Vidya Vikas Education Trust's
Universal College of Engineering, Kaman Road, Vasai-401208
Accredited B+ Grade NAAC

EXPERIMENT NO - 01

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

Theory:

Bluetooth: -

- Bluetooth is, with the infrared, one of the major wireless technologies developed to achieve WPAN. Bluetooth is a wireless LAN technology used to connect devices of different functions such as telephones, computers (laptop or desktop), notebooks, cameras, printers and so on.

- Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 30,000-member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE standardized Bluetooth as IEEE 802.15. l, 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. A network of patents applies to the technology, which are licensed to individual qualifying devices.

- Nowadays, Bluetooth technology is used for several computer and non-computer application:

1. It is used for providing communication between peripheral devices like wireless mouse or keyboard with the computer.
2. It is used by modern healthcare devices to send signals to monitors.
3. It is used by modern communicating devices like mobile phone, PDAs, palmtops etc to transfer data rapidly.
4. It is used for dial up networking. Thus, allowing a notebook computer to call via a mobile phone.
5. It is used for cordless telephoning to connect a handset and its local base station.

6. It also allows hands-free voice comml Inication with headset.

7. It also enables a mobile computer to connect to a fixed LAN.

8. It can also be used for file transfer operations from one mobile phone to another.

9. Bluetooth uses omni directional radio waves that can through walls or other non-metal barriers.

Bluetooth devices have a built-in short-range radio transmitter. The rate provided is IMbps and uses 2.4 GHz bandwidth.

Bluetooth is that when the device is within the scope of a other devices automatically start the transfer information without the user noticing. a small network between the devices is created and the user can have accessed as if there were cables.

Bluetooth Architecture : -
Bluetooth architecture defines two types of networks:

I. Piconet

2. Scatternet

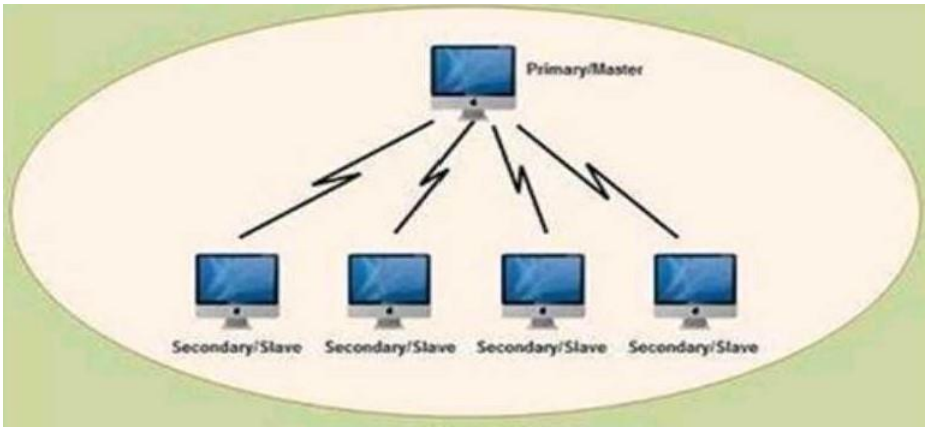
1. Piconet

- Piconet is a Bluetooth network that consists of one primary (master) node and seven active secondary (slave) nodes.

- Thus, piconet can have up to eight active nodes (1 master and 7 slaves) or stations within the distance of 10 meters.

- There can be only one primary or master station in each piconet.

- The communication between the primary and the secondary can be one-to-one or one-tomany.

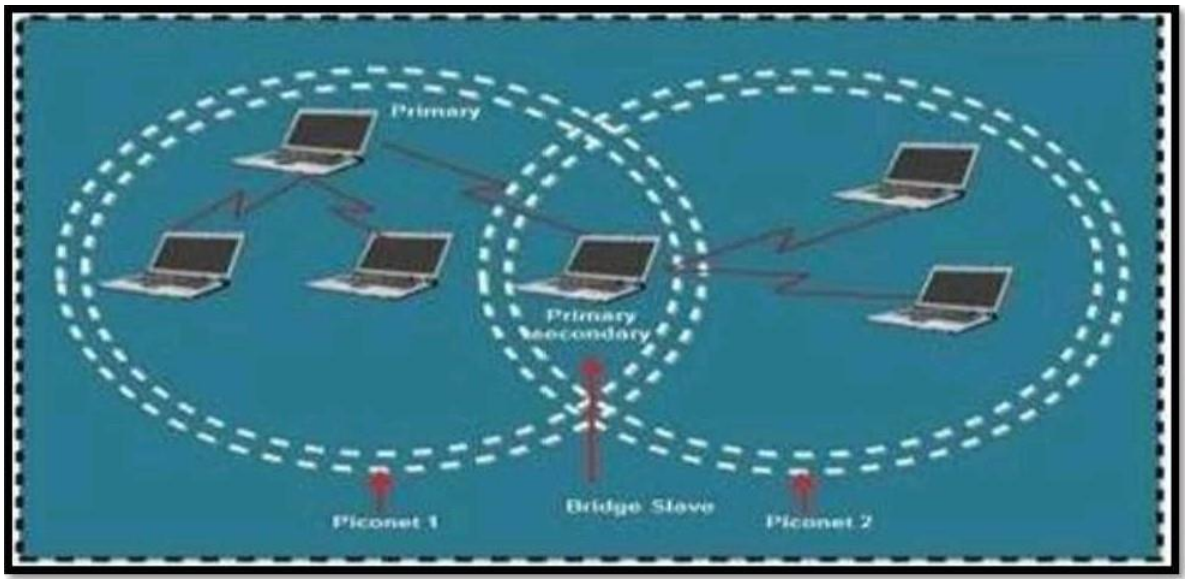


Piconet

- All communication is between master and a slave. Salve-slave communication is not possible.
- In addition to seven active slave station, a piconet can have up to 255 parked nodes. These parked nodes are secondary or slave stations and cannot take part in communication until it is moved from parked state to active state.

2. Scatternet

- Scatternet is formed by combining various piconets.
- A slave in one piconet can act as a master or primary in other piconet.
- Such a station or node can receive messages from the master in the first piconet and deliver the message to its slaves in other piconet where it is acting as master. This node is also called bridge slave.
- Thus a station can be a member of two piconets.
- A station cannot be a master in two piconets



Bluetooth pairing

In order that devices can connect easily and quickly, a scheme known as Bluetooth pairing may be used. Once Bluetooth pairing has occurred two devices may communicate with eachother. Bluetooth pairing is generally initiated manually by a device user. The Bluetooth link for the device is made visible to other devices. They may then be paired.

The Bluetooth pairing process is typically triggered automatically the first time a device receives a connection request from a device with which it is not yet paired. In order that Bluetooth pairing may occur, a password has to be exchanged between the two devices. This password or "Passkey" as it is more correctly termed is a code shared by both Bluetooth devices. It is used to ensure that both users have agreed to pair with each other. The process of Bluetooth pairing is summarised below:

- Bluetooth device looks for other Bluetooth devices in range:To be found by other Bluetooth devices, the first device, Device 1 must be set to discoverable mode — this will allow other Bluetooth devices in the vicinity to detect its presence and attempt to establish a connection.

