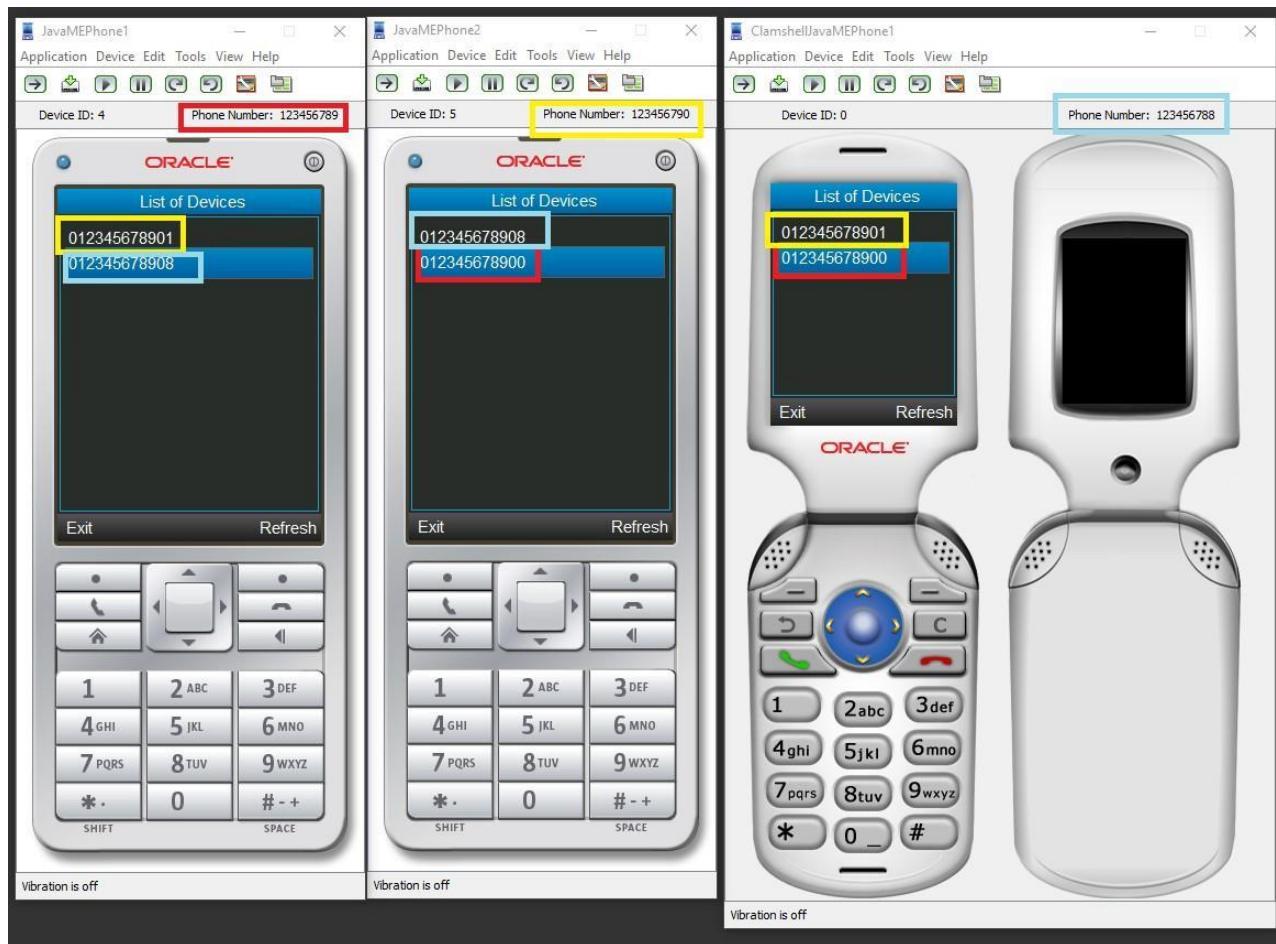


**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**





**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**



### Conclusion:

Thus, we have studied the mobile node discovery experiment properly and executed using the java code and after successfully running the code we have checked the output and it was proper.



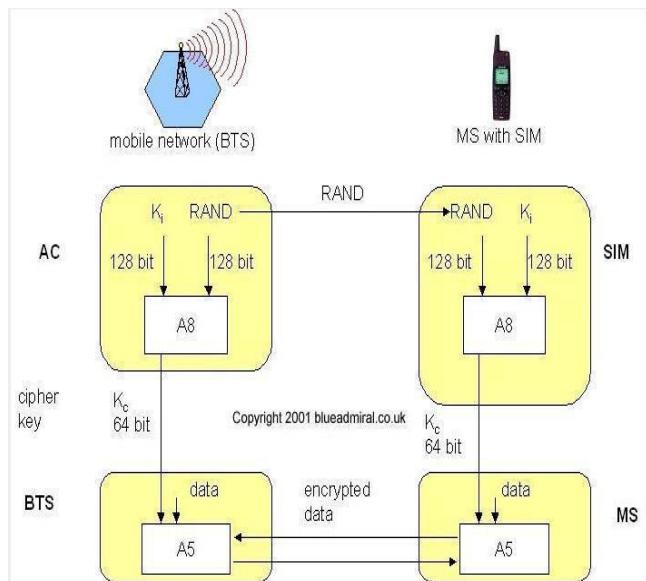
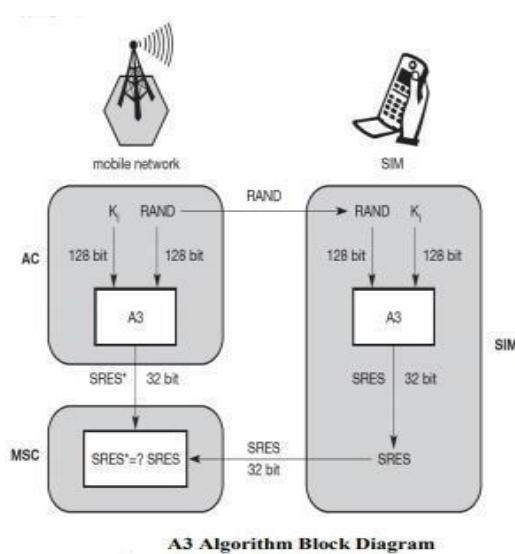
### Experiment No.: 5

**Aim:** Implementation of GSM security algorithms

**(A3/A5/A8) Theory:**

The security procedures in GSM are aimed at protecting the network against unauthorized access and protecting the privacy of mobile subscriber against eavesdropping, eavesdropping on subscriber communication is prevented by ciphering the information. To protect identity and location of the subscriber the appropriate signaling channels are ciphered and Temporary Subscriber Identity (TMSI) instead of IMSI is used over the radio path. At the time of initiating a service, the mobile terminal is powered on the subscriber may be required to enter 4-8 digits Password Identification Number (PIN) to validate the ownership of the SIM. At the time of service provisioning the IMSI, the individual subscriber authentication key (Ki), the authentication algorithm (A3), the cipher key generation algorithm (A8) and the encryption algorithm (A5) are programmed into the SIM by GSM operator. The A3 ciphering algorithm is used to authenticate each mobile by verifying the user password within the SIM with the cryptographic key at the MSC. The A5 ciphering algorithm is used for encryption. It provides scrambling for 114 coded bits sent in each TS. The A8 is used for ciphering key. The IMSI and the secret authentication key (Ki) are specific to each mobile station, the authentication algorithm A3 and A8 are different for different networks and operator's encryption algorithm A5 is unique and needs to be used across all GSM network operators. The authentication center is responsible for all security aspects and its function is closely linked with HLR. The secret authentication key (Ki) is not known to mobile user and is the property of service provider, the home system of the mobile station (MS) generates the random number say Rand which is 126-bit number. This random number is sent to MS. The MS uses A3 algorithm to authenticate the user. The algorithm A3 uses Ki and Rand number to generate a signed

result called s\_RES. MS sends s\_RES to home system of MS. In the home system authentication contains Ki and it also uses the same authentication algorithm A3 to authenticate the valid user. The A3 algorithm use Ki and Rand generated by home system to generate a signed result called  $\ll(s\rr)_RES$ . The s\_RES generated by MS and authentication center are compared. If both s\_RES are identical only then the user is valid and access is granted otherwise no



### Code:

```

import random

k=random.getrandbits(128)
m=random.getrandbits(128)
kb=bin(k)[2:]
mb=bin(m)[2:]
kbl=kb[0:64]
kbr=kb[64:]
mbl=mb[0:64]
mbr=mb[64:]
a1=int(kbl,2)^int(mbr,2)
a2=int(kbr,2)^int(mbl,2)
a3=a1^a2
a4=bin(a3)[2:].zfill(64)
a5=a4[0:32]
a6=a4[32:]"
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

a7=int(a5,2)^int(a6,2)



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

```
print("128 Bit Key = ",kb)
print("128 Random Bits Generated = ",mb)
print("RES/SRES = ",bin(a7)[2:].zfill(len(a5)))
```

<https://github.com/vinaynpp/mcc>

**Output:**

```
128 Bit Key =
1111101110100110010000010010011000100111001111010011101011010001111000111000001111
011101110110111010100010110101000111010001
```

```
128 Random Bits Generated =
1100000100010001011000101110010011011010110011001000110101110001001000010100101001
000001001111000000100001100100111111000100
```

```
RES/SRES = 111101101101000001011110001101
```

**Conclusion:** Thus, we have performed the experiment to implement the A3, A5 and A8 algorithm of the GSM security and we have implemented and executed it properly with proper output.



### Experiment No.:

#### **6 Aim:** Illustration of Hidden Terminal Problem (NS-2)

#### **Theory:**

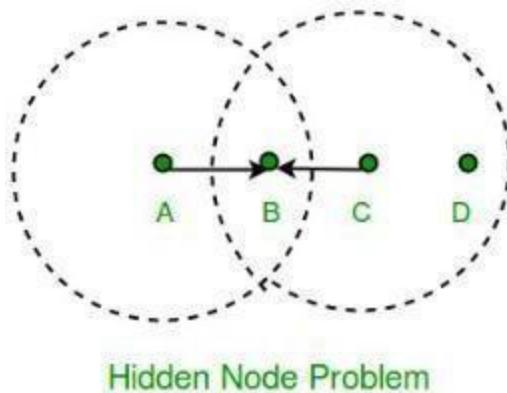
A wireless network with lack of centralized control entity, sharing of wireless bandwidth among network access nodes i.e. medium access control (MAC) nodes must be organized in decentralized manner. The hidden terminal problem occurs when a terminal is visible from a wireless access point (APs), but not from other nodes communicating with that AP. This situation leads the difficulties in medium access control sublayer over wireless networking.

In a formal way hidden terminal are nodes in a wireless network that are out of range of other node or a collection of nodes. Consider a wireless networking, each node at the far edge of the access point's range, which is known as A, can see the access point, but it is unlikely that the same node can see a node on the opposite end of the access point's range, C. These nodes are known as hidden. The problem is when nodes A and C start to send packets simultaneously to the access point B. Because the nodes A and C are out of range of each other and so cannot detect a collision while transmitting, Carrier sense multiple access with collision detection (CSMA/CD) does not work, and collisions occur, which then corrupt the data received by the access point. To overcome the hidden node problem, RTS/CTS handshaking (IEEE 802.11 RTS/CTS) is implemented in conjunction with the Carrier sense multiple accesses with collision avoidance (CSMA/CA) scheme. The same problem exists in a MANET.

The transmission range of access point A reaches at B, but not at access point C, similarly transmission range of access point C reaches B, but not at A. These nodes are known as hidden terminals. The problem occurs when nodes A and C start to send data packets simultaneously to the access point B. Because the access points A and C are out of range of each other and resultant they cannot detect a collision while transmitting, Carrier sense multiple access with collision detection (CSMA/CD) does not work, and collisions occur, which then corrupt the data received by the access point B due to the hidden terminal problem.

The hidden terminal analogy is described as follows:

- Terminal A sends data to B, terminal C cannot hear A
- Terminal C wants to send data to B, terminal C senses a “free” medium (CS fails) and starts transmitting
- Collision at B occurs, A cannot detect this collision (CD fails) and continues with its transmission to B
- Terminal A is “hidden” from C and vice versa.



Hidden Node Problem

The solution of hidden terminal problem is as follows.

When A wants to send a packet to B, A first sends a Request-to-send (RTS) to B.

On receiving RTS, B responds by sending Clear-to-Send (CTS).

When C overhears a CTS, it keeps quiet for the duration of the transfer. Transfer duration is included in both RTS and CTS.

RTS and CTS are short frames, reduces collision chance.

#### Code:

```
https://github.com/vinaynpp/mcc
BEGIN{
    sim_end = 200;
    i=0;
    while (i<=sim_end) {sec[i]=0; i+=1;};
}

{
    if ($1=="r" && $7=="cbr"&& $3=="_0_") {
        sec[int($2)]+=$8;
    };
}

END{
    i=0;
    while (i<=sim_end) {print i " " sec[i]; i+=1;};
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

}# Define options



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
set val(chan)      Channel/WirelessChannel    ;# channel type
set val(prop)      Propagation/FreeSpace     ;# radio-propagation model
set val(netif)     Phy/WirelessPhy          ;# network interface type
set val(mac)       Mac/802_11              ;# MAC type
set val(ifq)       Queue/DropTail/PriQueue   ;# interface queue type
set val(ll)        LL                      ;# link layer type
set val(ant)       Antenna/OmniAntenna      ;# antenna
model set val(ifqlen)           10000          ;#
max packet in ifq set val(nn)      5            ;# number of
mobilenodes
set val(rp)        DSR                      ;# routing protocol
set val(x)         600                      ;# X dimension of
topography set val(y)           600           ;# Y
dimension of topography set val(stop) 100          ;# time of
simulation end
set val(R)         300
set opt(tr)        out.tr
set ns      [new Simulator]
set tracefd [open $opt(tr) w]
set windowVsTime2 [open win.tr w]
set namtrace  [open simwrls.nam w]
Mac/802_11 set dataRate_ 1.2e6
Mac/802_11 set RTSThreshold_ 100
$ns trace-all $tracefd
#$ns use-newtrace
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo      [new Topography]

