Experiment No. 11

Aim: To implement notification application in Java and Android.

Requirements: Compatible version of Java, Android Studio and Windows (Supports System Tray).

Theory:

Notification

A notification is a message that Android displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app. Users can tap the notification to open your app or take an action directly from the notification.

Notification anatomy

The design of a notification is determined by system templates—your app simply defines the contents for each portion of the template. Some details of the notification appear only in the expanded view.

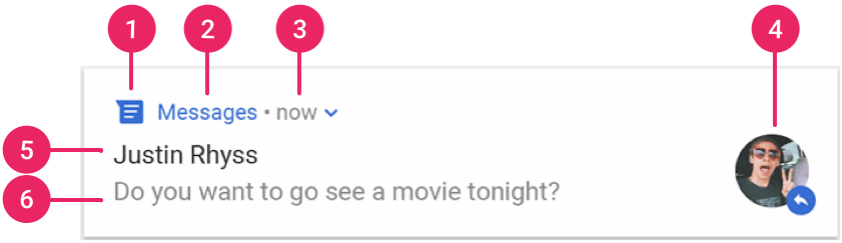


Figure 7. A notification with basic details

The most common parts of a notification are indicated in figure 7 as follows:

1. Small icon: This is required and set with setSmallIcon().
2. App name: This is provided by the system.

3. Time stamp: This is provided by the system but you can override with setWhen() or hide it with setShowWhen(false).

4. Large icon: This is optional (usually used only for contact photos; do not use it for your app icon) and set with setLargeIcon().

5. Title: This is optional and set with setContentTitle().

6. Text: This is optional and set with setContentText().

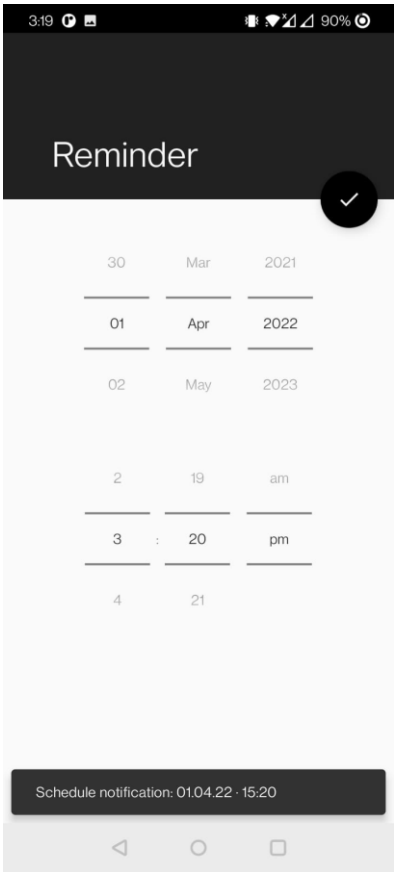
Android Implementation

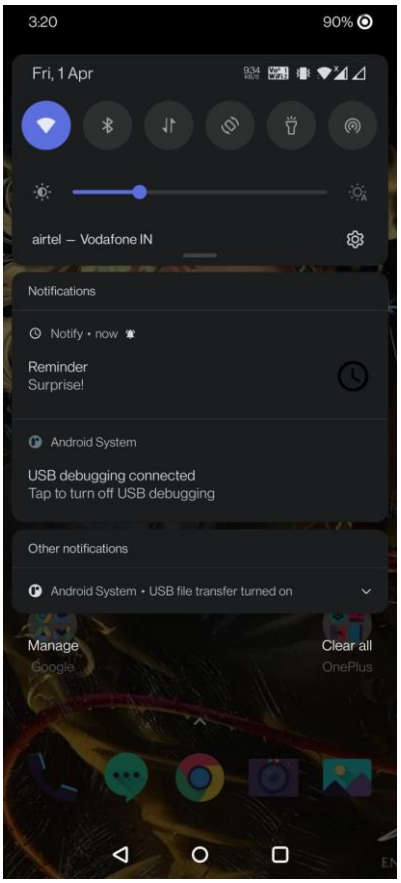
Code:

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        userInterface()
    }

    private fun userInterface() {
        setSupportActionBar(toolbar)

        val titleNotification = getString(R.string.notification_title)
        collapsing_toolbar_l.title = titleNotification

        done_fab.setOnClickListener {
            val customCalendar = Calendar.getInstance()
            customCalendar.set(
                date_p.year, date_p.month, date_p.dayOfMonth, time_p.hour, time_p.minute, 0
            )
            val customTime = customCalendar.timeInMillis
            val currentTime = currentTimeMillis()
            if (customTime > currentTime) {
                val data = Data.Builder().putInt(NOTIFICATION_ID, 0).build()
                val delay = customTime - currentTime
                scheduleNotification(delay, data)

                val titleNotificationSchedule = getString(R.string.notification_schedule_title)
                val patternNotificationSchedule = getString(R.string.notification_schedule_pattern)
                make(
```

