**LAB MANUAL**

**Mobile Computing Lab**

**Name: Mayank Satish Pippal**

**Class: TE Computer Engineering**

**Sem: VI Roll No:92**

# 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 | |
| **1** | Develop and demonstrate mobile applications using various tools |
| **2** | Articulate the knowledge of GSM, CDMA & Bluetooth technologies and demonstrate it. |
| **3** | Students will able to carry out simulation of frequency reuse, hidden/exposed terminal problem |
| **4** | Implement security algorithms for mobile communication network |
| **5** | 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 |  |
| 2 | Implementation a Bluetooth network with application as transfer of a file from one device to another. |  |
| 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. |  |
| 4 | To implement Mobile node discovery. |  |
| 5 | Implementation of GSM security algorithms (A3/A5/A8) |  |
| 6 | Illustration of Hidden Terminal Problem (NS-2) |  |
| 7 | Develop an application that uses GUI components. |  |
| 8 | Write an application that draws basic graphical primitives on the screen. |  |
| 9 | Develop an application that makes use of database. |  |
| 10 | Implement an application that creates an alert upon receiving a message. |  |

## Experiment No.: 1 GUI

**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: Main Acitivity.java

## package com.example.mayankapplication;

## import android.os.Bundle;

## import com.google.android.material.snackbar.Snackbar;

## import androidx.appcompat.app.AppCompatActivity;

## import android.view.View;

## import androidx.navigation.NavController;

## import androidx.navigation.Navigation;

## import androidx.navigation.ui.AppBarConfiguration;

## import androidx.navigation.ui.NavigationUI;

## import com.example.mayankapplication.databinding.ActivityMainBinding;

## import android.view.Menu;

## import android.view.MenuItem;

## public class MainActivity extends AppCompatActivity {

## private AppBarConfiguration appBarConfiguration;

## private ActivityMainBinding binding;

## @Override

## protected void onCreate(Bundle savedInstanceState) {

## super.onCreate(savedInstanceState);

## binding = ActivityMainBinding.inflate(getLayoutInflater());

## setContentView(binding.getRoot());

## setSupportActionBar(binding.toolbar);

## NavController navController = Navigation.findNavController(this, R.id.nav\_host\_fragment\_content\_main);

## appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph()).build();

## NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);

## binding.fab.setOnClickListener(new View.OnClickListener() {

## @Override

## public void onClick(View view) {

## Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH\_LONG)