$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

#
# Create nn mobilenodes [$val(nn)] and attach them to the channel.
#
# configure the nodes

$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channelType $val(chan) \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
```



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

-movementTrace ON

Phy/WirelessPhy set CSThresh 30.5e-10



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
for {set i 0} {$i < $val(nn) } { incr i } {
    set node_($i) [$ns node]
}
$node_(0) set X_ $val(R)
$node_(0) set Y_ $val(R)
$node_(0) set Z_ 0
$node_(1) set X_ $val(R)
$node_(1) set Y_ 0
$node_(1) set Z_ 0
$node_(2) set X_ 0
$node_(2) set Y_ $val(R)
$node_(2) set Z_ 0
$node_(3) set X_ [expr $val(R) *2]
$node_(3) set Y_ $val(R)
$node_(3) set Z_ 0
$node_(4) set X_ $val(R)
$node_(4) set Y_ [expr $val(R) *2]
$node_(4) set Z_ 0
for {set i 0} {$i<$val(nn)} {incr i} {
    $ns initial_node_pos $node_($i) 30
}
# Generation of movements
$ns at 0 "$node_(1) setdest $val(R) $val(R) 3.0"
$ns at 0 "$node_(2) setdest $val(R) $val(R) 3.0"
$ns at 0 "$node_(3) setdest $val(R) $val(R) 3.0"
$ns at 0 "$node_(4) setdest $val(R) $val(R) 3.0"
# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
#$tcp set class_ 2
set tcp [new Agent/UDP]
$tcp set class_ 2
set sink [new Agent/Null]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at 0.0 "$ftp start"
#####
# For coloring but doesnot work
#####
$tcp set fid_ 1
$ns color 1 blue
#####
set tcp [new Agent/UDP]
$tcp set class_ 2
set sink [new Agent/Null]
$ns attach-agent $node_(2) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at 0.0 "$ftp start"
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

set tcp [new Agent/UDP]



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
$tcp set class_2
set sink [new Agent/Null]
$ns attach-agent $node_(3) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at 0.0 "$ftp start"
set tcp [new Agent/UDP]
$tcp set class_2
set sink [new Agent/Null]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at 0.0 "$ftp start"
# Telling nodes when the simulation ends
#for {set i 0} {$i < $val(nn)} { incr i } {
# $ns at $val(stop) "$node_($i) reset";
#}

# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at $val(stop) "puts \"end simulation\" ; $ns halt"
proc stop {} {
exec awk -f fil.awk out.tr > out.xgr
exec xgraph out.xgr &

global ns tracefd namtrace
$ns flush-trace
close $tracefd
close $namtrace
exec nam simwrls.nam &
}

$ns run
```

**Output:**

- 1) The node 0 and 2 want to send data to node 1 the range of node 0 and 2 is limited to 1 they do not know that other node is also sending data to 1 and therefore collision occurs.



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

The screenshot shows a Linux desktop environment with a dark theme. A terminal window is open, titled "Terminal". The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The status bar at the top right shows the date and time as "Fri 06:15" and the user information as "vyd@vyd-VirtualBox: ~/Downloads". The terminal window displays the following command and its output:

```
vyd@vyd-VirtualBox:~/Downloads$ ns vinay.tcl
num_nodes is set 5
warning: Please use -channel as shown in tcl/ex/wireless-mitf.tcl
INITIALIZE THE LIST xListHead
```



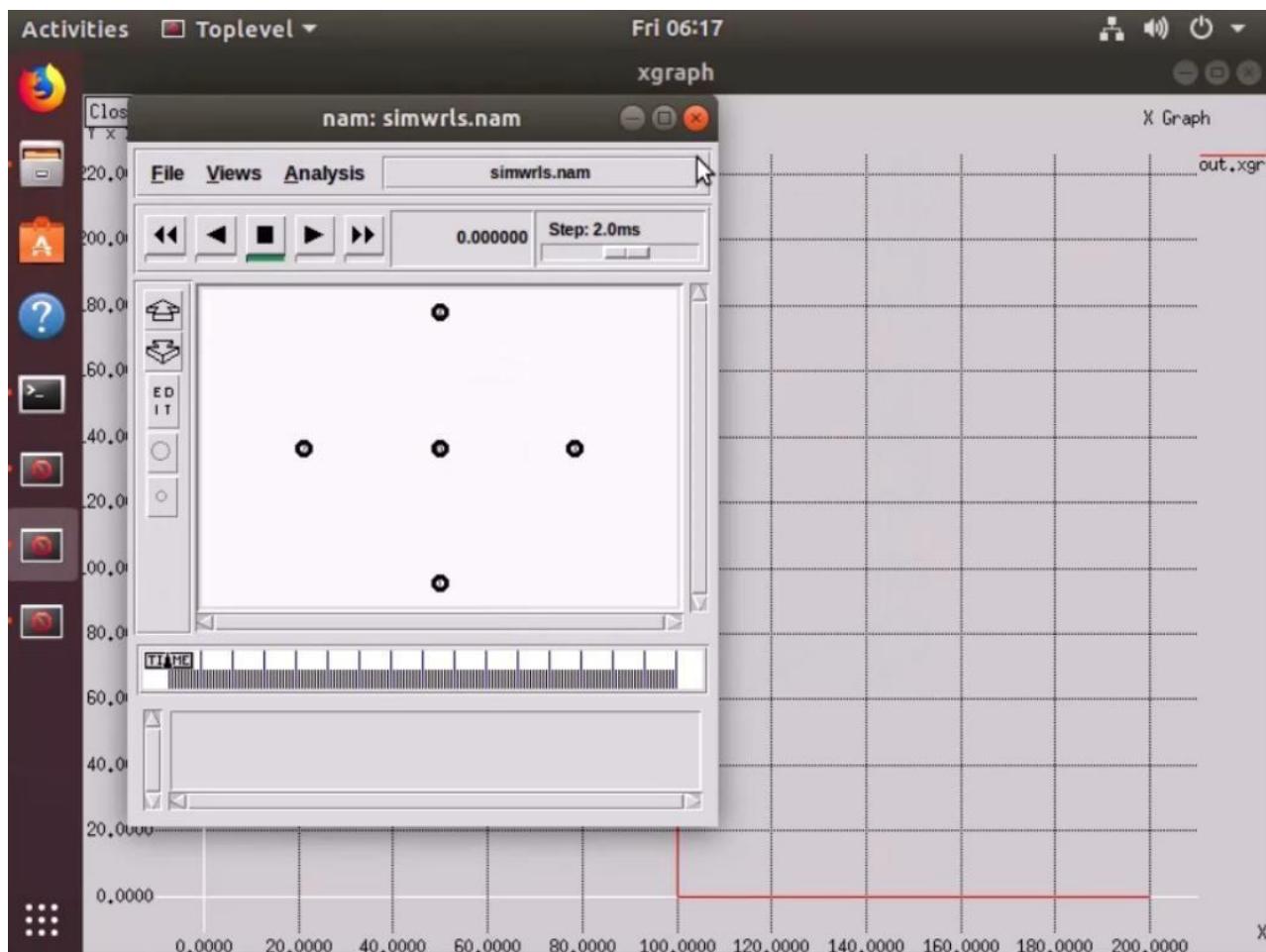
Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

Activities Terminal Fri 06:17  
vyd@vyd-VirtualBox: ~/Downloads

```
File Edit View Search Terminal Help
98.543203: d: 6.180665, Pr: 5.033381e-06
98.543203: d: 6.180665, Pr: 5.033381e-06
98.543203: d: 8.740780, Pr: 2.516691e-06
98.543565: d: 4.369304, Pr: 1.007177e-05
98.543879: d: 4.368362, Pr: 1.007611e-05
98.543879: d: 6.177797, Pr: 5.038056e-06
98.543879: d: 6.177797, Pr: 5.038056e-06
98.543879: d: 8.736724, Pr: 2.519028e-06
98.545841: d: 4.362476, Pr: 1.010332e-05
98.546275: d: 4.361174, Pr: 1.010935e-05
98.546275: d: 6.167631, Pr: 5.054677e-06
98.546275: d: 6.167631, Pr: 5.054677e-06
98.546275: d: 8.722347, Pr: 2.527338e-06
98.546637: d: 4.360088, Pr: 1.011439e-05
98.546951: d: 4.359146, Pr: 1.011876e-05
98.546951: d: 6.164763, Pr: 5.059381e-06
98.546951: d: 6.164763, Pr: 5.059381e-06
98.546951: d: 8.718291, Pr: 2.529691e-06
98.548913: d: 4.353260, Pr: 1.014614e-05
```