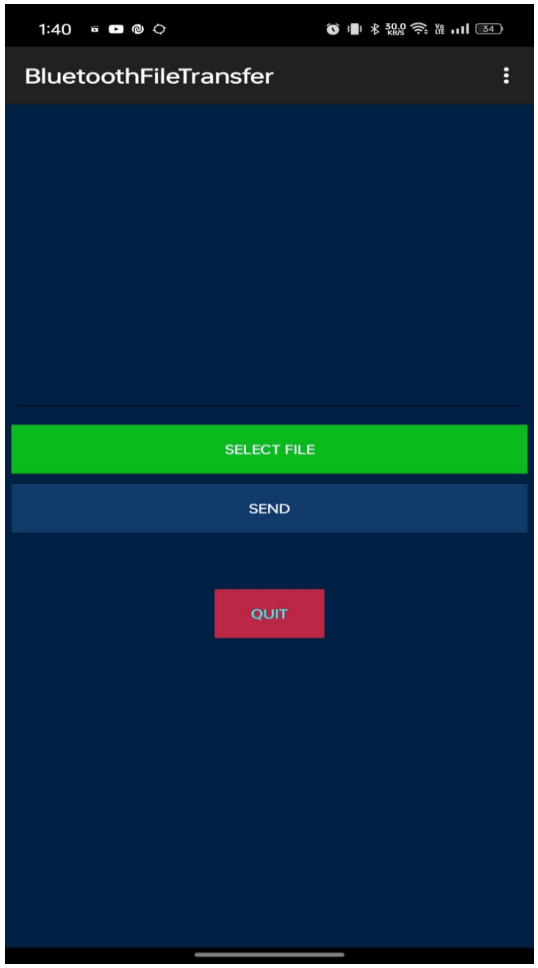
•Two Bluetooth devices find each other:When the two devices: Device 1 and device 2 find each other it is possible to detect what they are. Normally the discoverable device will indicate what type of device it is - cellphone, headset, etc., along with its Bluetooth device name. The Bluetooth device name is the can be allocated by the user, or it will be the one allocated during manufacture,

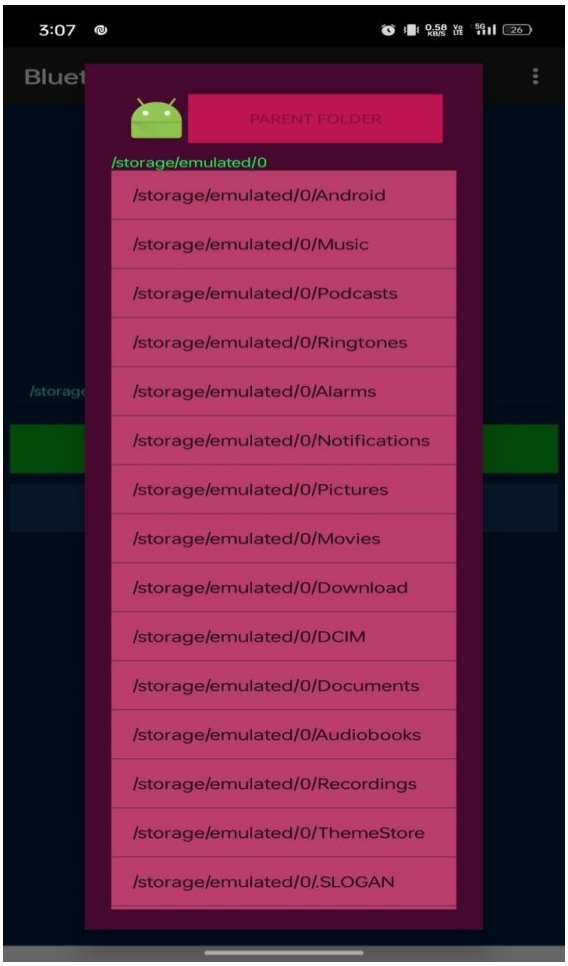
•Prompt for Passkey: Often the default passkey is set to "0000", but it is advisable to use something else as hackers will assume most people will not change this,

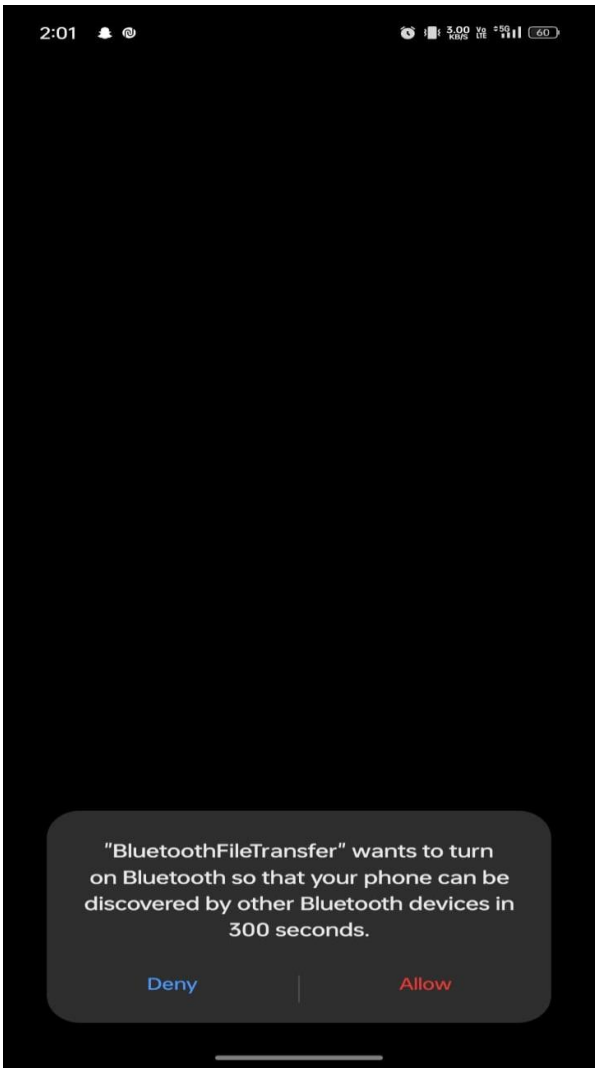
Communication is established: Once the Bluetooth pairing has occurred, data can be exchanged between the devices.

GITHUB LINK : https://github.com/shreyaucoe/Mobile-Computing

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 – 02

Aim : Write an application that draws basic painting on the screen.

Theory :

Painting in Android Studio involves creating custom drawing and painting functionalities within your Android application. This can include drawing shapes, lines, text, and applying various effects like colors, gradients, and transparency. Here's the theory behind implementing painting in Android Studio:

Canvas and Paint:

Canvas: In Android, a Canvas is an object that provides the drawing surface for your application. You can draw onto a Canvas using various drawing methods.

Paint: Paint is an object that holds the style and color information about how to draw geometries, text, and bitmaps. It defines how to draw shapes, colors, and styles.

Custom Views:

Painting in Android is often implemented within custom views. A custom view allows you to create unique user interface elements with custom drawing logic.

You can create a custom view by subclassing existing views like View or SurfaceView and oven•iding their onDraw(Canvas canvas) method.

Drawing Shapes and Lines:

Use methods like drawRect(), drawCircle(), drawLine(), etc., provided by the Canvas object to draw basic shapes and lines.