## .setAction("Action", null).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) {

## return true;

## }

## return super.onOptionsItemSelected(item);

## }

## @Override

## public boolean onSupportNavigateUp() {

## NavController navController = Navigation.findNavController(this, R.id.nav\_host\_fragment\_content\_main);

## return NavigationUI.navigateUp(navController, appBarConfiguration)

## || super.onSupportNavigateUp();

## }

## }

## Fragment\_first.xml

## <?xml version="1.0" encoding="utf-8"?>

## <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

## xmlns:app="http://schemas.android.com/apk/res-auto"

## xmlns:tools="http://schemas.android.com/tools"

## android:layout\_width="match\_parent"

## android:layout\_height="match\_parent"

## tools:context=".FirstFragment">

## <TextView

## android:background="@color/purple\_200"

## android:id="@+id/textview\_first"

## android:layout\_width="wrap\_content"

## android:layout\_height="wrap\_content"

## android:text="@string/hello\_first\_fragment"

## app:layout\_constraintBottom\_toTopOf="@id/button\_first"

## app:layout\_constraintEnd\_toEndOf="parent"

## app:layout\_constraintHorizontal\_bias="0.64"

## app:layout\_constraintStart\_toStartOf="parent"

## app:layout\_constraintTop\_toTopOf="parent" />

## <Button

## android:id="@+id/button\_first"

## android:layout\_width="164dp"

## android:layout\_height="75dp"

## android:text="@string/next"

## app:layout\_constraintBottom\_toBottomOf="parent"

## app:layout\_constraintEnd\_toEndOf="parent"

## app:layout\_constraintHorizontal\_bias="0.0"

## app:layout\_constraintStart\_toStartOf="parent"

## app:layout\_constraintTop\_toBottomOf="@id/textview\_first" />

## <Button

## android:id="@+id/button"

## android:layout\_width="145dp"

## android:layout\_height="84dp"

## android:text="Button"

## tools:layout\_editor\_absoluteX="212dp"

## tools:layout\_editor\_absoluteY="435dp" />

## </androidx.constraintlayout.widget.ConstraintLayout>

## String.xml

## <resources>

## <string name="app\_name">Mayank Application</string>

## <string name="action\_settings">Settings</string>

## <!-- Strings used for fragments for navigation -->

## <string name="first\_fragment\_label">SIMPLE APPLICATION</string>

## <string name="second\_fragment\_label">MOBILE APPLICATION</string>

## <string name="next">NEXT</string>

## <string name="previous">PREVIOUS</string>

## <string name="hello\_first\_fragment">HELLO WORLD!</string>

## <string name="hello\_third\_fragment">This is my app!</string>

## <string name="hello\_second\_fragment">Hello second fragment. Arg: %1$s</string>

## </resources>

## Colours.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="purple\_200">#FFBB86FC</color>

<color name="purple\_500">#FF6200EE</color>

<color name="purple\_700">#FF3700B3</color>

<color name="teal\_200">#FF03DAC5</color>

<color name="teal\_700">#FF018786</color>

<color name="black">#FF000000</color>

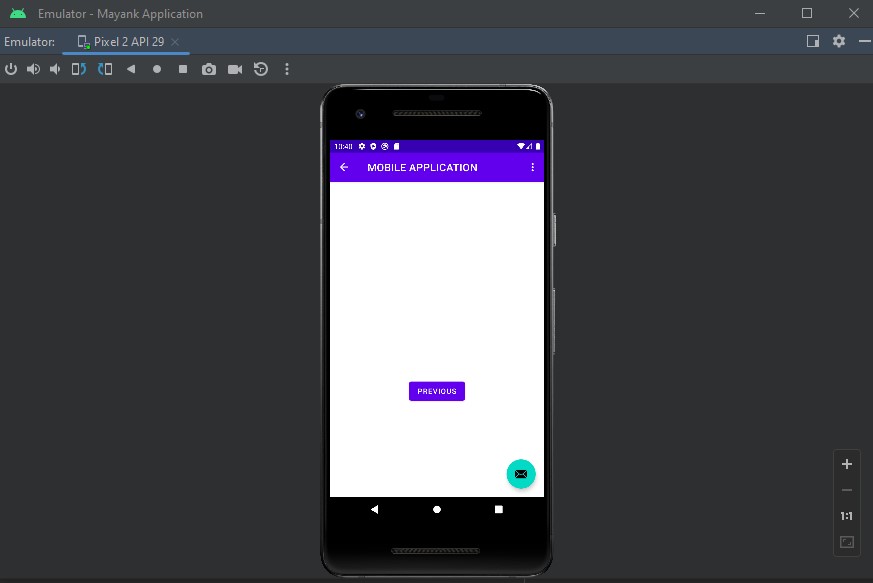
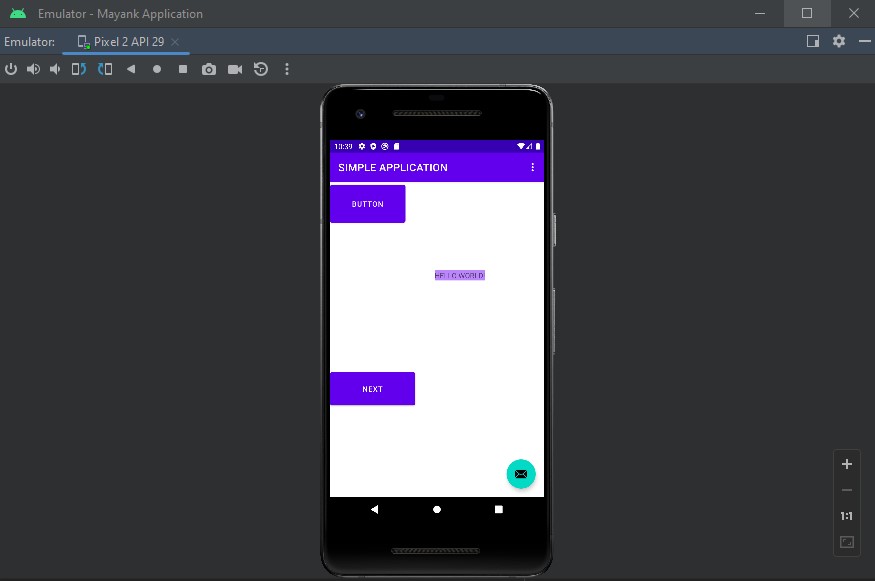
<color name="white">#FFFFFFFF</color>

<color name="screenBackground">#FFEE58</color>

</resources>

**GITHUB-LINK** 🡪 https://github.com/minto702/JavaGUI

**OUTPUT:**

****

## Experiment No.:2 BLUETOOTH

**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](https://en.wikipedia.org/wiki/ISM_band) [medical](https://en.wikipedia.org/wiki/ISM_band) [radio bands, from 2.402 GHz to 2.480 GHz, and building personal area networks (PANs). It was](https://en.wikipedia.org/wiki/ISM_band) originally conceived as a wireless alternative to [RS-232](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Bluetooth_Special_Interest_Group) 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/ (your**](https://github.com/_________(your) link)

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&lt;String&gt; fileList = new ArrayList&lt;String&gt;();

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.opendailog);

send=(Button)findViewById(R.id.sendBtooth);

buttonopenDailog.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dataPath.setText(&quot;&quot;);

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(&quot;File Selector&quot;);

dialog.setCancelable(true);

dialog.setCanceledOnTouchOutside(true);

textFolder = (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&lt;?&gt; 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&lt;String&gt; directoryList = new ArrayAdapter&lt;String&gt;(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,&quot;\*\*\* Now Bluetooth is off... Thanks.