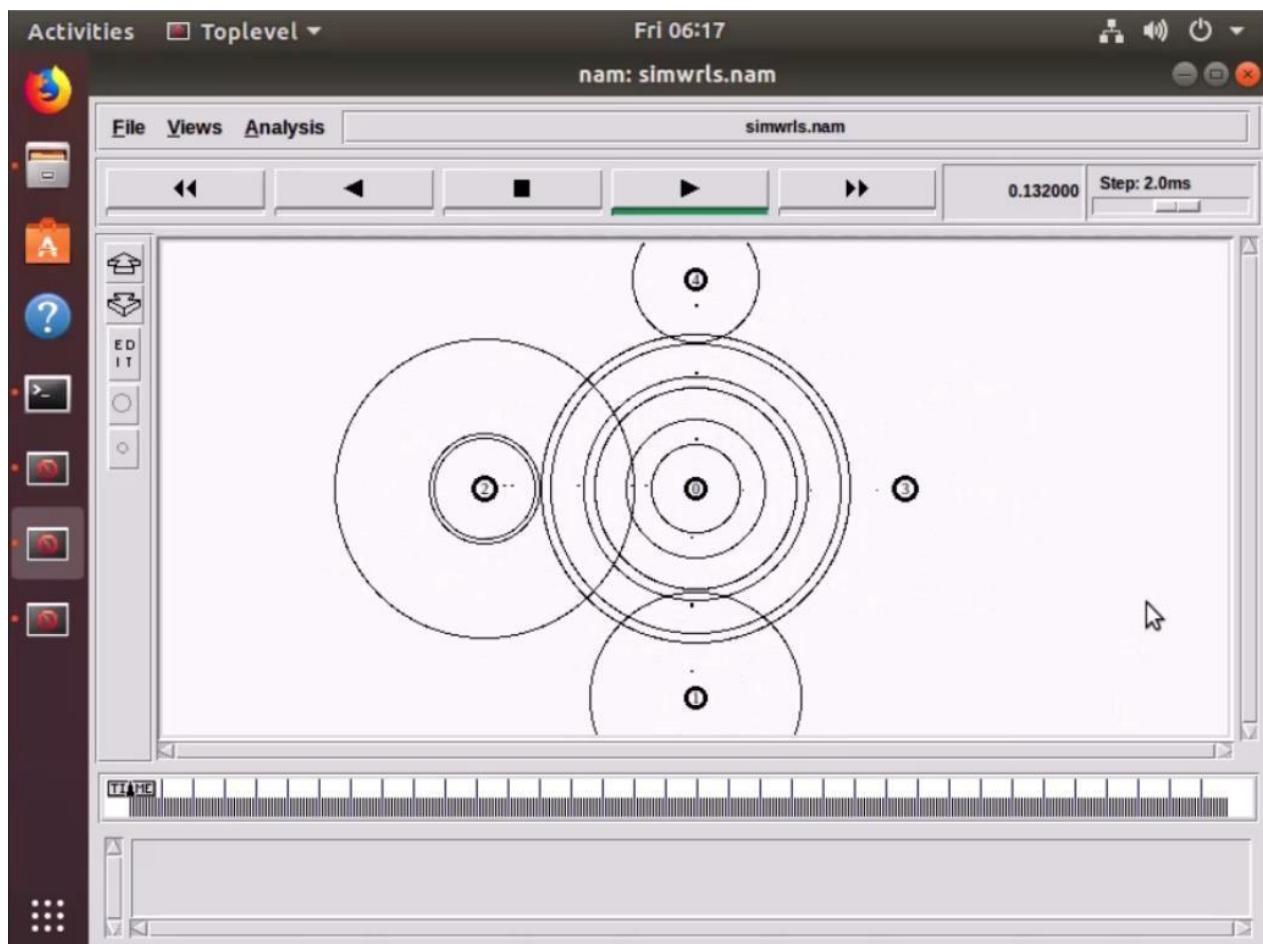


Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC



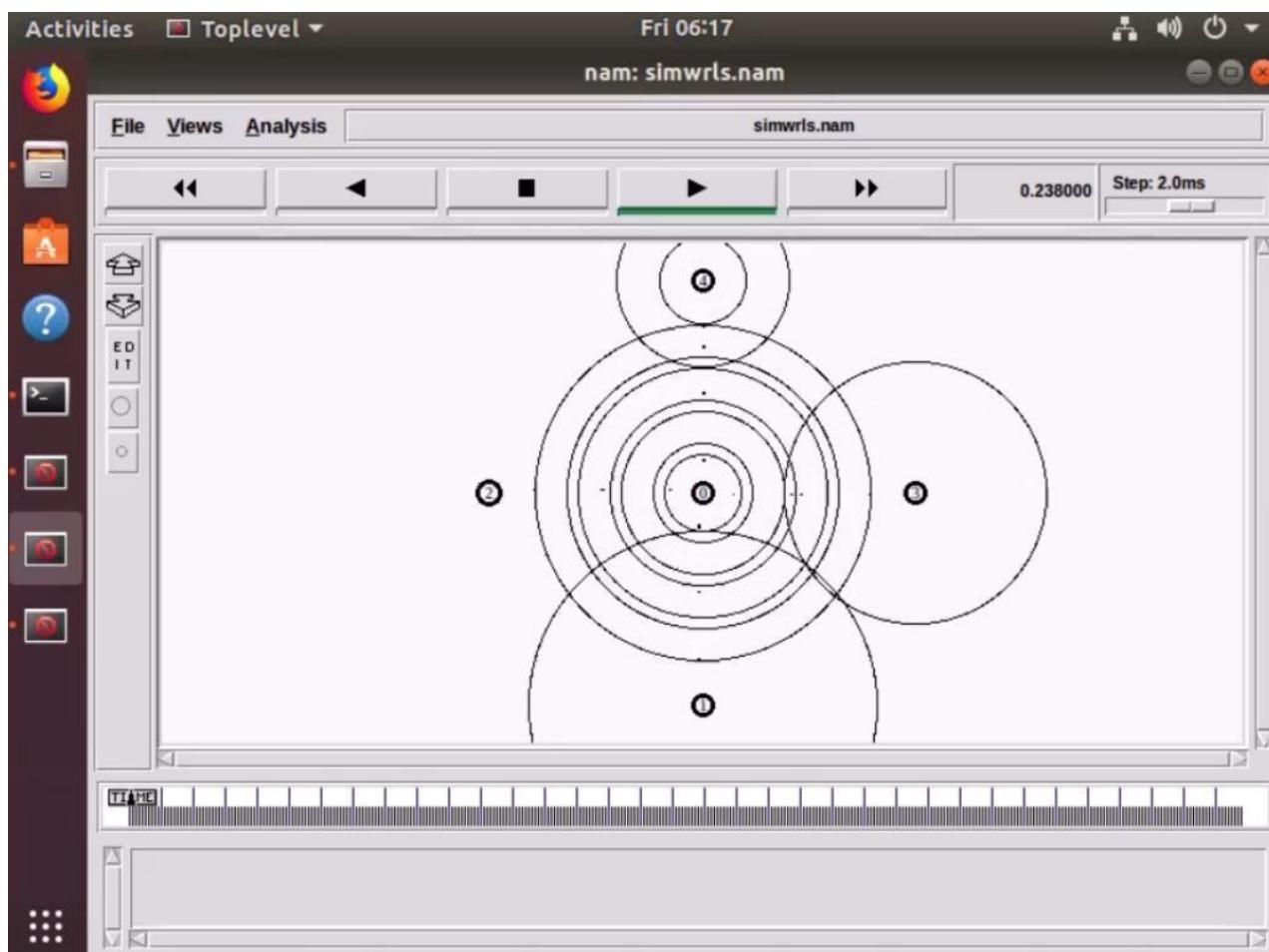


Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC



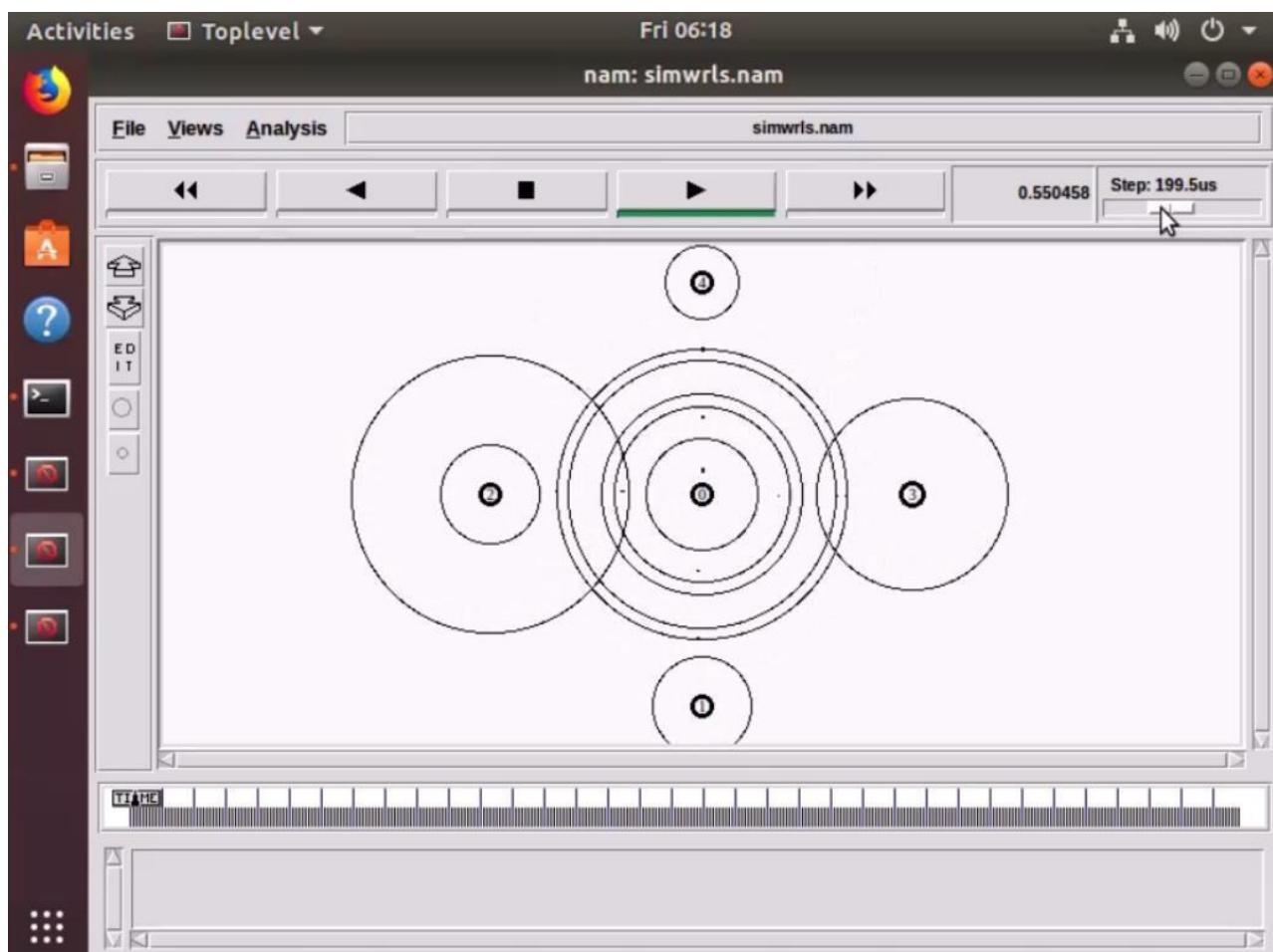


Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC





Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC





<https://github.com/vinaynpp/mcc>

### Conclusion:

Thus, we have performed the experiment of and illustrated the hidden terminal problem using NS2 and properly explained the same which helps to understand better.



### Experiment No.

**7 Aim:** Develop an application that uses GUI

components. **Theory:**

A typical user interface of an android application consists of action bar and the application content area.

- Main Action Bar
- View Control
- Content Area
- Split Action Bar

The basic unit of android application is the activity. A UI is defined in an xml file. During compilation, each element in the XML is compiled into equivalent Android GUI class with attributes represented by methods.

#### View and ViewGroups

An activity is consist of views. A view is just a widget that appears on the screen. It could be button etc. One or more views can be grouped together into one GroupView. Example of ViewGroup includes layouts.

#### Types of layout

There are many types of layout. Some of which are listed below –

- Linear Layout
- Absolute Layout
- Table Layout
- Frame Layout
- Relative Layout

The basic building block for user interface is a **View** object which is created from the View class and occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components like buttons, text fields, etc.

The **ViewGroup** is a subclass of **View** and provides invisible container that hold other Views or other ViewGroups and define their layout properties.



At third level we have different layouts which are subclasses of ViewGroup class and a typical layout defines the visual structure for an Android user interface and can be created either at run time using **View/ViewGroup** objects or you can declare your layout using simple XML file **main\_layout.xml** which is located in the res/layout folder of your project.

**Code:**

```
object Components {

    const val ratingBar = ":libraries:rating-bar"
    const val dialogs = ":libraries:dialogs"
    const val imageSlider = ":libraries:image-slider"
    const val phoneNumber = ":libraries:phonenumbers"
    const val toolbar = ":libraries:toolbar"
    const val suggestionInputView = ":libraries:suggestion-input-view"
    const val cardInputView = ":libraries:card-input-view"
    const val quantityPickerView = ":libraries:quantity-picker-view"
    const val timelineView = ":libraries:timeline-view"
    const val touchDelegator = ":libraries:touch-delegate"
    const val fitOptionMessageView = ":libraries:fit-option-message-view"

}

https://github.com/vinaynpp/mcc
object ComponentVersions {

    const val toolbarVersion = "2.0.5"
    const val suggestionInputViewVersion = "1.0.14"
    const val ratingBarVersion = "1.0.2"
    const val imageSliderVersion = "1.0.8"
    const val phoneNumberVersion = "1.0.2"
    const val dialogsVersion = "1.2.5"
    const val cardInputViewVersion = "1.1.2"
    const val quantityPickerViewVersion = "1.2.4"
    const val timelineViewVersion = "1.0.0"
    const val touchDelegatorVersion = "1.0.0"
    const val fitOptionMessageView = "1.0.0"
}

object Configs {

    const val compileSdkVersion = 29
    const val minSdkVersion = 21
    const val targetSdkVersion = 29
    const val buildToolsVersion = "29.0.3"

    const val applicationId = "com.trendyol.uicomponents"
    const val group = "com.trendyol.ui-components"
}
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

object Dependencies {



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
const val kotlinJDK = "org.jetbrains.kotlin:kotlin-stdlib-jdk8:1.3.61"

const val coreKtx = "androidx.core:core-ktx:1.3.2"
const val appCompat = "androidx.appcompat:appcompat:1.2.0"
const val material = "com.google.android.material:material:1.4.0"
const val constraintLayout = "androidx.constraintlayout:constraintlayout:2.0.4"
const val recyclerView = "androidx.recyclerview:recyclerview:1.2.0"
const val circleIndicator = "com.github.MertNYuksel:CircleIndicator:2a2e973374"
const val glide = "com.github.bumptech.glide:glide:4.11.0"
const val glideCompiler = "com.github.bumptech.glide:compiler:4.11.0"
const val lifecycleExtensions = "androidx.lifecycle:lifecycle-extensions:2.1.0"
const val lifecycleCompiler = "androidx.lifecycle:lifecycle-compiler:2.1.0"
}

object Plugins {

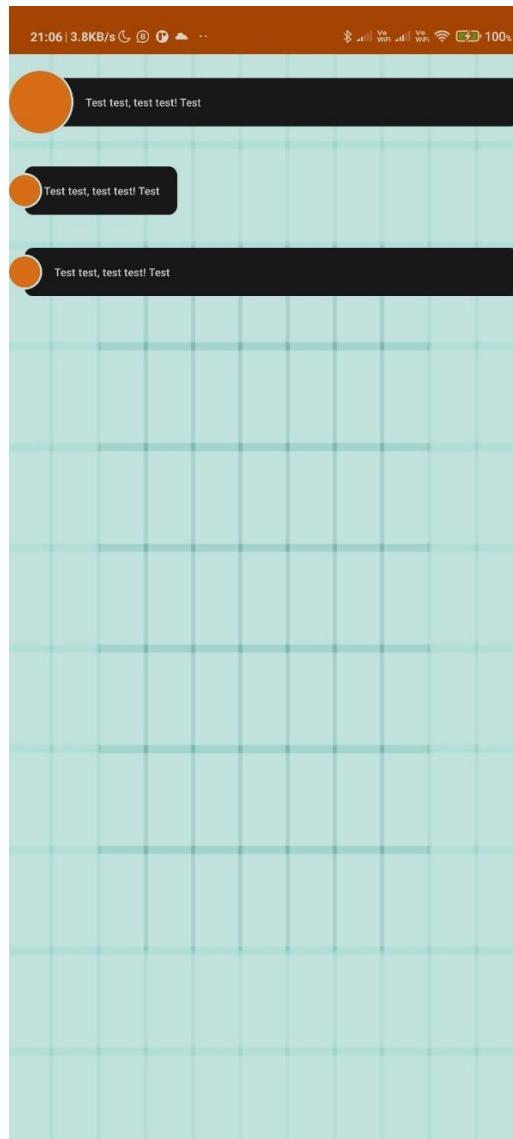
    const val androidGradlePlugin = "com.android.tools.build:gradle:4.2.1"
    const val kotlinGradlePlugin = "org.jetbrains.kotlin:kotlin-gradle-plugin:1.5.0"
    const val mavenGradlePlugin = "com.github.dcendents:android-maven-gradle-plugin:2.1"

    const val androidApplication = "com.android.application"
    const val androidLibrary = "com.android.library"
    const val kotlinAndroid = "kotlin-android"
    const val kotlinKapt = "kotlin-kapt"
    const val kotlinParcelize = "kotlin-parcelize"
    const val androidMaven = "com.github.dcendents.android-maven"
}
```



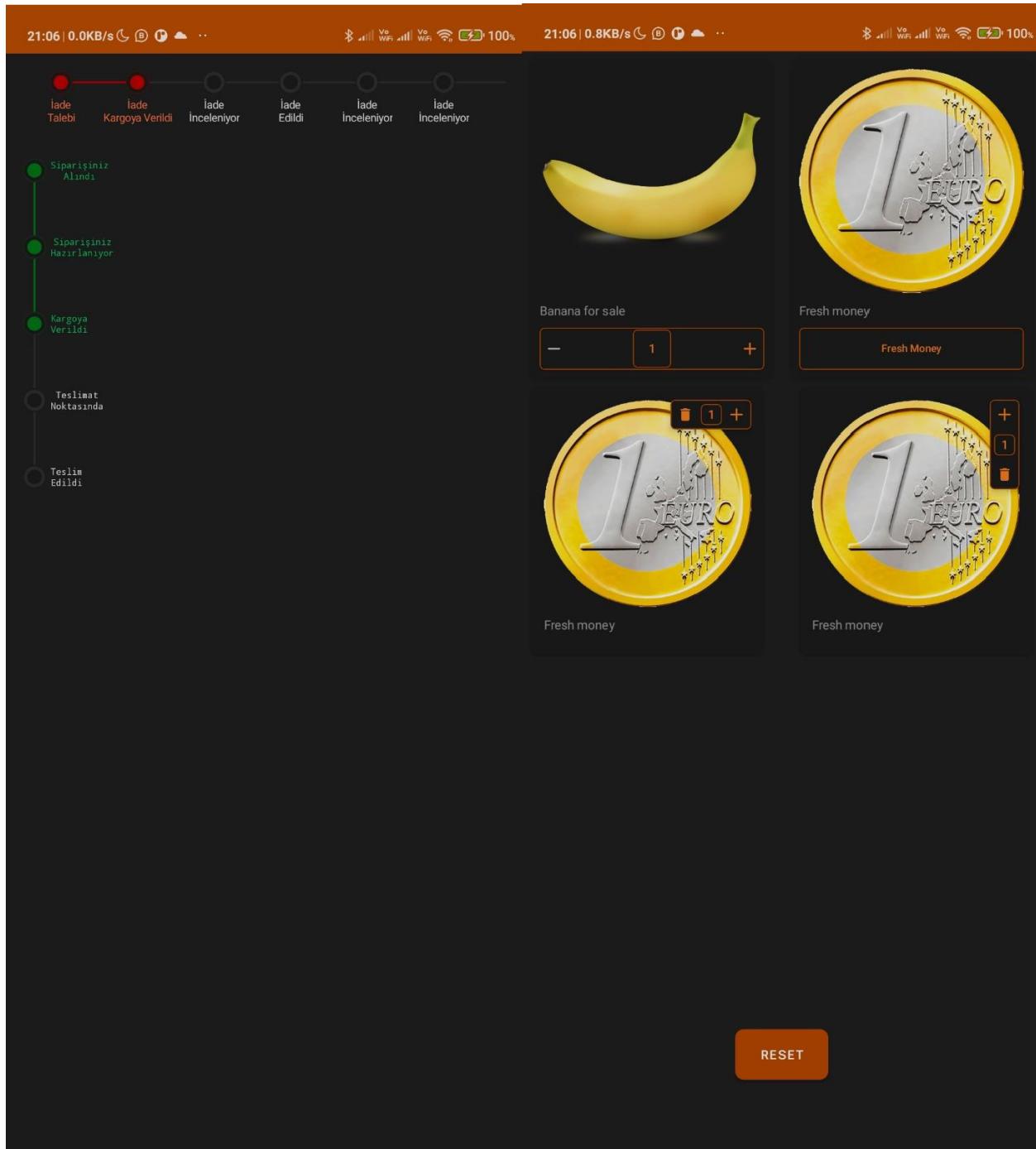
**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