You can customize the appearance of shapes by setting properties in the Paint object, such as stroke width, color, and style.

Handling Touch Events:

To enable users to draw on the screen, you need to handle touch events like onTouchEvent(MotionEvent event) in your custom view.

Capture touch events (e.g., ACTION_DOWN, ACTION_MOVE, ACTION UP) to track user interactions with the screen.

Use the touch input coordinates to draw paths or shapes on the Canvas as the user interacts with the screen.

Implementing Undo/Redo Functionality:

Allow users to undo or redo their drawing actions. Maintain a stack of drawing commands or paths.

When the user performs an undo action, remove the last drawing command from the stack and redraw the canvas.

For redo, restore the removed drawing command from the undo stack and redraw the canvas.

Saving and Sharing Drawings:

Implement functionality to save the user's drawing as an image file (e.g., PNG or JPEG) locally on the device.

Provide options for users to share their drawings via other applications like messaging apps or social media platforms.

Optimizing Performance:

Drawing complex graphics or animations can impact performance, especially on older devices.

Optimize performance by minimizing unnecessary drawing operations, caching reusable graphic elements, and using hardware acceleration where possible.

Testing and Debugging:

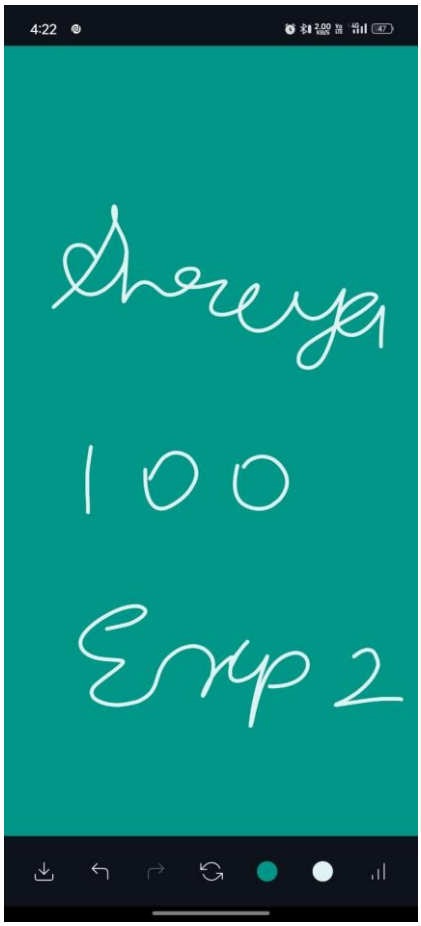
Thoroughly test your painting functionality on different screen sizes, resolutions, and Android versions to ensure compatibility.

Use debugging tools provided by Android Studio to identify and fix any issues with drawing logic or touch event handling.

By understanding these theoretical concepts, you can effectively implement painting functionality in your Android application using Android Studio. This allows you to create custom drawing and painting applications or integrate painting features into existing applications for various purposes like note-taking, sketching, or creative expression.

GITHUB LINK: **https://github.com/shreyaucoe/Mobile-Computing**

OUTPUT:



CONCLUSION:

Thus we have Implemented the application that draws basic painting on the screen.

EXPERIMENT NO - 03

Aim: Develop a native application that uses GPS location information.

Theory:

INTRODUCTION:

Location tracking has wide applications in the field of mobile application development. A common everyday example is a use of Internet and mobile map applications on smartphones like Find my device, Geo-Tracker etc that can calculate your current position using your GPS location and show where you're located on a map. Because your location coordinates are sent to a map application, your location is now known by the application, which is location tracking. So, this is an Android App that will access GPS to figure out your current location and keep updating the Marker on the map to give the effect of 'Tracking".

It particularly makes use of the Global Positioning System i.e. GPS using the latitude and longitude given as input by the user. It then points to that location notifying the user that the location coordinates are updated and saved in the Database.

The Global Positioning System (GPS), originally NAVSTAR GPS, is a satellitebased radio navigation system owned by the United States government and operated by the United States Air Force. It is a global navigation satellite system (GNSS) that provides geo- location and time information to a GPS receiver anywhere on or near the Earth where thereis an unobstructed line of sight to four or more GPS satellites. Obstacles such as mountains and buildings block the relatively weak GPS signals.

The GPS does not require the user to transmit any data, and it operates independently of any telephonic or internet reception, though these technologies can enhance the usefulness of the GPS positioning information.

The GPS provides critical positioning capabilities to military, civil, and commercial users around the world. The United States government created the system, maintains it, and makes it freely accessible to anyone with a GPS receiver.

FEATURES:

1. Track your own location just by clicking on a button.
2. Track other locations by using Latitude and Longitude coordinates of the location.
3. All the coordinates will be saved to Database as Tracking History.
4. Notification Alert will be received when the Data is saved to the database.
5. A Toast Notification alert for getting the last location.

IMPLEMENTATION:

To build this application, we used Open Source Android Studio based on Java and XML. The layout for the application was built using relative layout, buttons and some text boxes using xml.

Database used for the application is Google Firebase, which is connected to the application using SHA-I signature and a google-config file. We used Google Maps as the main layout. On the layout we use, GPS using Location listener and Location Manager.

Canvas was used to draw a white rectangle. On which all the textboxes and buttons reside. Using Notification Manager, we created notifications for our app which notifies once the location data is pushed onto the Firebase server. Using Toast Notifications, User can click on the button to know the last location which is read from the Firebase Server.

GITHUB LINK: https://github.com/shreyaucoe/Mobile-Computing\

OUTPUT:

CONCLUSION:

Thus we have successfully track GPS location in android studio.

EXPERIMENT NO - 04

Aim : Implementation of income tax/loan EMI calculator and deploy the same on real devices (Implementation of any real time application).

Theory :

EMI

EMI is a fixed payment amount made by a borrower to a lender at a specified date each calendar month. It is used to repay both the interest and principal amount of a loan over a predetermined period. The EMI calculation is based on three primary factors: principal amount, interest rate, and loan tenure.

Principal Amount (P): This is the initial amount of the loan, which is the total amount borrowed from the lender.

Interest Rate (R): This is the rate at which interest is charged on the loan amount. It is typically expressed as an annual percentage rate (APR). To calculate the monthly interest rate, divide the annual rate by 12.

Loan Tenure (N): This is the duration over which the loan is to be repaid, usually expressed in months.

The formula to calculate EMI is derived from the annuity formula for the present value of future cash flows:

$$E = (P.r.(1+rY) / ((1+r)^n - 1)$$

Where:

P = Principal loan amount r = Monthly interest rate (annual

interest rate divided by 12) n = Loan tenure in months

The numerator calculates the present value of the future cash flows, which represents the monthly payment required to repay the loan amount with interest over the specified period. The denominator represents the present value of the future cash flows discounted at the monthly interest rate.

INCOME TAX

Income tax is a tax imposed by the government on individuals or entities based on their income or profits earned during a financial year. The calculation of income tax involves several components, including taxable income, tax deductions, tax slabs, and tax rates.

Taxable Income: This is the total income earned by an individual or entity during a financial year after considering exemptions and deductions.