```
                coordinator_l,
                titleNotificationSchedule + SimpleDateFormat(
                    patternNotificationSchedule, getDefault()
                ).format(customCalendar.time).toString(),
                LENGTH_LONG
            ).show()
        } else {
            val errorNotificationSchedule = getString(R.string.notification_schedule_error)
            make(coordinator_l, errorNotificationSchedule, LENGTH_LONG).show()
        }
    }
}

private fun scheduleNotification(delay: Long, data: Data) {
    val notificationWork = OneTimeWorkRequest.Builder(NotifyWork::class.java)
        .setInitialDelay(delay, MILLISECONDS).setInputData(data).build()

    val instanceWorkManager = WorkManager.getInstance(this)
    instanceWorkManager.beginUniqueWork(NOTIFICATION_WORK,          REPLACE,
notificationWork).enqueue()
}
}
```

Output:



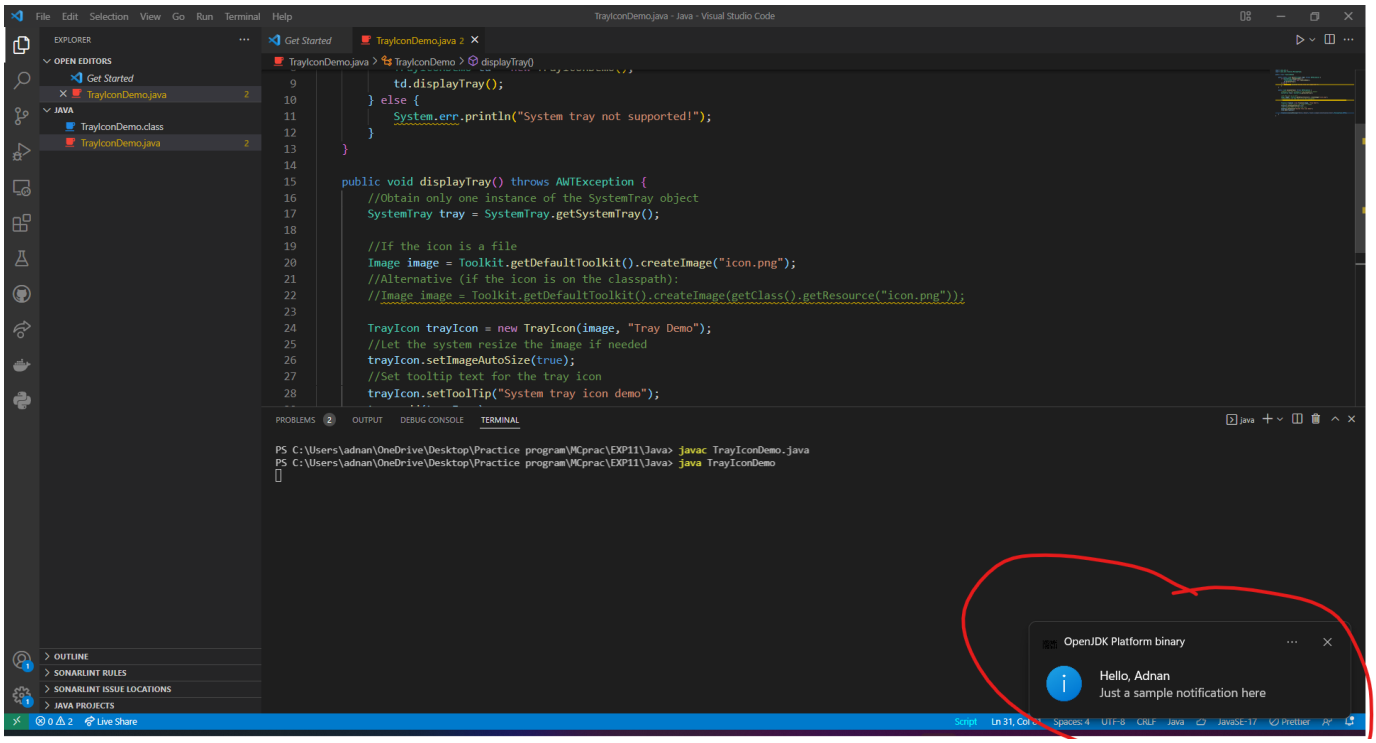


Java Implementation

Code:

```
import java.awt.*;
import java.awt.TrayIcon.MessageType;

public class TrayIconDemo{

   public static void main(String[] args) throws AWTException {
      if (SystemTray.isSupported()) {
         TrayIconDemo td = new TrayIconDemo();
         td.displayTray();
      } else {
         System.err.println("System tray not supported!");
      }
   }

   public void displayTray() throws AWTException {
      //Obtain only one instance of the SystemTray object
      SystemTray tray = SystemTray.getSystemTray();

      //If the icon is a file
      Image image = Toolkit.getDefaultToolkit().createImage("icon.png");
      //Alternative (if the icon is on the classpath):
      //Image image = Toolkit.getDefaultToolkit().createImage(getClass().getResource("icon.png"));

      TrayIcon trayIcon = new TrayIcon(image, "Tray Demo");
      //Let the system resize the image if needed
      trayIcon.setImageAutoSize(true);
      //Set tooltip text for the tray icon
      trayIcon.setToolTip("System tray icon demo");
      tray.add(trayIcon);

      trayIcon.displayMessage("Hello, Adnan", "Just a sample notification here", MessageType.INFO);
   }
}
```

Output:



Conclusion: We have successfully implemented Notification application in Java and Android Studio.