**Output:**





Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC





Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

Card Number

21:06 | 0.0KB/s

MM YY

Card Number:

Cvv:

VALIDATE

VALIDATE AND GET CARD INFORMATION

RESET

21:06 | 4.0KB/s

10 6 20 6 30 6 Other

LOAD

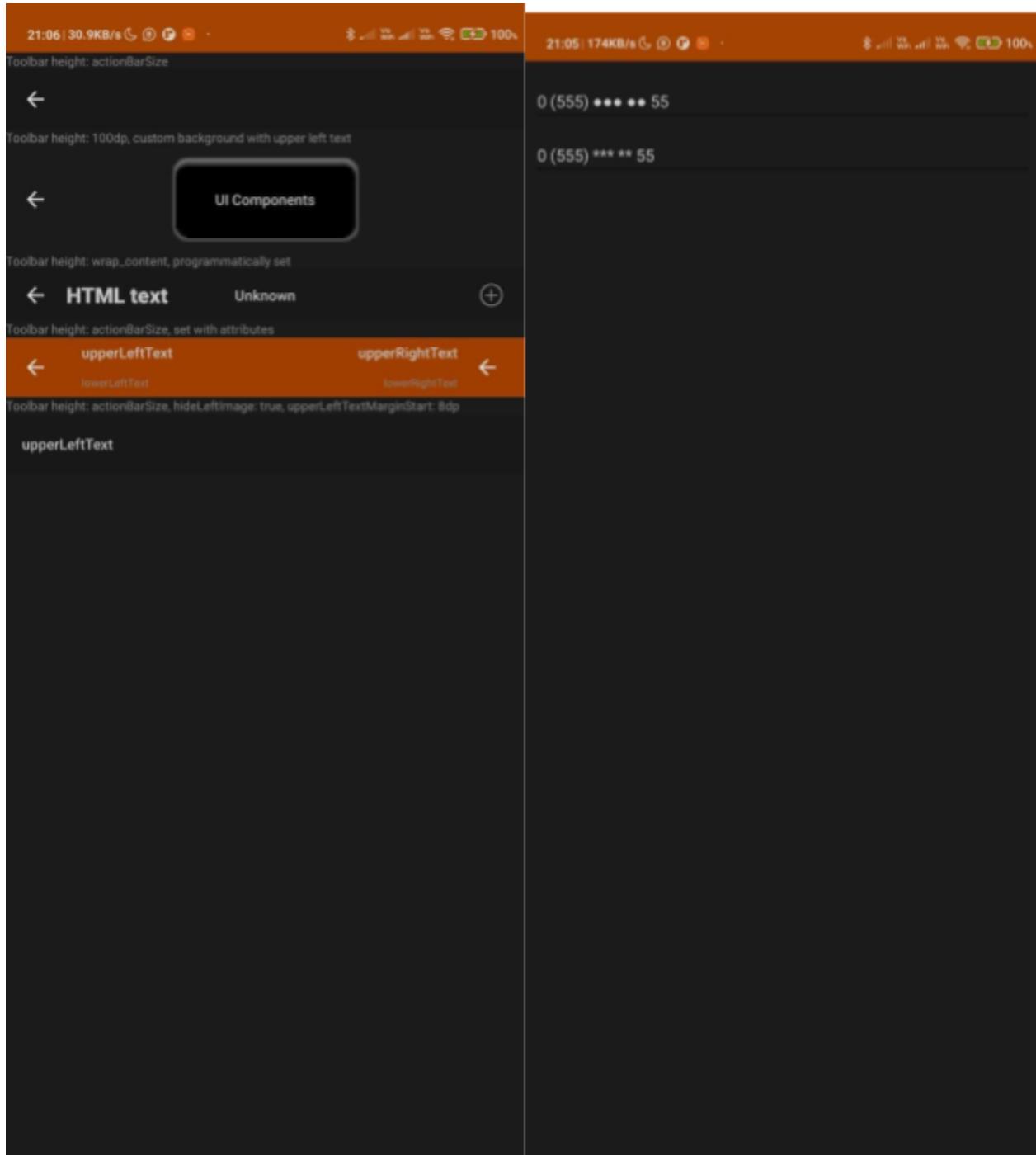
Selected Text:  
Selected Value:

Scan a new card

1	2	3	-
4	5	6	-
7	8	9	⌫
,	0	.	→



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC





Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

21:05 | 49.1KB/s | Select dialog | SHOW INFO DIALOG | SHOW AGREEMENT DIALOG | SHOW SELECTION DIALOG | SHOW SELECTION WITH SEARCH DIALOG | SHOW INFO DIALOG WITH WEBVIEW

Agreement Dialog Sample

Enter the main heading, usually the same as the title.

Be bold in stating your key points. Put them in a list:

- The first item in your list
- The second item; italicize key words

Visit Trendyol!

[Cancel](#) [Agree](#)



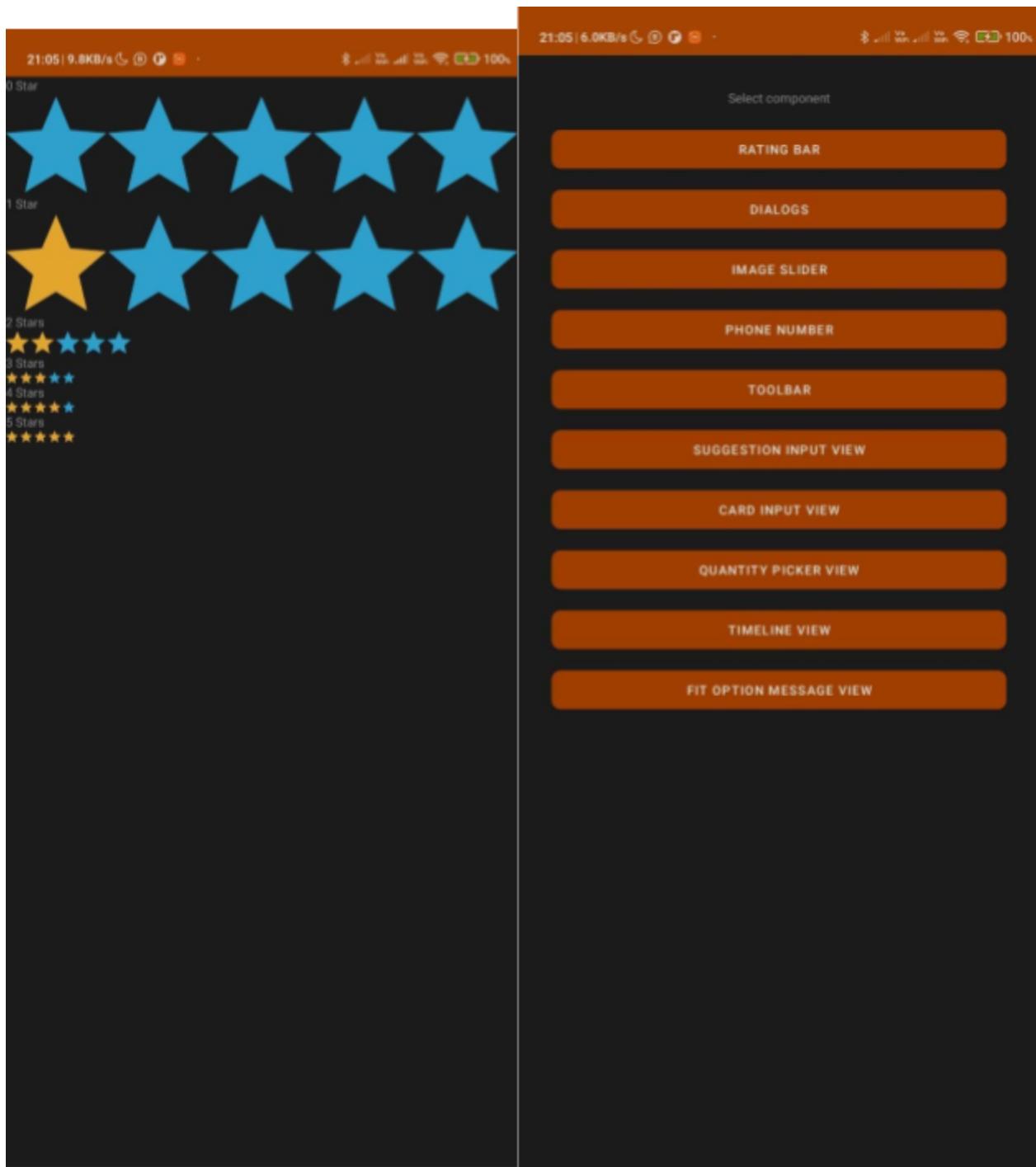
**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

The screenshot displays a mobile application interface with several UI elements:

- Top Status Bar:** Shows battery level at 100%, signal strength, and connectivity icons.
- Header:** A dark header bar with the text "Select dialog".
- Buttons:** Two orange rounded rectangular buttons labeled "SHOW INFO DIALOG".
- Section:** A dark box labeled "Selection Dialog Sample" containing:
  - A small envelope icon.
  - The text "Enter the main heading, usually the same as the title."
  - The text "Be bold in stating your key points. Put them in a list:"
  - A bulleted list:
    - The first item in your list
    - The second item; *italicize* key words
  - The text "Visit Trendyol!"
  - A list of 9 options, each represented by a radio button and a label:
    - 1th option
    - 2th option
    - 3th option
    - 4th option
    - 5th option
    - 6th option (selected, indicated by a checked radio button)
    - 7th option
    - 8th option
    - 9th option
- Buttons:** Four orange rounded rectangular buttons labeled "SHOW AGREEMENT DIALOG", "SHOW SELECTION DIALOG", "SHOW SELECTION WITH SEARCH DIALOG", and "SHOW INFO DIALOG WITH WEBVIEW".
- Bottom Dialog:** A semi-transparent dark dialog box titled "Info Dialog Sample" containing:
  - An Android icon.
  - A checkmark icon.
  - Text: "Lorem ipsum dolor sit amet, **consectetur adipiscing elit**, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. **Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.** [trendyol@trendyol.com](mailto:trendyol@trendyol.com) Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est [laborum.trendyol.com](http://laborum.trendyol.com). Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. +905066979797"



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC



**Conclusion:**

Thus, we have performed the experiment to use the GUI components in the android studio and made an app which shows the color change the text and font type used also different size of the



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

text, background color and successfully executed it.



### Experiment No.: 8

**Aim:** To make an application that draws basic graphical primitives on the screen.

**Theory:**

The android.graphics.Canvas can be used to draw graphics in android. It provides methods to draw oval, rectangle, picture, text, line etc.

The android.graphics.Paint class is used with canvas to draw objects. It holds the information of color and style.

Android Canvas class encapsulates the bitmaps used as surface. It exposes the draw methods which can be used for designing. Let us first clear the following terms:

**Bitmap:** The surface being drawn on.

**Paint:** It lets us specify how to draw the primitives on bitmap. It is also referred to as “Brush”.

**Canvas:** It supplies the draw methods used to draw primitives on underlying bitmap.

Each drawing object specifies a paint object to render. Let us see the available list of drawing objects and they are as follows:

**drawArc:** This draws an arc between the two angles bounded by an area of rectangle.

**drawBitmap:** It draws an bitmap on canvas.

**drawRGB/drawARGB/drawColor:** This fills the canvas with a single color.

**drawBitmapMesh:** It draws a bitmap using a mesh. It manipulates the appearance of target by moving points on it.

**drawCircle:** This draws a circle on a specified radius centered on a given point.

**drawLine(s):** It draws a line (or series of lines) between points.

**drawOval:** It draws an oval which is bounded by the area of rectangle. **drawPaint:** It fills the entire canvas with a specific paint.

**drawPath:** It draws a path as per specification.

**drawPicture:** It draws a picture specified on a rectangular area.

**drawPosText:** It draws a text string specifying the offset of each character.

**drawRect:** It draws a rectangle.

**drawRoundRect:** It draws a rectangle with round edges.

**drawText:** It draws a text string on canvas.

The **Paint** class consists of a paint brush and a palette. It lets us choose how to render the primitives drawn into canvas by draw methods. We can control the color, style, font, special effects etc can be modified by modifying the paint object. For instance, **setColor** method can be used to select the color of Paint. Paint class supports transparency so it can be used to control variety of shades or effects, etc. Let us create a simple example and see the basic usage of canvas



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

and paint.