Tax Deductions: Certain expenses and investments are eligible for tax deductions under various sections of the Income Tax Act. These deductions reduce the taxable income, thereby reducing the tax liability.

Tax Slabs and Rates: Income tax is levied at different rates based on the income slab in which an individual falls. Tax rates may vary for different categories of taxpayers (such as individuals, HUFs, companies) and different income levels.

The formula to calculate income tax varies depending on the tax regime (e.g., progressive tax system, flat tax system) and the specific provisions of the tax laws applicable in a particular jurisdiction.

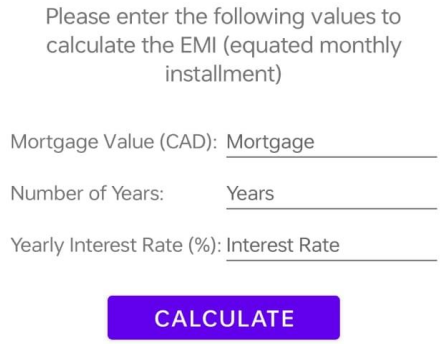
In progressive tax systems, such as the one commonly used in many countries, including India, the tax calculation involves applying different tax rates to different portions of the taxable income. The tax liability is determined by summing up the taxes payable on each slab of income.

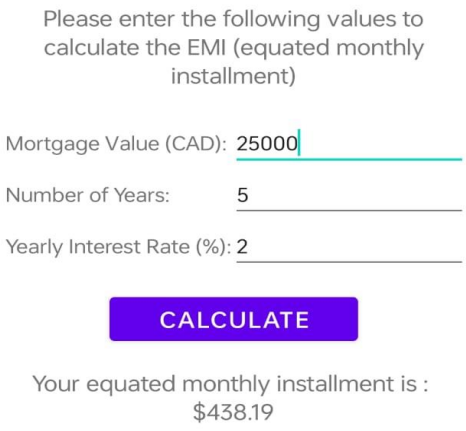
Tax calculations may also involve deductions, rebates, and surcharges, which further complicate the computation process. Taxpayers are often required to comply with various reporting and compliance requirements, including filing tax returns and maintaining records of income and expenses.

In summary, both EMI and income tax calculations involve complex mathematical formulas and considerations. Understanding these theories is essential for practical applications, such as developing financial tools and applications like EMI calculators and income tax calculator.

GITHUB LINK : https://github.com/shreyaucoe/Mobile-Computing\

OUTPUT :





CONCLUSION:

Thus we have successfully Implemented the income tax/loan EMI calculator and deploy the same on real devices.

EXPERIMENT NO - 05

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.

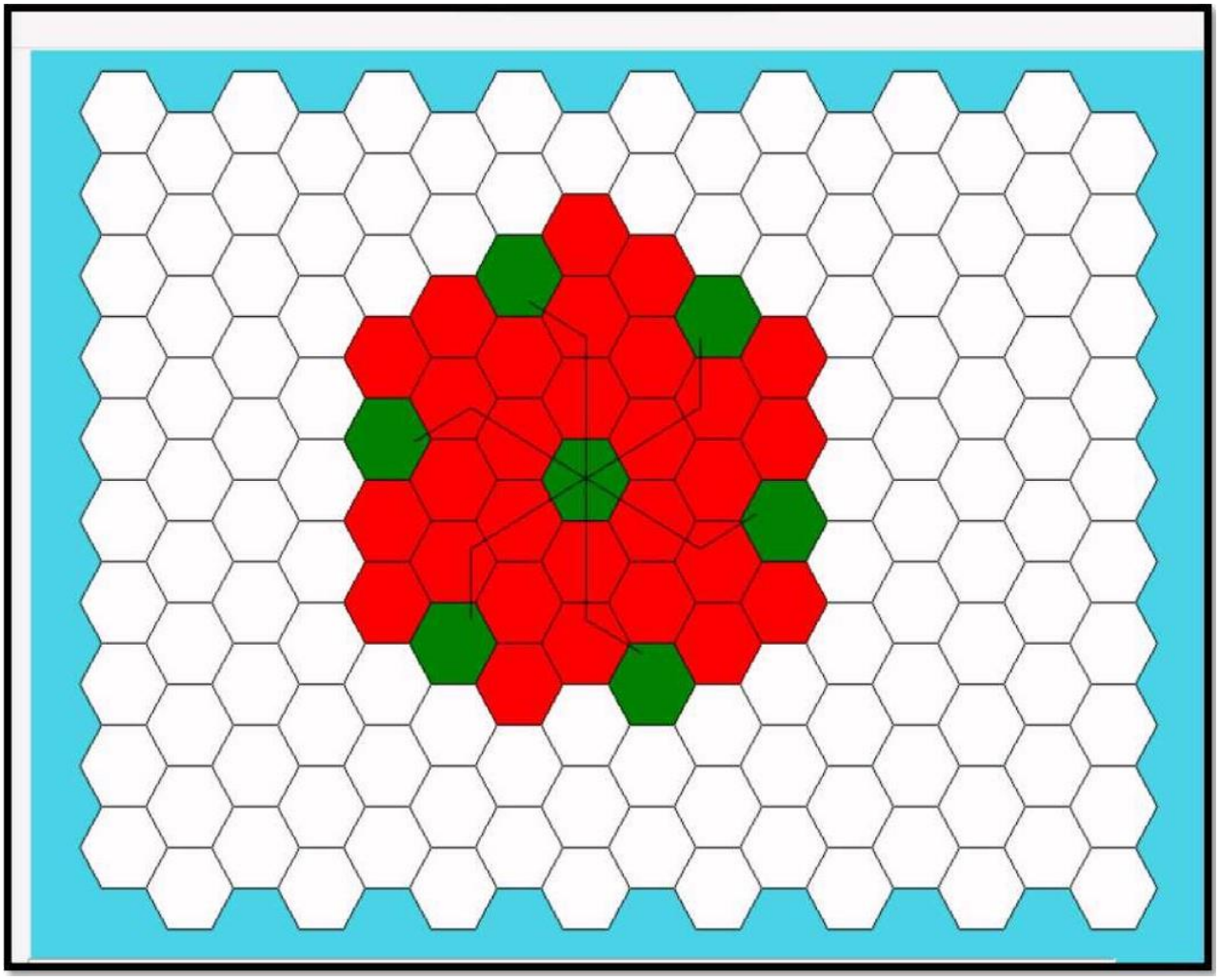
Co-channel Cells:

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,

$$N=iA2+ij+jA2$$

GITHUB LINK: https://github.com/shreyaucoe/Mobile-Computing\

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-06

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 View Groups

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 Group View. Example of View Group 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 View Group is a subclass of View and provides invisible container that hold other Views or other View Groups and define their layout properties.

At third level we have different layouts which are subclasses of View Group class and a typical layout defines the visual structure for an Android user interface and can be created either at run time using View/View Group 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.

Description:

1) Open android studio and select new android project

2) Give project name and select next

3) Choose the android version.

