

ICP 14

Notification, Calendar and Publishing apps

ICP Group 4

Name: Sailaja Narra

Email: sntnn@umsystem.edu

ICP Report:

<https://drive.google.com/file/d/1t4jtiJon8vXPGM0c9ml6lj61l4y2JvPB/view?usp=sharing>

ICP Video:

<https://umsystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8527df04-e4af-4245-befb-adfc0009ecbf>

GitHub (Sourcecode):

<https://github.com/UMKC-APL-WebMobileProgramming/ICP14-sailajanarra/tree/main/Source>

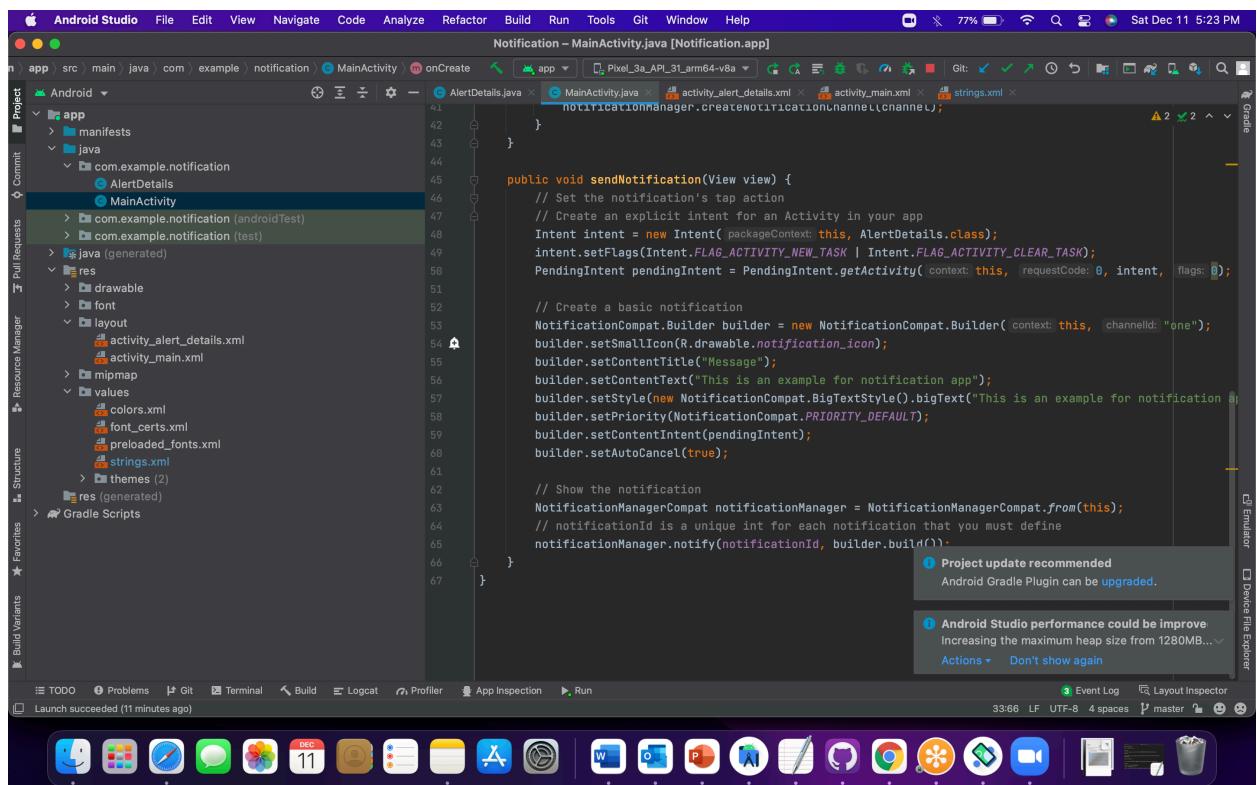
My Partner

- Partner: Venkata Mahesh Mokkapati
- Email: ymzwn@umsystem.edu
- Repo: <https://github.com/UMKC-APL-WebMobileProgramming/ICP14-Mahesh68>

Task:

1. Notification Event

This is the main activity code for this notification event



The screenshot shows the Android Studio interface with the project 'Notification' open. The main window displays the 'MainActivity.java' file under the 'src/main/java/com/example/notification' package. The code implements a 'sendNotification' method that creates a basic notification using the NotificationCompat.Builder class. The notification is set to show a small icon, a content title, and a content text. It is also set to auto-cancel. The 'Project' tool window on the left shows the app's structure, including Java files like 'AlertDetails.java' and 'MainActivity.java', and resource files like 'activity_main.xml' and 'activity_alert_details.xml'. The bottom status bar indicates a successful launch.

```
public void sendNotification(View view) {
    // Set the notification's tap action
    // Create an explicit intent for an Activity in your app
    Intent intent = new Intent(getApplicationContext(), AlertDetails.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    PendingIntent pendingIntent = PendingIntent.getActivity(context, 0, intent, 0);

    // Create a basic notification
    NotificationCompat.Builder builder = new NotificationCompat.Builder(context, "one");
    builder.setSmallIcon(R.drawable.notification_icon);
    builder.setContentTitle("Message");
    builder.setContentText("This is an example for notification app");
    builder.setStyle(new NotificationCompat.BigTextStyle().bigText("This is an example for notification app"));
    builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);
    builder.setContentIntent(pendingIntent);
    builder.setAutoCancel(true);

    // Show the notification
    NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
    // notificationId is a unique int for each notification that you must define
    notificationManager.notify(notificationId, builder.build());
}
```