**Code:**

```
package co.martinbaciga.drawingtest.ui.component;
https://github.com/vinaynpp/mcc
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.Path;
import android.graphics.PorterDuff;
import android.util.AttributeSet;
import android.view.MotionEvent;
import android.view.View;

import java.util.ArrayList;

public class DrawingView extends View
{
    private Path mDrawPath;
    private Paint mBackgroundPaint;
    private Paint mDrawPaint;
    private Canvas mDrawCanvas;
    private Bitmap mCanvasBitmap;

    private ArrayList<Path> mPaths = new ArrayList<>();
    private ArrayList<Paint> mPaints = new ArrayList<>();
    private ArrayList<Path> mUndonePaths = new ArrayList<>();
    private ArrayList<Paint> mUndonePaints = new ArrayList<>();

    // Set default values
    private int mBackgroundColor = 0xFFFFFFFF;
    private int mPaintColor = 0xFF660000;
    private int mStrokeWidth = 10;

    public DrawingView(Context context, AttributeSet attrs)
    {
        super(context, attrs);
        init();
    }

    private void init()
    {
        mDrawPath = new Path();
        mBackgroundPaint = new Paint();
        initPaint();
    }
}
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

private void initPaint()



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
{  
    mDrawPaint = new Paint();  
    mDrawPaint.setColor(mPaintColor);  
    mDrawPaint.setAntiAlias(true);  
    mDrawPaint.setStrokeWidth(mStrokeWidth);  
    mDrawPaint.setStyle(Paint.Style.STROKE);  
    mDrawPaint.setStrokeJoin(Paint.Join.ROUND);  
    mDrawPaint.setStrokeCap(Paint.Cap.ROUND);  
}  
  
private void drawBackground(Canvas canvas)  
{  
    mBackgroundPaint.setColor(mBackgroundColor);  
    mBackgroundPaint.setStyle(Paint.Style.FILL);  
    canvas.drawRect(0, 0, this.getWidth(), this.getHeight(), mBackgroundPaint);  
}  
  
private void drawPaths(Canvas canvas)  
{  
    int i = 0;  
    for (Path p : mPaths)  
    {  
        canvas.drawPath(p, mPaints.get(i));  
        i++;  
    }  
}  
  
@Override  
protected void onDraw(Canvas canvas)  
{  
    drawBackground(canvas);  
    drawPaths(canvas);  
  
    canvas.drawPath(mDrawPath, mDrawPaint);  
}  
  
@Override  
protected void onSizeChanged(int w, int h, int oldw, int oldh)  
{  
    super.onSizeChanged(w, h, oldw, oldh);  
  
    mCanvasBitmap = Bitmap.createBitmap(w, h, Bitmap.Config.ARGB_8888);  
  
    mDrawCanvas = new Canvas(mCanvasBitmap);  
}  
  
@Override  
public boolean onTouchEvent(MotionEvent event)  
{  
    float touchX = event.getX();  
    float touchY = event.getY();  
  
    switch (event.getAction())
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

{



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
        case MotionEvent.ACTION_DOWN:
            mDrawPath.moveTo(touchX, touchY);
            //mDrawPath.addCircle(touchX, touchY, mStrokeWidth/10, Path.Direction.CW);
            break;
        case MotionEvent.ACTION_MOVE:
            mDrawPath.lineTo(touchX, touchY);
            break;
        case MotionEvent.ACTION_UP:
            mDrawPath.lineTo(touchX, touchY);
            mPaths.add(mDrawPath);
            mPaints.add(mDrawPaint);
            mDrawPath = new Path();
            initPaint();
            break;
        default: return false;
    }

    invalidate();
    return true;
}

public void clearCanvas()
{
    mPaths.clear();
    mPaints.clear();
    mUndonePaths.clear();
    mUndonePaints.clear();
    mDrawCanvas.drawColor(0, PorterDuff.Mode.CLEAR);
    invalidate();
}

public void setPaintColor(int color)
{
    mPaintColor = color;
    mDrawPaint.setColor(mPaintColor);
}

public void setPaintStrokeWidth(int strokeWidth)
{
    mStrokeWidth = strokeWidth;
    mDrawPaint.setStrokeWidth(mStrokeWidth);
}

public void setBackgroundColor(int color)
{
    mBackgroundColor = color;
    mBackgroundPaint.setColor(mBackgroundColor);
    invalidate();
}
```



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

```
public Bitmap getBitmap()
{
    drawBackground(mDrawCanvas);
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
        drawPaths(mDrawCanvas)
        ; return mCanvasBitmap;
    }

    public void undo()
    {
        if (mPaths.size() > 0)
        {
            mUndonePaths.add(mPaths.remove(mPaths.size() - 1));
            mUndonePaints.add(mPaints.remove(mPaints.size() - 1));
            invalidate();
        }
    }

    public void redo()
    {
        if (mUndonePaths.size() > 0)
        {
            mPaths.add(mUndonePaths.remove(mUndonePaths.size() - 1));
            mPaints.add(mUndonePaints.remove(mUndonePaints.size() - 1));
            invalidate();
        }
    }
}

package co.martinbaciga.drawingtest.domain.manager;

import android.Manifest;
import android.app.Activity;
import android.content.pm.PackageManager;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;

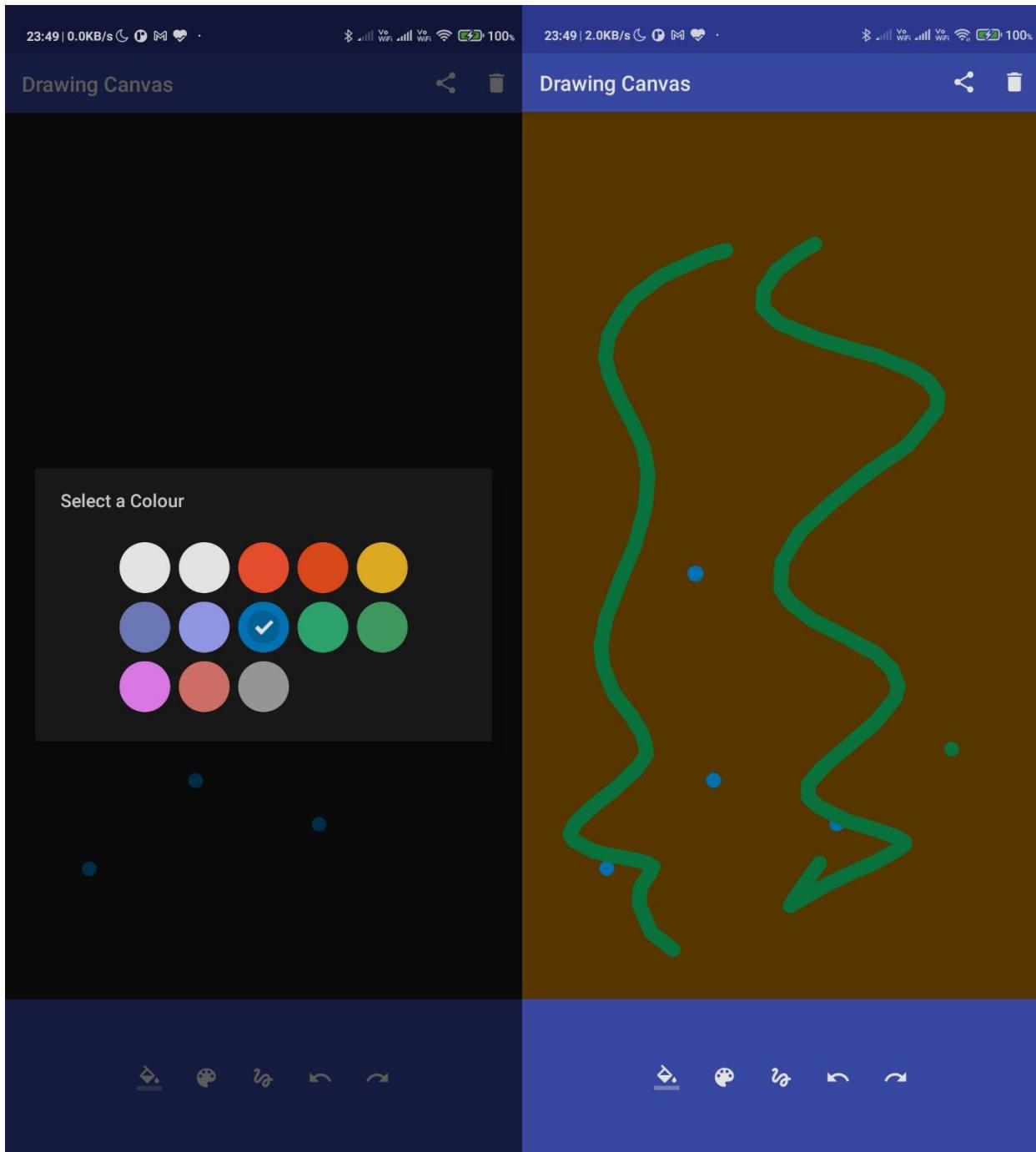
public class PermissionManager
{
    public static final int REQUEST_WRITE_STORAGE = 112;

    public static boolean checkWriteStoragePermissions(Activity activity)
    {
        boolean hasPermission = (ContextCompat.checkSelfPermission(activity,
Manifest.permission.WRITE_EXTERNAL_STORAGE) == PackageManager.PERMISSION_GRANTED);
        if (!hasPermission) {
            ActivityCompat.requestPermissions(activity,
                new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
                REQUEST_WRITE_STORAGE);
        }
        return hasPermission;
    }
}
```



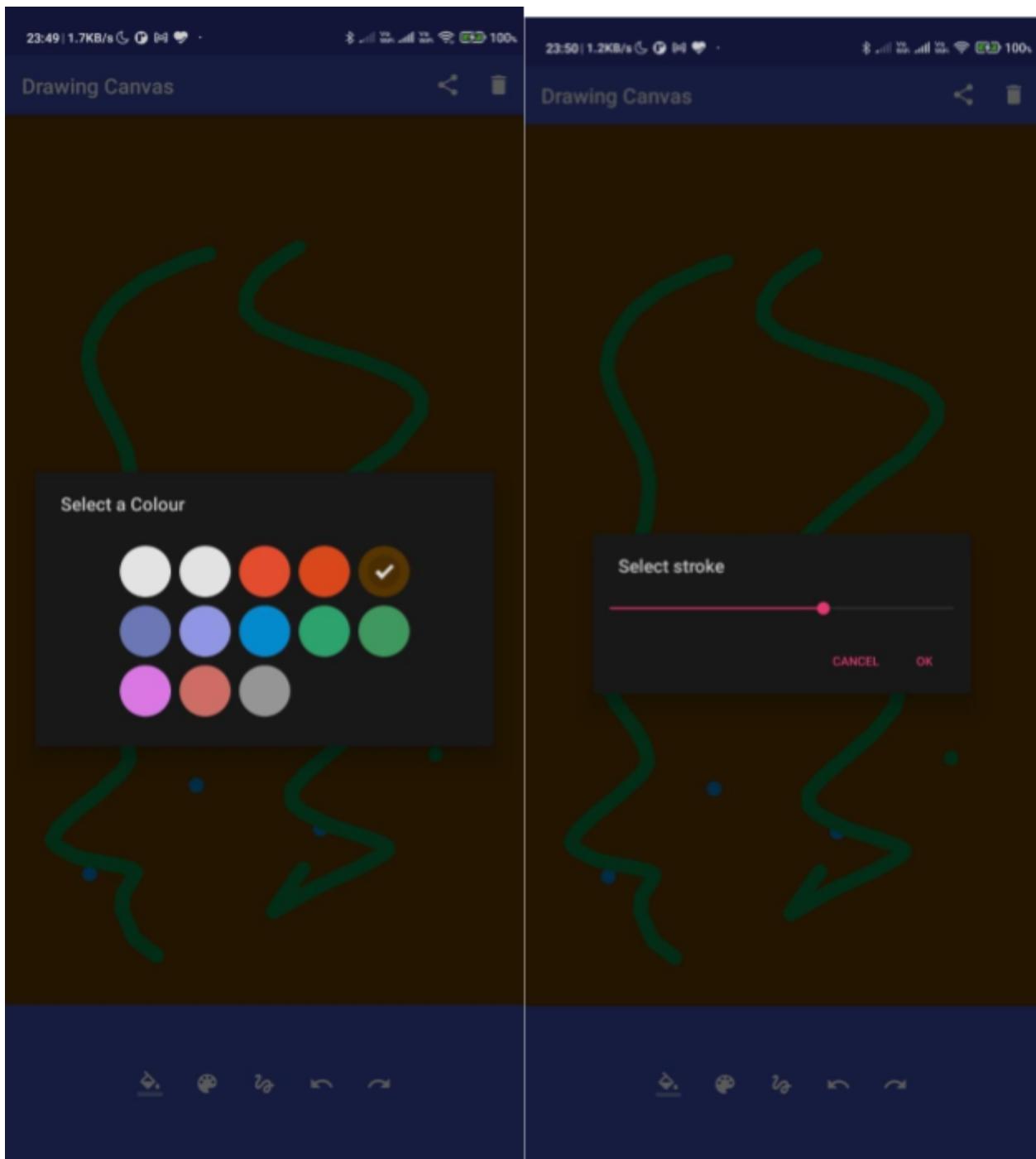
Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