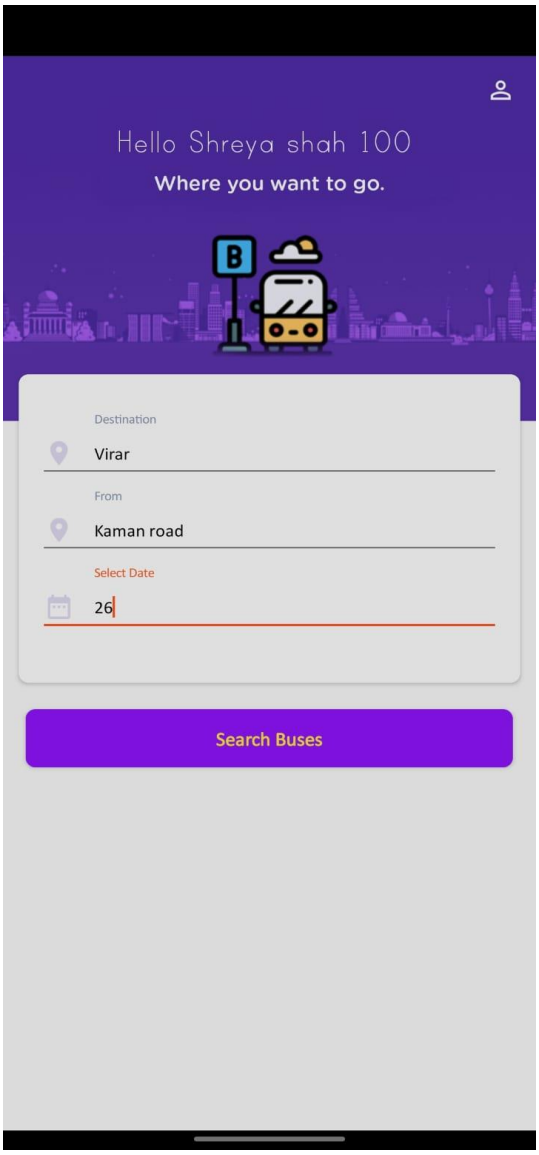
4) Enter the package name. Package name must be two word separated bycomma and click finish

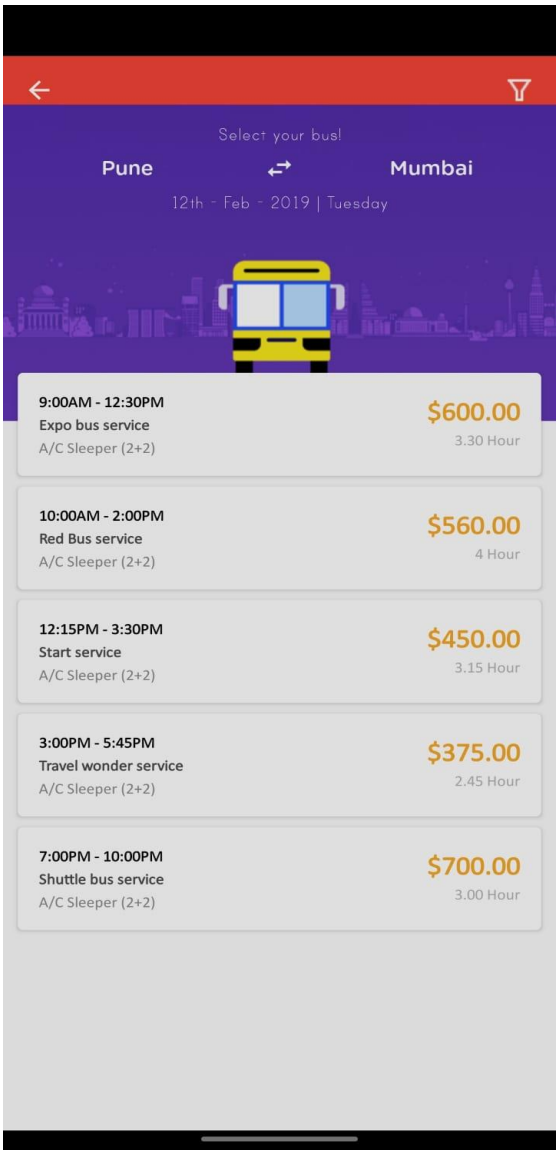
5) Go to package explorer in the left hand side. Select our project. 6) Go to res folder and select layout. Double click the main.xml file

7) Now you can see the Graphics layout window.

GITHUB LINK: https://github.com/shreyaucoe/Mobile-Computing\

OUTPUT :





CONCLUSION:

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

EXPERIMENT NO - 07

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, etc

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 open Or Create Database with your database name and mode as a parameter.

Database - Insertion we can create table or insert data into table using exec SQL method defined in SQLite Database 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 raw Query and it will return a result set 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 SQLite Open Helper. It automatically manages the creation and update of the database.

Description:

1)Open android studio and select new android project.

2)Give project name and select next

3)Choose the android version.

4) Enter the package name. Package name must be two word separated by comma and click finish

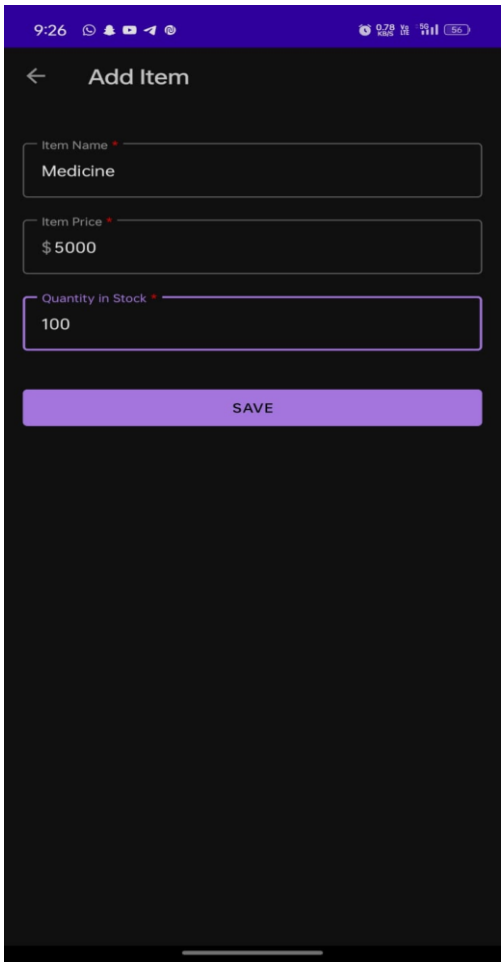
5)Go to package explorer in the left hand side. Select our project.

6)Go to res folder and select layout. Double click the main.xml file

7)Now you can see the Graphics layout window

GITHUB LINK: https://github.com/shreyaucoe/Mobile-Computing

OUTPUT:



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-08

Aim : Illustration of Hidden Terminal/Exposed terminal Problem.

Consider two Wi-fi base stations (STA) and an access point (AP) located along the x _axis. All the nodes are fixed. The AP is situated at the middle of the two STA, the distance of separation being 150 m. [variable]. Node #0 and node #1 are the hidden terminals. Both are transmitting some data to the AP (almost at same rate) at the same time. The loss across the wireless link between each STA and the AP is fixed at 50 dB irrespective of the distance of separation. To study how RTS/CTS helps in wireless networks,

1. No RTS/CTS is being sent.
2. Nodes do exchange RTS/CTS packets. Compare the no. of packet retransmissions required in both the cases (as obtained in the output) and compare the results.

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.