\*\*\*&quot;,Toast.LENGTH\_LONG).show();

finish(); }

//Method for send file via bluetooth

public void sendViaBluetooth() {

if(!dataPath.equals(null)){

if (btAdatper == null) {

Toast.makeText(this, &quot;Device not support bluetooth&quot;, Toast.LENGTH\_LONG).show();

} else {

enableBluetooth();

}

}else{

Toast.makeText(this,&quot;Please select a file.&quot;,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 &amp;&amp; requestCode == REQUEST\_BLU) {

Intent i = new Intent();

i. setAction(Intent.ACTION\_SEND);

i.setType(&quot;\*/\*&quot;);

File file = new File(dataPath.getText().toString());

i.putExtra(Intent.EXTRA\_STREAM, Uri.fromFile(file));

PackageManager pm = getPackageManager();

List&lt;ResolveInfo&gt; list = pm.queryIntentActivities(i, 0);

if (list.size() &gt; 0) {

String packageName = null;

String className = null;

boolean found = false;

for (ResolveInfo info : list) {

packageName = info.activityInfo.packageName;

if (packageName.equals(&quot;com.android.bluetooth&quot;)) {

className = info.activityInfo.name;

found = true;

break;

}

}

//CHECK BLUETOOTH available or not------------------------------------------------

if (!found) {

Toast.makeText(this, &quot;Bluetooth not been found&quot;, Toast.LENGTH\_LONG).show();

} else {

i. setClassName(packageName, className);

startActivity(i);

}

}

} else {

Toast.makeText(this, &quot;Bluetooth is cancelled&quot;, 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, &quot;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\nDeveloper: Santosh Kumar Singh\nContact:

superssingh@gmail.com\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*&quot;, Toast.LENGTH\_LONG).show();

return true;

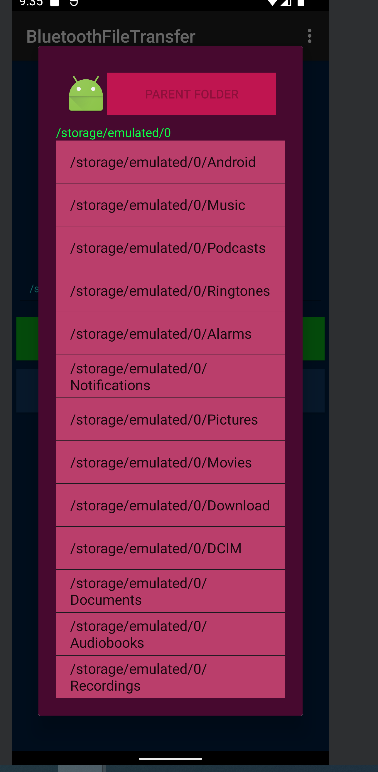
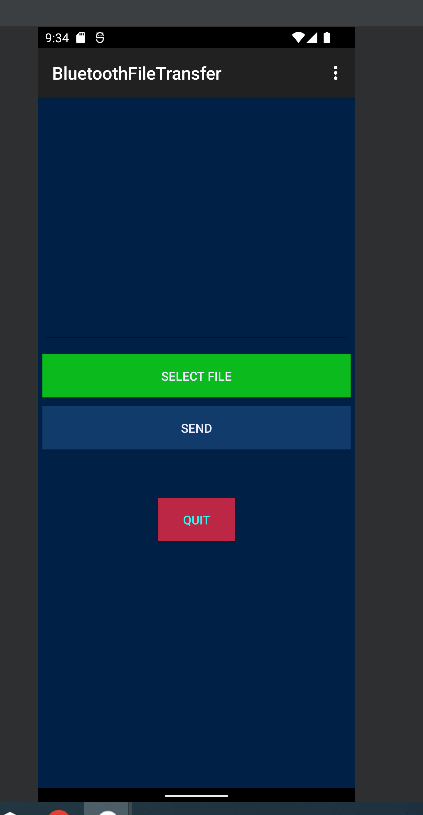
}

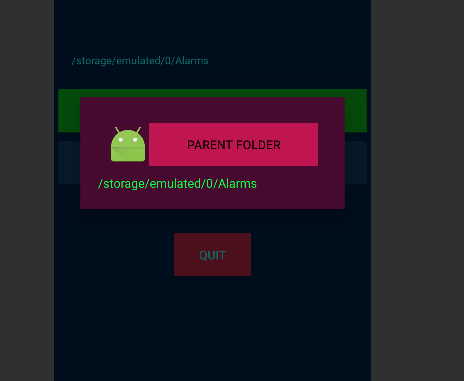
return super.onOptionsItemSelected(item);

}

}

**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.