### Output:



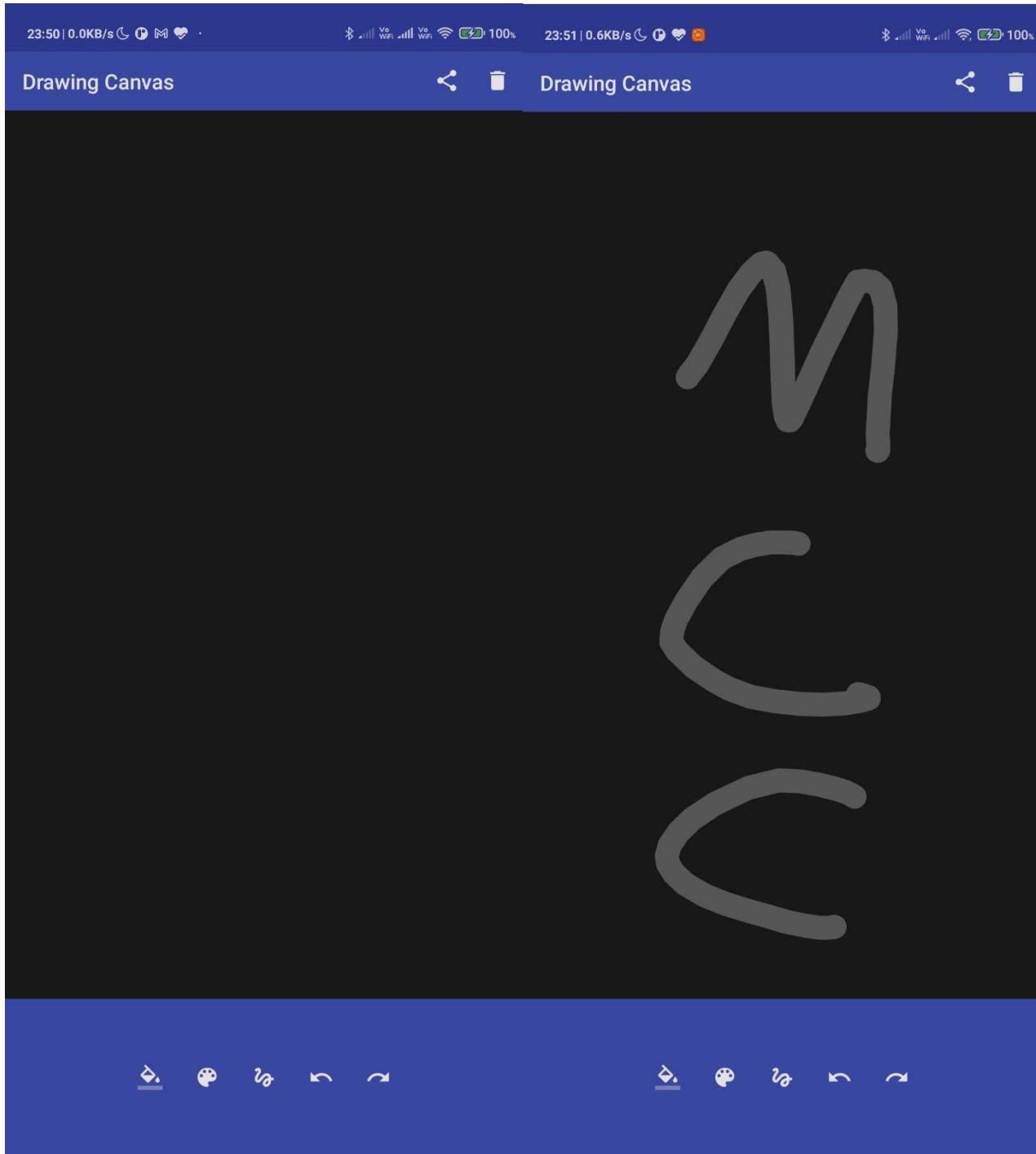


Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC





Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC



### Conclusion:

Thus, we have performed the experiment to draw basic graphical primitives on the screen in the android app using canvas in android studio and over here we have drawn circle and rectangle as



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

example and successfully executed it.



### Experiment No.:

**9 Aim:** Develop an application that makes use of database

#### **Theory:**

SQLite is a opensource SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation.

SQLite supports all the relational database features. In order to access this database, you don't need to establish any kind of connections for it like JDBC,ODBC e.t.c

#### **Database - Package**

The main package is android.database.sqlite that contains the classes to manage your own databases

#### **Database - Creation**

In order to create a database you just need to call this method openOrCreateDatabase with your database name and mode as a parameter.

#### **Database - Insertion**

we can create table or insert data into table using execSQL method defined in SQLiteDatabase class.

#### **Database - Fetching**

We can retrieve anything from database using an object of the Cursor class. We will call a method of this class called rawQuery and it will return a resultset with the cursor pointing to the table. We can move the cursor forward and retrieve the data.

#### **Database - Helper class**

For managing all the operations related to the database , an helper class has been given and is called SQLiteOpenHelper. It automatically manages the creation and update of the database.

#### **Code:**

```
package com.example.inventory
import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.navigation.NavController
import androidx.navigation.fragment.NavHostFragment
import androidx.navigation.ui.NavigationUI.setupActionBarWithNavController
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
class MainActivity : AppCompatActivity(R.layout.activity_main) {

    private lateinit var navController: NavController

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        // Retrieve NavController from the NavHostFragment
        val navHostFragment = supportFragmentManager
            .findFragmentById(R.id.nav_host_fragment) as NavHostFragment
        navController = navHostFragment.navController
        // Set up the action bar for use with the NavController
        setupActionBarWithNavController(this, navController)
    }

    /**
     * Handle navigation when the user chooses Up from the action bar.
     */
    override fun onSupportNavigateUp(): Boolean {
        return navController.navigateUp() || super.onSupportNavigateUp()
    }
}
```

<https://github.com/vinaynpp/mcc>

```
package com.example.inventory

import android.content.Context.INPUT_METHOD_SERVICE
import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.view.inputmethod.InputMethodManager
import android.widget.TextView
import androidx.fragment.app.Fragment
import androidx.fragment.app.activityViewModels
import androidx.navigation.fragment.findNavController
import androidx.navigation.fragment.navArgs
import com.example.inventory.data.Item
import com.example.inventory.databinding.FragmentAddItemBinding

/**
 * Fragment to add or update an item in the Inventory database.
 */

```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
class AddItemFragment : Fragment() {

    // Use the 'by activityViewModels()' Kotlin property delegate from the fragment-ktx artifact
    // to share the ViewModel across fragments.
    private val viewModel: InventoryViewModel by activityViewModels {
        InventoryViewModelFactory(
            (activity?.application as InventoryApplication).database
                .itemDao()
        )
    }
    private val navigationArgs: ItemDetailFragmentArgs by navArgs()

    lateinit var item: Item

    // Binding object instance corresponding to the fragment_add_item.xml layout
    // This property is non-null between the onCreateView() and onDestroyView() lifecycle callbacks,
    // when the view hierarchy is attached to the fragment
    private var _binding: FragmentAddItemBinding? = null
    private val binding get() = _binding!!

    override fun onCreateView(
        inflater: LayoutInflater,
        container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        _binding = FragmentAddItemBinding.inflate(inflater, container, false)
        return binding.root
    }

    /**
     * Returns true if the EditTexts are not empty
     */
    private fun isEntryValid(): Boolean {
        return viewModel.isEntryValid(
            binding.itemName.text.toString(),
            binding.itemPrice.text.toString(),
            binding.itemCount.text.toString(),
        )
    }

    /**
     * Binds views with the passed in [item] information.
     */
    private fun bind(item: Item) {
        val price = "%.2f".format(item.itemPrice)
        binding.apply {

```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
        itemName.setText(item.itemName, TextView.BufferType.SPANNABLE)
        itemPrice.setText(price, TextView.BufferType.SPANNABLE)
        itemCount.setText(item.quantityInStock.toString(), TextView.BufferType.SPANNABLE)
        saveAction.setOnClickListener { updateItem() }
    }

}

/***
 * Inserts the new Item into database and navigates up to list fragment.
 */
private fun addNewItem() {
    if (isEntryValid()) {
        viewModel.addNewItem(
            binding.itemName.text.toString(),
            binding.itemPrice.text.toString(),
            binding.itemCount.text.toString(),
        )
        val action = AddItemFragmentDirections.actionAddItemFragmentToItemListFragment()
        findNavController().navigate(action)
    }
}

/***
 * Updates an existing Item in the database and navigates up to list fragment.
 */
private fun updateItem() {
    if (isEntryValid()) {
        viewModel.updateItem(
            this.navigationArgs.itemId,
            this.binding.itemName.text.toString(),
            this.binding.itemPrice.text.toString(),
            this.binding.itemCount.text.toString()
        )
        val action = AddItemFragmentDirections.actionAddItemFragmentToItemListFragment()
        findNavController().navigate(action)
    }
}

/***
 * Called when the view is created.
 * The itemId Navigation argument determines the edit item or add new item.
 * If the itemId is positive, this method retrieves the information from the database and
 * allows the user to update it.
 */
override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
    super.onViewCreated(view, savedInstanceState)
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
val id = navigationArgs.itemId
if (id > 0) {
    viewModel.retrieveItem(id).observe(this.viewLifecycleOwner) { selectedItem ->
        item = selectedItem
        bind(item)
    }
} else {
    binding.saveAction.setOnClickListener {
        addNewItem()
    }
}
}

/**
 * Called before fragment is destroyed.
 */
override fun onDestroyView() {
    super.onDestroyView()
    // Hide keyboard.
    val inputMethodManager = requireActivity().getSystemService(INPUT_METHOD_SERVICE) as
        InputMethodManager
    inputMethodManager.hideSoftInputFromWindow(requireActivity().currentFocus?.windowToken, 0)
    _binding = null
}
}
```

```
package com.example.inventory

import android.app.Application
import com.example.inventory.data.ItemRoomDatabase
```

```
class InventoryApplication : Application() {
    // Using by lazy so the database is only created when needed
    // rather than when the application starts
    val database: ItemRoomDatabase by lazy { ItemRoomDatabase.getDatabase(this) }
}
```

```
package com.example.inventory

import androidx.lifecycle.LiveData
import androidx.lifecycle.ViewModel
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
import androidx.lifecycle.ViewModelProvider
import androidx.lifecycle.asLiveData
import androidx.lifecycle.viewModelScope
import com.example.inventory.data.Item
import com.example.inventory.data.ItemDao
import kotlinx.coroutines.launch

/**
 * View Model to keep a reference to the Inventory repository and an up-to-date list of all items.
 */
class InventoryViewModel(private val itemDao: ItemDao) : ViewModel() {

    // Cache all items form the database using LiveData.
    val allItems: LiveData<List<Item>> = itemDao.getItems().asLiveData()

    /**
     * Returns true if stock is available to sell, false otherwise.
     */
    fun isStockAvailable(item: Item): Boolean {
        return item.quantityInStock > 0
    }

    /**
     * Updates an existing Item in the database.
     */
    fun updateItem(
        itemId: Int,
        itemName: String,
        itemPrice: String,
        itemCount: String
    ) {
        val updatedItem = getUpdatedItemEntry(itemId, itemName, itemPrice, itemCount)
        updateItem(updatedItem)
    }

    /**
     * Launching a new coroutine to update an item in a non-blocking way
     */
    private fun updateItem(item: Item) {
        viewModelScope.launch {
            itemDao.update(item)
        }
    }
}
```

<https://github.com/vinaynpp/mcc>



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
/**  
 * Decreases the stock by one unit and updates the database.  
 */  
fun sellItem(item: Item) {  
    if (item.quantityInStock > 0) {  
        // Decrease the quantity by 1  
        val newItem = item.copy(quantityInStock = item.quantityInStock - 1)  
        updateItem(newItem)  
    }  
}  
  
/**  
 * Inserts the new Item into database.  
 */  
fun addNewItem(itemName: String, itemPrice: String, itemCount: String) {  
    val newItem = getNewItemEntry(itemName, itemPrice, itemCount)  
    insertItem(newItem)  
}  
  
/**  
 * Launching a new coroutine to insert an item in a non-blocking way  
 */  
private fun insertItem(item: Item) {  
    viewModelScope.launch {  
        itemDao.insert(item)  
    }  
}  
  
/**  
 * Launching a new coroutine to delete an item in a non-blocking way  
 */  
fun deleteItem(item: Item) {  
    viewModelScope.launch {  
        itemDao.delete(item)  
    }  
}  
  
/**  
 * Retrieve an item from the repository.  
 */  
fun retrieveItem(id: Int): LiveData<Item> {  
    return itemDao.getItem(id).asLiveData()  
}  
  
/**  
 * Returns true if the EditTexts are not empty  
 */
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
/*
fun isEntryValid(itemName: String, itemPrice: String, itemCount: String): Boolean {
    if (itemName.isBlank() || itemPrice.isBlank() || itemCount.isBlank()) {
        return false
    }
    return true
}

/**
 * Returns an instance of the [Item] entity class with the item info entered by the user.
 * This will be used to add a new entry to the Inventory database.
 */
private fun getNewItemEntry(itemName: String, itemPrice: String, itemCount: String): Item {
    return Item(
        itemName = itemName,
        itemPrice = itemPrice.toDouble(),
        quantityInStock = itemCount.toInt()
    )
}

/**
 * Called to update an existing entry in the Inventory database.
 * Returns an instance of the [Item] entity class with the item info updated by the user.
 */
private fun getUpdatedItemEntry(
    itemId: Int,
    itemName: String,
    itemPrice: String,
    itemCount: String
): Item {
    return Item(
        id = itemId,
        itemName = itemName,
        itemPrice = itemPrice.toDouble(),
        quantityInStock = itemCount.toInt()
    )
}