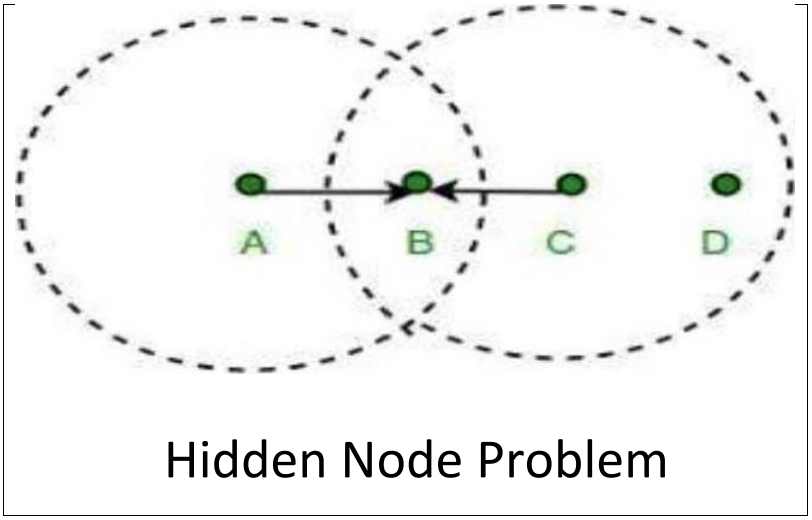
To overcome the hidden node problem, RTS/CTS handshaking (IEEE 802.11 RTS/CTS) is implemented in conjunction with the Carrier sense multiple

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

of

The solution of hidden terminal problem is as follows.

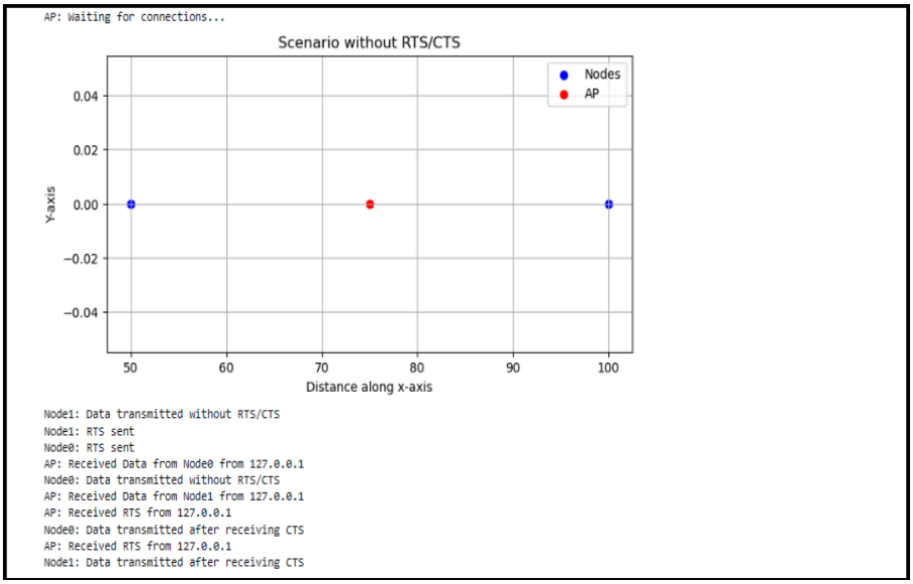
When A wants to send a packet to B, A first sends a Request-to-send (RTS) to B.

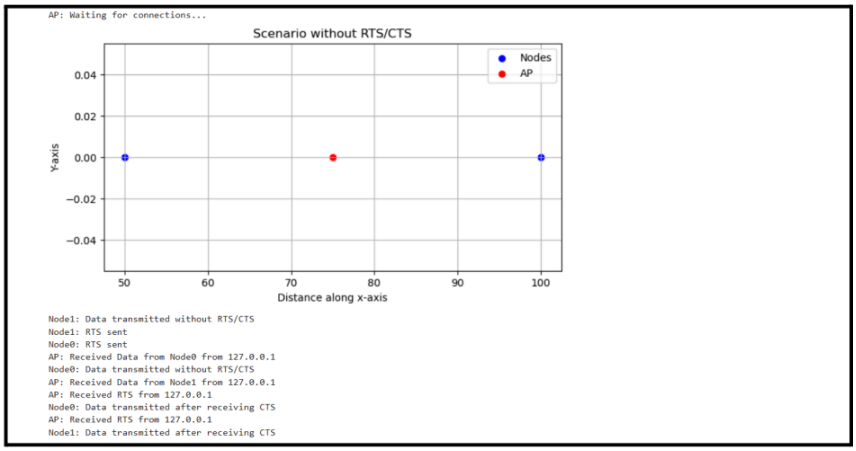
On receiving RTS, B responds by sending Clear-to-Send (CTS).

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.

GITHUB LINK: https://github.com/shreyaucoe/Mobile-Computing

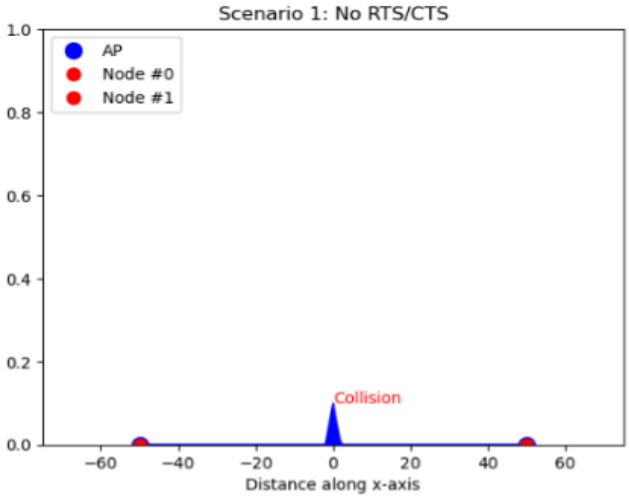
OUTPUT :