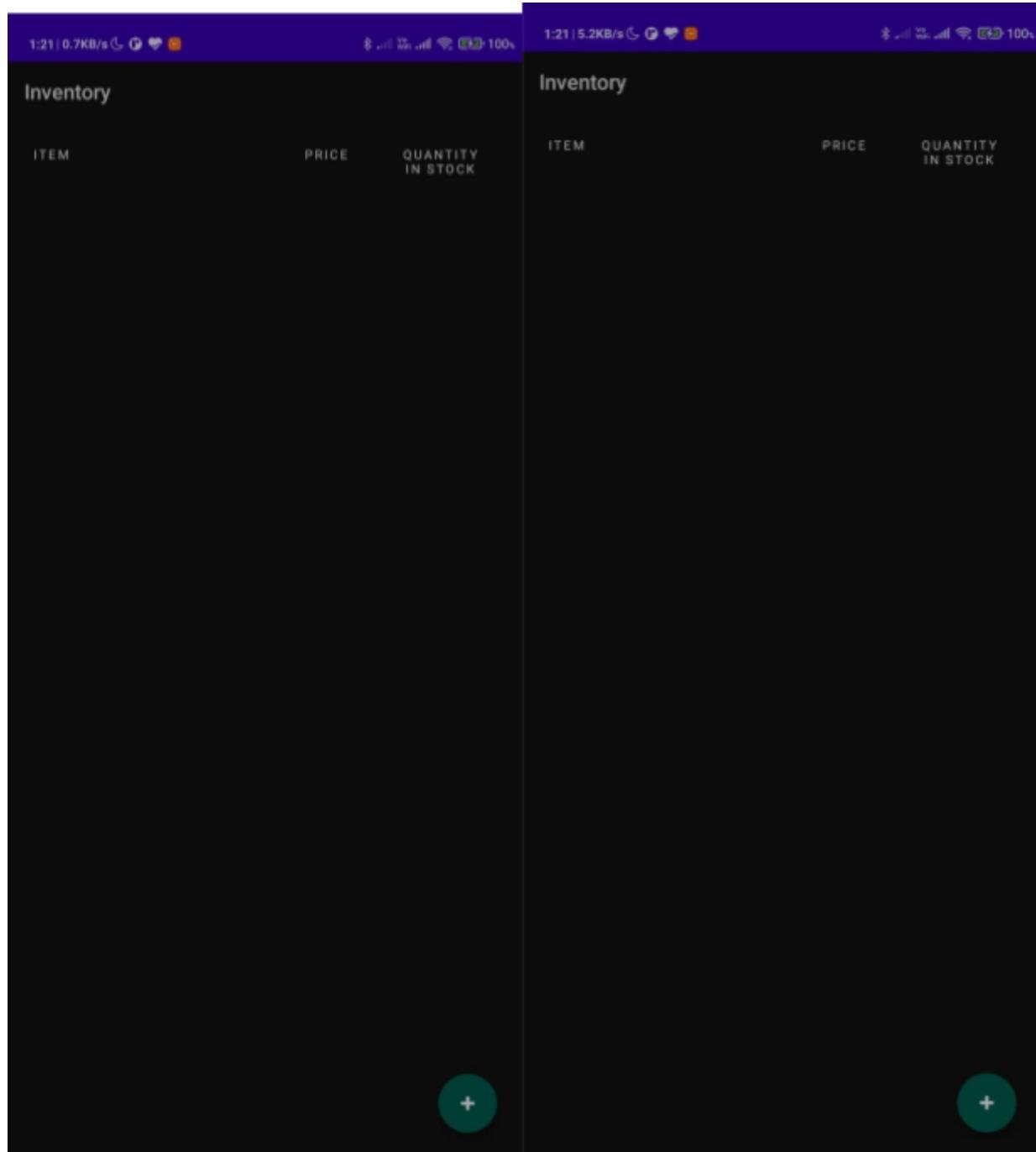
/**
 * Factory class to instantiate the [ViewModel] instance.
 */
class InventoryViewModelFactory(private val itemDao: ItemDao) : ViewModelProvider.Factory {
    override fun <T : ViewModel> create(modelClass: Class<T>): T {
        if (modelClass.isAssignableFrom(InventoryViewModel::class.java)) {
            @Suppress("UNCHECKED_CAST")
            return InventoryViewModel(itemDao) as T
        }
        throw IllegalArgumentException("Unknown ViewModel class")
    }
}
```



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

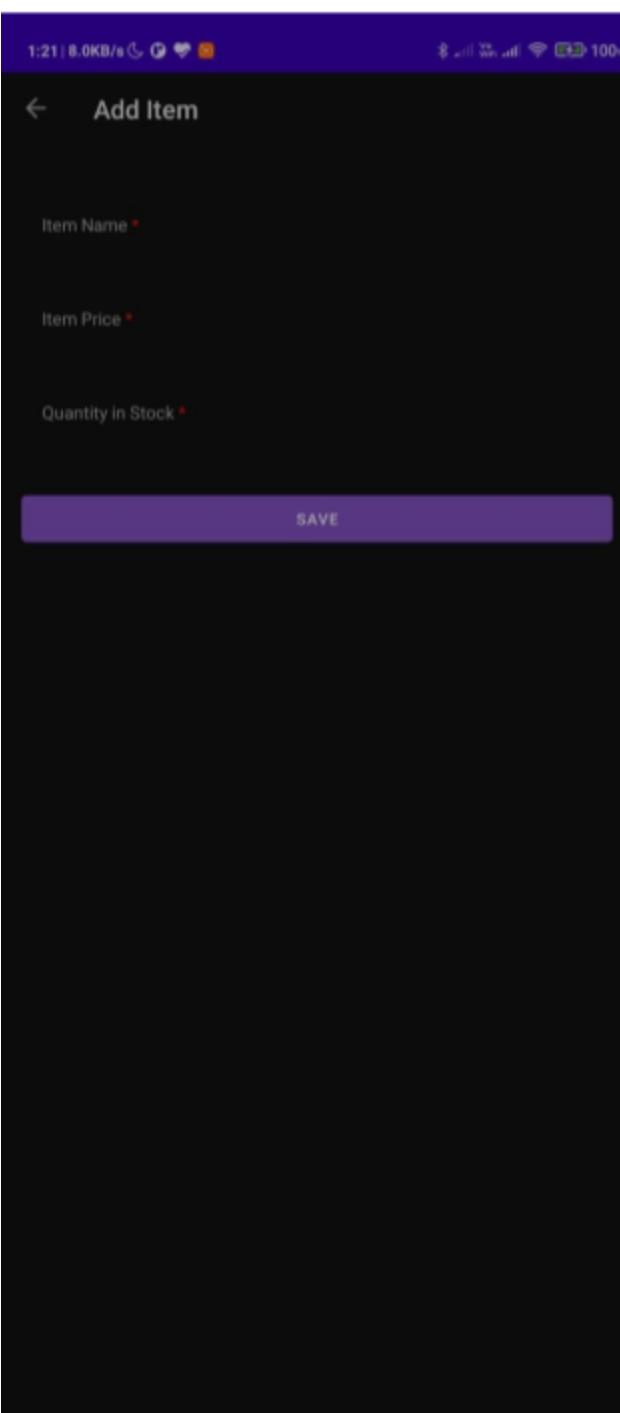
```
        return InventoryViewModel(itemDao) as T
    }
    throw IllegalArgumentException("Unknown ViewModel class")
}
}
```

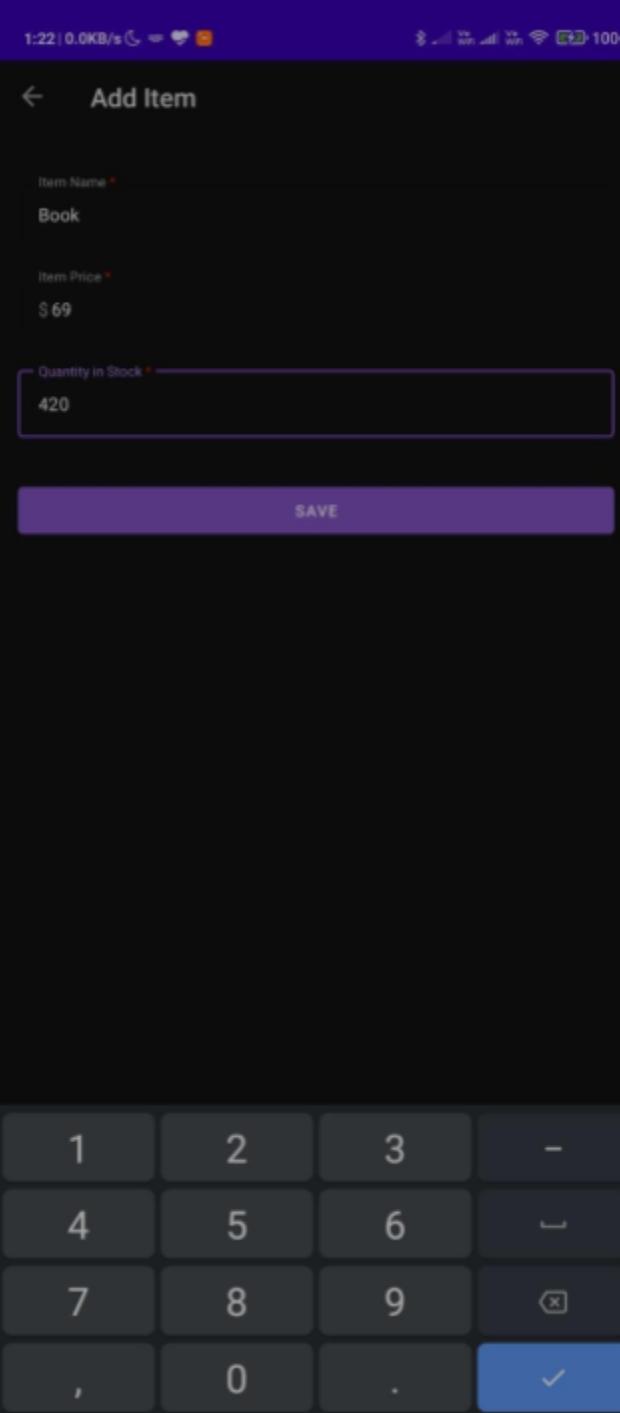
**Output:**





Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC





1	2	3	-
4	5	6	—
7	8	9	✖
,	0	.	✓



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

1:22 | 0.1KB/s

Inventory

ITEM	PRICE	QUANTITY IN STOCK
Book	₹69.00	420

+ +

1:22 | 0.7KB/s

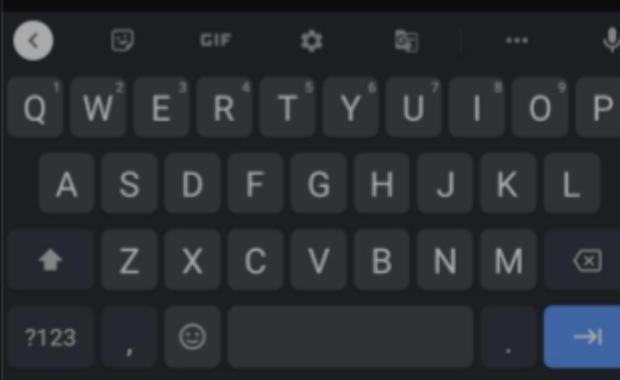
Add Item

Item Name \*

Item Price \*

Quantity in Stock \*

SAVE





**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

1:22 | 2.0KB/s ⓘ ❤️ 📺 1:23 | 0.0KB/s ⓘ ❤️ 📺 1:23 | 0.0KB/s ⓘ ❤️ 📺 100% 100%

**Inventory** ← **Item Details**

ITEM	PRICE	QUANTITY IN STOCK	Phone
Book	₹69.00	420	₹1,23,456.00 Quantity in Stock: 986
Phone	₹1,23,456.00	987	<b>SELL</b>

**DELETE**

**+** **edit**



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

1:23 | 0.0KB/s ⏻ 🌐 ❤️ 📺

1:23 | 0.0KB/s ⏻ 🌐 WiFi ⏻ WiFi 🔋 100%

1:23 | 0.0KB/s ⏻ 🌐 WiFi ⏻ WiFi 🔋 100%

← Item Details

Phone  
₹1,23,456.00  
Quantity in Stock: 986

SELL

DELETE

Inventory

ITEM	PRICE	QUANTITY IN STOCK
Book	₹69.00	420

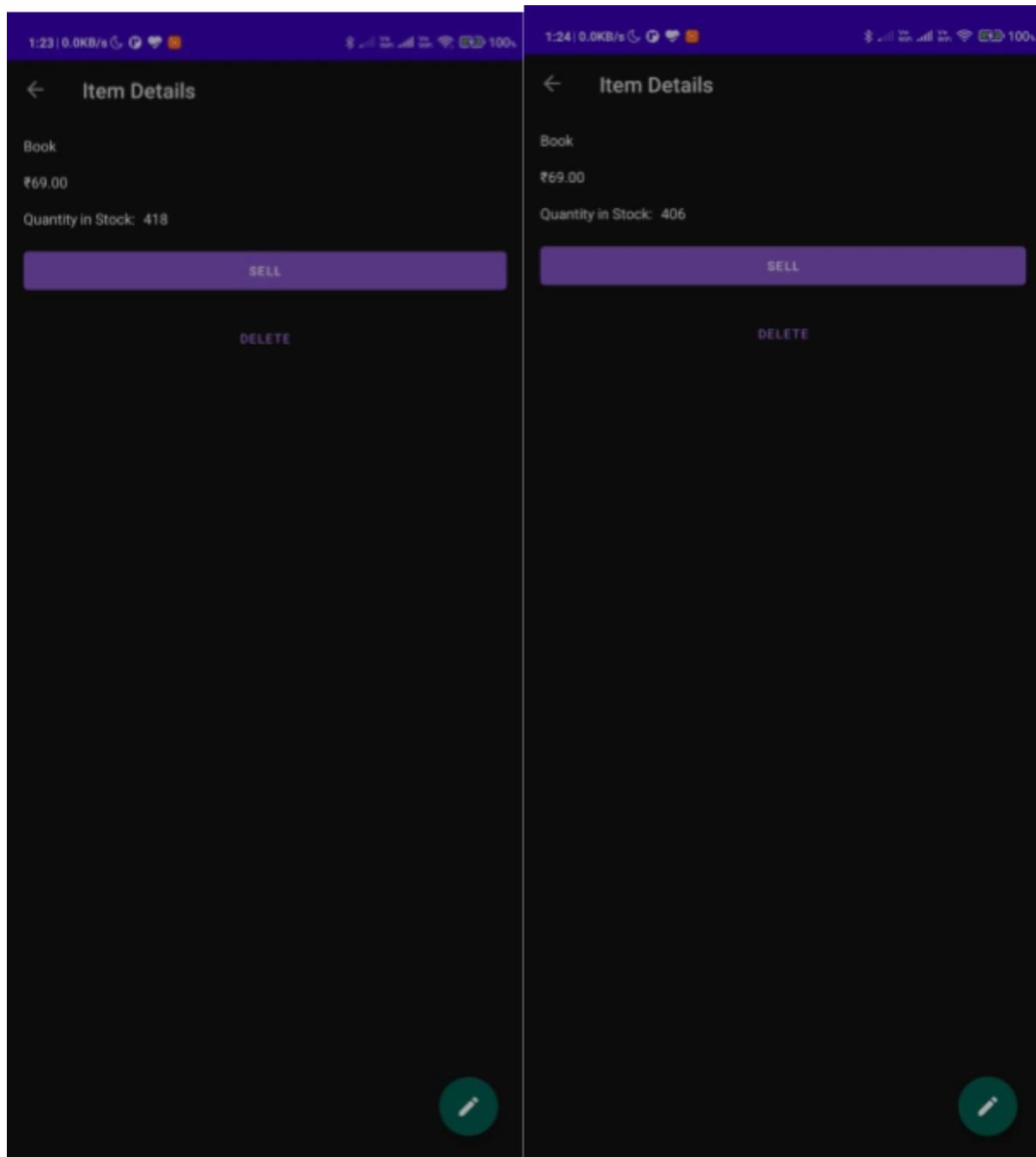
Attention  
Are you sure you want to delete?

NO YES

+



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC



**Conclusion:**

Thus, we have performed the experiment and made an app that uses SQLite database over here takes the student name and roll number and input and stores that in database which can be viewed later. We have successfully executed the experiment.



### Experiment No.: 10

**Aim:** Implement an application that creates an alert upon receiving a message.

#### **Theory:**

A notification is a message you can display to the user outside of your application's normal UI. When you tell the system to issue a notification, it first appears as an icon in the notification area. To see the details of the notification, the user opens the notification drawer. Both the notification area and the notification drawer are system-controlled areas that the user can view at any time.

Android **Toast** class provides a handy way to show users alerts but problem is that these alerts are not persistent which means alert flashes on the screen for a few seconds and then disappears.

To see the details of the notification, you will have to select the icon which will display notification drawer having detail about the notification. While working with emulator with virtual device, you will have to click and drag down the status bar to expand it which will give you detail as follows. This will be just **64 dp** tall and called normal view.

#### Create and Send Notifications

You have simple way to create a notification. Follow the following steps in your application to create a notification –

##### **Step 1 - Create Notification Builder**

As a first step is to create a notification builder using *NotificationCompat.Builder.build()*. You will use Notification Builder to set various Notification properties like its small and large icons, title, priority etc.

##### **Step 2 - Setting Notification Properties**

Once you have **Builder** object, you can set its Notification properties using Builder object as per your requirement. But this is mandatory to set at least following –

- A small icon, set by **setSmallIcon()**
- A title, set by **setContentTitle()**
- Detail text, set by **setContentText()**

##### **Step 3 - Attach Actions**

This is an optional part and required if you want to attach an action with the notification. An action allows users to go directly from the notification to an **Activity** in your application, where they can look at one or more events or do further work.

##### **Step 4 - Issue the notification**



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

Finally, you pass the Notification object to the system by calling NotificationManager.notify() to send your notification. Make sure you call **NotificationCompat.Builder.build()** method on



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

builder object before notifying it. This method combines all of the options that have been set and return a new **Notification** object.

**Code:**