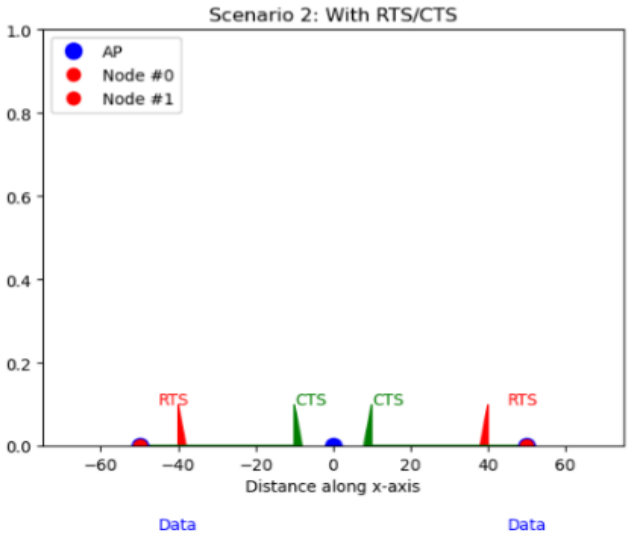


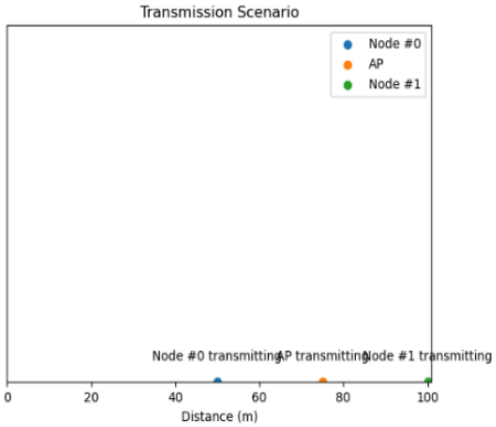


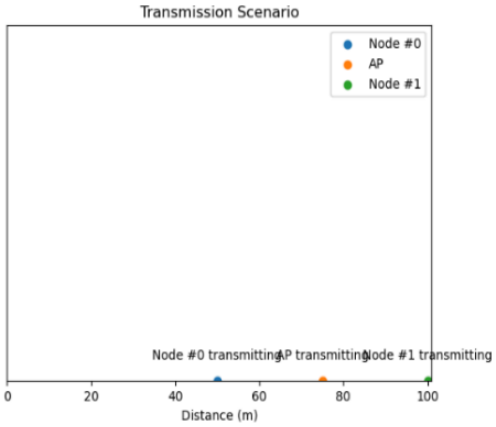
Distance (m)

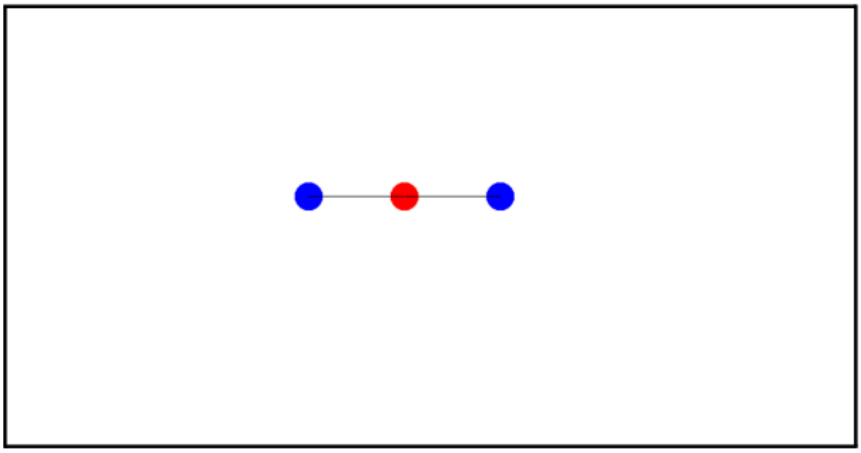
```
Scenario without RTS/CTS:
Node #0 transmitting to AP...
Transmission successful: Node #0 -> AP
Node #1 transmitting to AP...
Transmission successful: Node #1 -> AP

Scenario with RTS/CTS:
Node #0 sends RTS to AP
AP sends CTS to Node #0
Node #0 transmitting to AP...
Transmission successful: Node #0 -> AP
Node #1 waits for CTS from AP
AP sends CTS to Node #1
Node #1 transmitting to AP...
Transmission successful: Node #1 -> AP
Scenario without RTS/CTS:
Node #0 transmitting to AP...
Transmission successful: Node #0 -> AP
Node #1 transmitting to AP...
Transmission successful: Node #1 -> AP
```











CONCLUSION:

From this experiment, we learnt about the hidden terminal problem. This problem is a transmission problem that arises when two or more stations who are out of range of each other transmit simultaneously to a common recipient. We have performed this experiment using NS2 and properly the same which helps to understand this problem properly

EXPERIMENT NO-09

Aim : Write 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 colour 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.

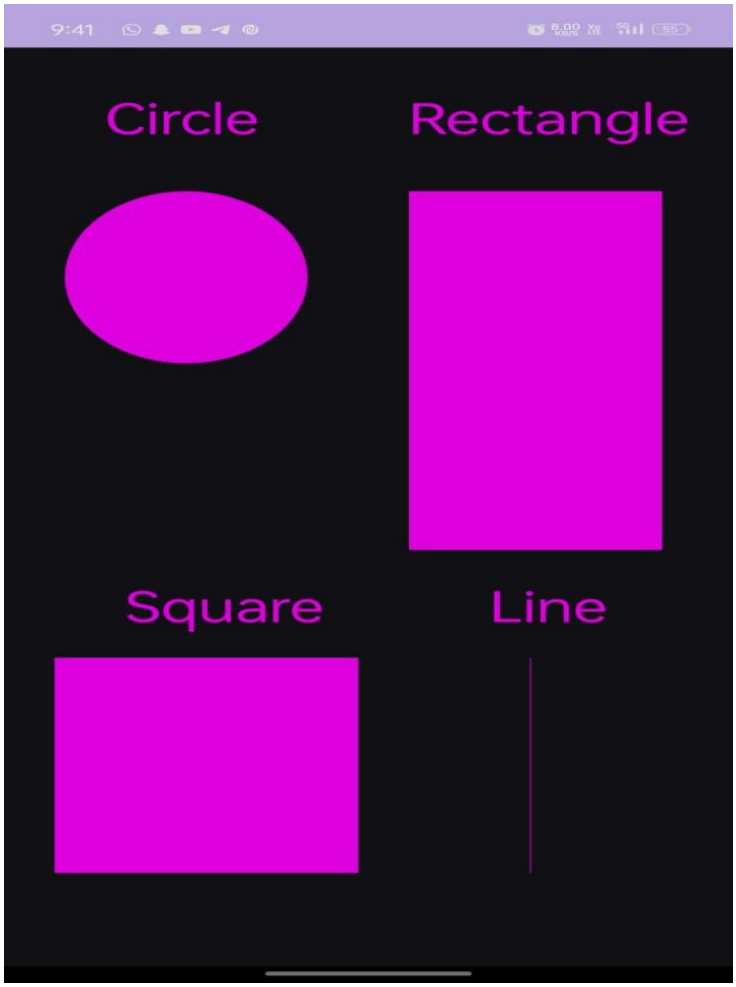
drawRoundRect: it draws a rectangle with round edges.

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 colour, style, font, special effects etc can be modified by modifying the paint object. For instance, set Colour method can be used to select the colour 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 and paint.

GITHUB LINK: **https://github.com/shreyaucoe/Mobile-Computing**

OUTPUT:



CONCLUSION:

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

EXPERIMENT NO – 10

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

Theory :

Notification is a kind of message, alert, or status of an application (probably running in the background) that is visible or available in the Android's UI elements. This application could be running in the background but not in use by the user. The purpose of a notification is to notify the user about a process that was initiated in the application either by the user or the system. This article could help someone who's trying hard to create a notification for developmental purposes.

Notifications could be of various formats and designs depending upon the developer. In General, one must have witnessed these four types of notifications:

1. Status Bar Notification (appears in the same layout as the current time, battery percentage)
2. Notification drawer Notification (appears in the drop-down menu)
3. Heads-Up Notification (appears on the overlay screen, ex: Whatsapp notification, OTP messages)
4. Lock-Screen Notification (I guess you know it)

PROCEDURE:

Create a new Android Application

l. In Eclipse go to File->New->Project

2. Select an Android Project from the Android Folder and press Next.

3. Fill in the details of your Android application.

a. Project Name: The project name and folder that Eclipse will store the project files
b. Build Target: The version of the Android SDK that will be used when you build your program. Select a platform that is equal to or lower than the target chosen for the AVD.

c. Application Name: This is the name of the application.
d. Package Name: The namespace that all of the source code will reside under.
e. Create Activity: The name for that class stub that is generated by the plugin.