```
package com.fernando.basicnotification

import android.app.Notification
import android.app.NotificationChannel
import android.app.NotificationManager
import android.content.Context
import android.os.Build
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import androidx.core.app.NotificationCompat
import androidx.core.app.NotificationManagerCompat
import
com.fernando.basicnotification.databinding.ActivityMainBinding
import java.nio.channels.Channels

class MainActivity : AppCompatActivity() {
    private lateinit var bindViews: ActivityMainBinding
    private lateinit var notificationManager: NotificationManagerCompat
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        bindViews = ActivityMainBinding.inflate(layoutInflater)
        setContentView(bindViews.root)
        notificationManager = NotificationManagerCompat.from(this)

        bindViews.btnChannel1.setOnClickListener {
            val title = bindViews.edtTitle.text.toString()
            val message = bindViews.edtMessage.text.toString()

            val notification = NotificationCompat.Builder(this, App.CHANNEL_ID_01)
                .setSmallIcon(R.drawable.ic_baseline_ring_volume_24)
                .setContentTitle(title)
                .setContentText(message)
                .setPriority(NotificationCompat.PRIORITY_HIGH)
                .setCategory(NotificationCompat.CATEGORY_MESSAGE)
                .build()
            notificationManager.apply {
                notify(1, notification)
            }
        }

        bindViews.btnChannel2.setOnClickListener {
            val title = bindViews.edtTitle.text.toString()
            val message = bindViews.edtMessage.text.toString()

            val notification = NotificationCompat.Builder(this, App.CHANNEL_ID_02)
                .setSmallIcon(R.drawable.ic_baseline_ring_volume_24)
```



**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**

.setContentTitle(title)  
.setContentText(message)  
.setPriority(NotificationCompat.PRIORITY\_LOW)



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

```
.build()
notificationManager.apply {
    notify(2, notification)
}

}
}

package com.fernando.basicnotification

import android.app.Application
import android.app.NotificationChannel
import android.app.NotificationManager
import android.content.Context
import android.os.Build

class App: Application() {

    override fun onCreate() {
        super.onCreate()
        createNotificationChannel()
    }

    private fun createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val channel1 = NotificationChannel(
                CHANNEL_ID_01,
                "Canal 1",
                NotificationManager.IMPORTANCE_DEFAULT).apply {
                    description = DESCRIPTION_CHANNEL1 }

            val channel2 = NotificationChannel(
                CHANNEL_ID_02,
                "Canal 2",
                NotificationManager.IMPORTANCE_LOW).apply {
                    description = DESCRIPTION_CHANNEL2
                }

            val manager = getSystemService(NotificationManager::class.java)
            manager.createNotificationChannels(listOf(channel1, channel2))
        }
    }

    companion object{
        const val CHANNEL_ID_01 = "channel1"
        const val CHANNEL_ID_02 = "channel2"
        const val CHANNEL_FERNANDO = "fernando"
        const val DESCRIPTION_CHANNEL1 = "Descrição do canal do Fernando 1"
        const val DESCRIPTION_CHANNEL2 = "Descrição do canal do Fernando 2"
    }
}

package com.fernando.basicnotification
```



**Vidya Vikas Education Trust's**  
**Universal College of Engineering, Kaman Road,**  
**Vasai-401212 Accredited by B+ Grade by NAAC**

import org.junit.Test



```
import org.junit.Assert.*  
  
/**  
 * Example local unit test, which will execute on the development machine (host).  
 *  
 * See [testing documentation](http://d.android.com/tools/testing).  
 */  
class ExampleUnitTest {  
    @Test  
    fun addition_isCorrect() {  
        assertEquals(4, 2 + 2)  
    }  
}  
  
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.fernando.basicnotification">  
  
    <application  
        android:name=".App"  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="@string/app_name"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportsRtl="true"  
        android:theme="@style/Theme.BasicNotification"  
    >  
        <activity android:name=".MainActivity">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

**Output:**



Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC

BasicNotification

Title

Message

NOTIFICATION CHANNEL 1

NOTIFICATION CHANNEL 2

BasicNotification

Title

Message

NOTIFICATION CHANNEL 1

NOTIFICATION CHANNEL 2

q w e r t y u i o p

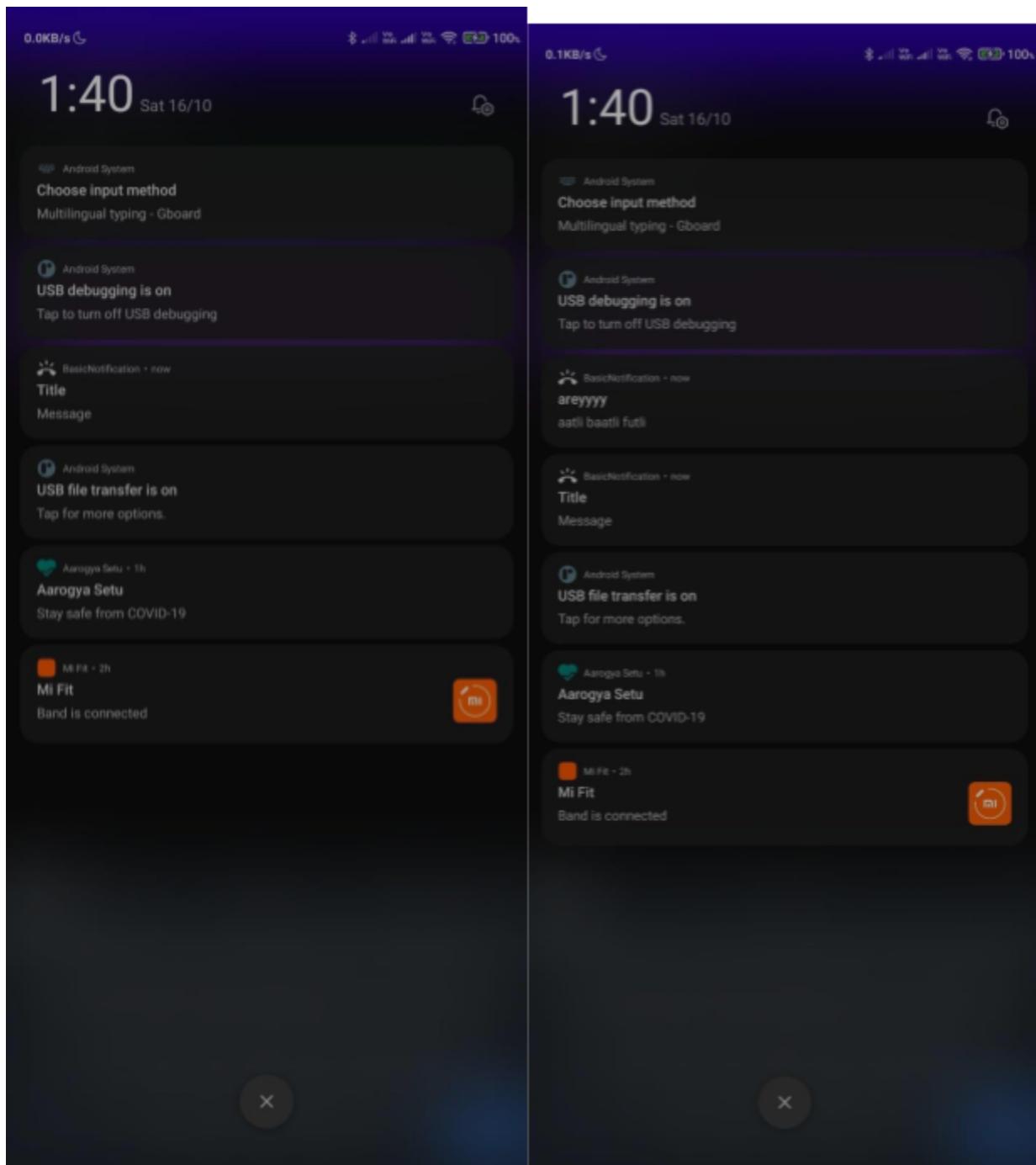
a s d f g h j k l

z x c v b n m

?123 , . →!

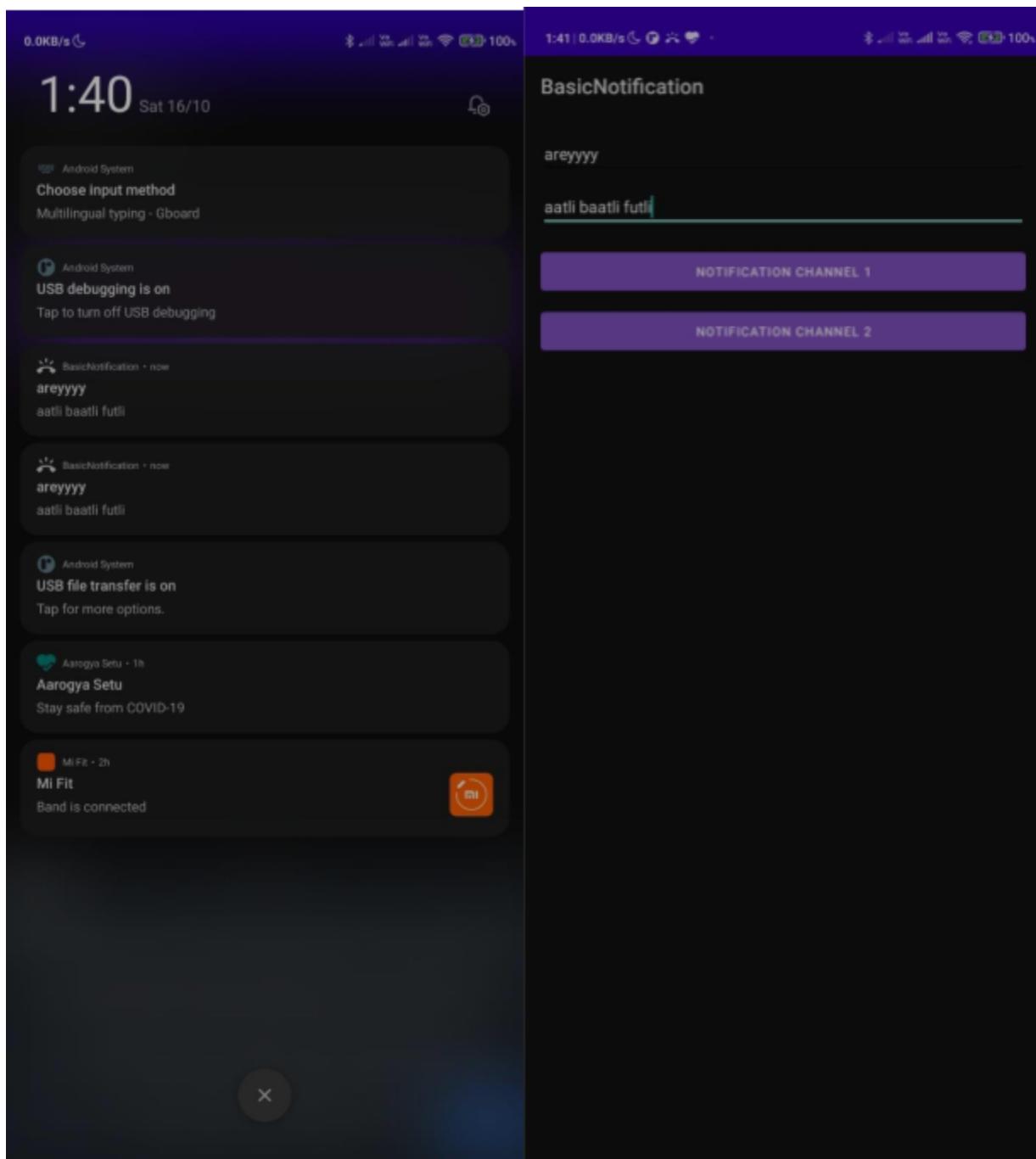


Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC





**Vidya Vikas Education Trust's  
Universal College of Engineering, Kaman Road,  
Vasai-401212 Accredited by B+ Grade by NAAC**



**Conclusion:**

Thus, we have performed the experiment in which we have developed an app where a alert is created when a message is sent and we have executed it properly.