4. The values that are used in this example are:
a. Project Name: SDcard
b. Build Target: 2.3.3
c. Application Name: SDcard
d. Package Name: com. SDcard.example
e. Create Activity: SDcard

5. Click on Finish.

Coding the Application

1. Open AndroidMainfest.xml which is located in res->values-> AndroidMainfest.xml. This file will hold all of the text that our layout will use.
2. Click on the AndroidMainfest.xml at the bottom to bring up the raw xml file.

Editing the the java code

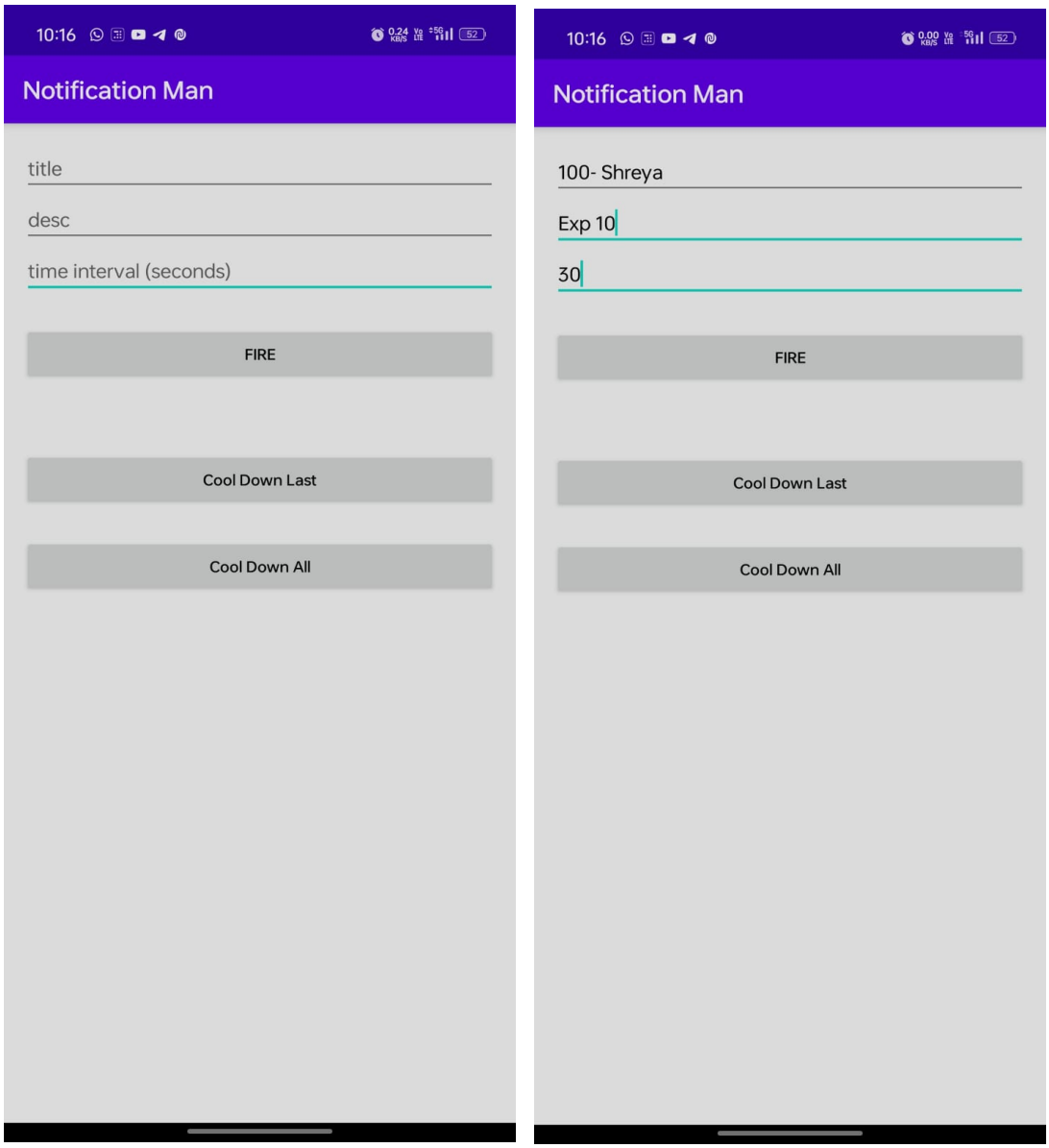
1. Open SampleApp.java from the left hand side.
2. Save the files.

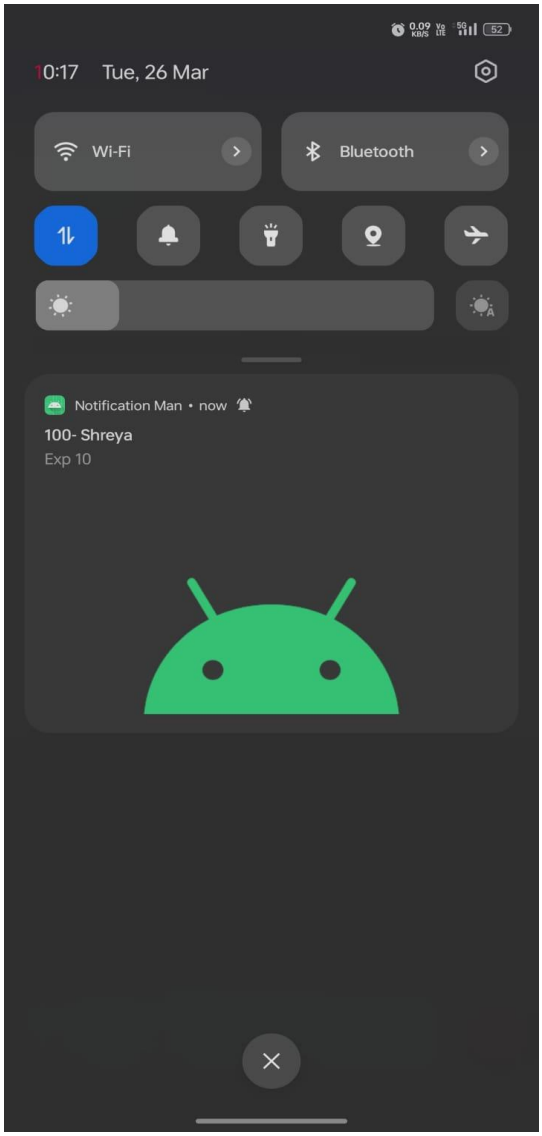
Running the Application

l. Click on the green circle with the white arrow.
2. Choose the AVD that we created in a previous step.
3. The android AVD will load and the program will run

GITHUB LINK: https://github.com/shreyaucoe/Mobile-Computing

OUTPUT:





CONCLUSION:

Thus the application that creates an alert upon receiving a message is executed successfully by using Android Studio.