Basic notification setting configuration

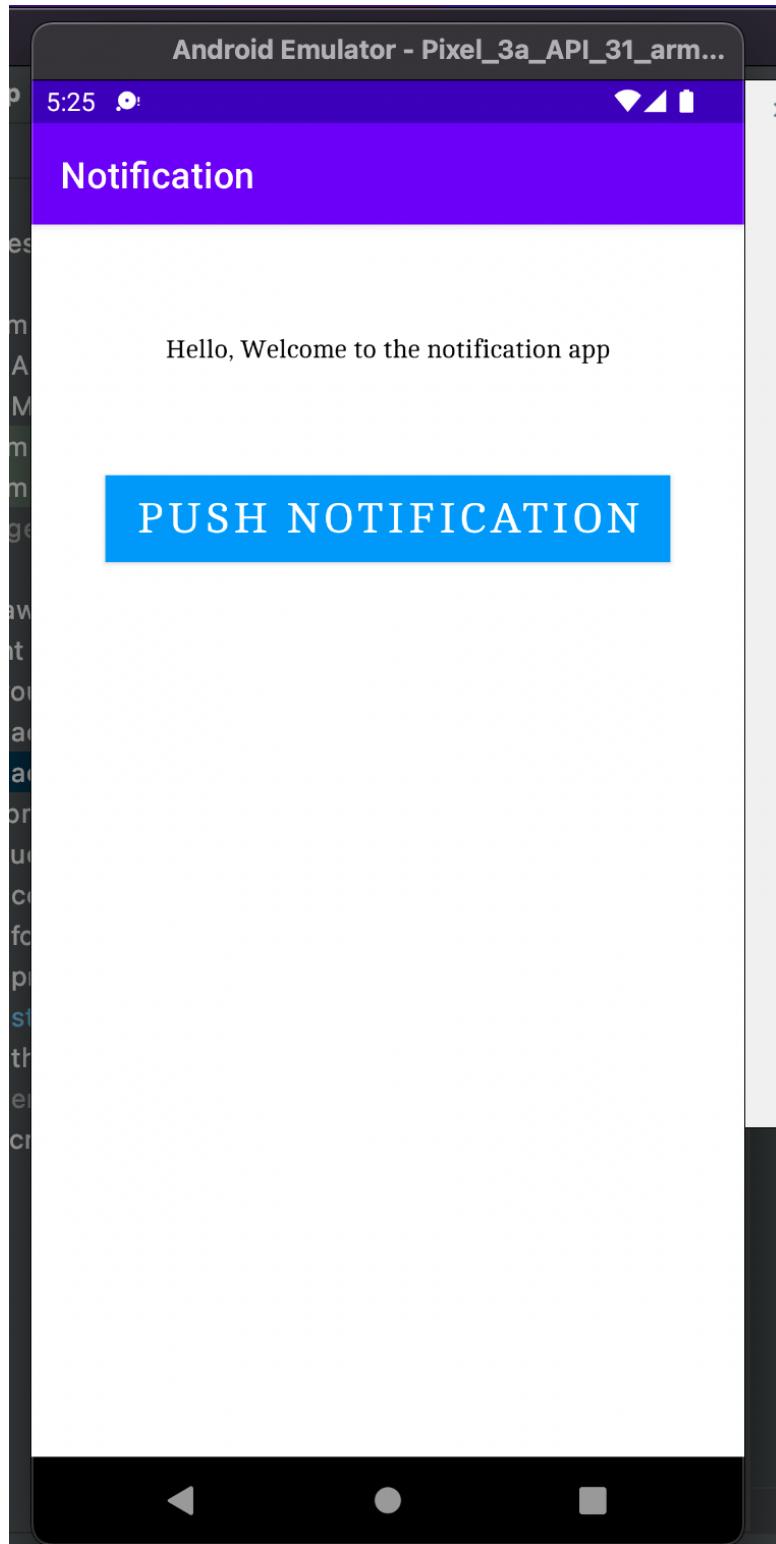
```
// Create a basic notification
NotificationCompat.Builder builder = new NotificationCompat.Builder(context: this, channelId: "or
builder.setSmallIcon(R.drawable.notification_icon);
builder.setContentTitle("Message");
builder.setContentText("This is an example for notification app");
builder.setStyle(new NotificationCompat.BigTextStyle().bigText("This is an example for notifi
builder.setPriority(NotificationCompat.PRIORITY_DEFAULT);
builder.setContentIntent(pendingIntent);
builder.setAutoCancel(true);
```

On Successful launch of the application:

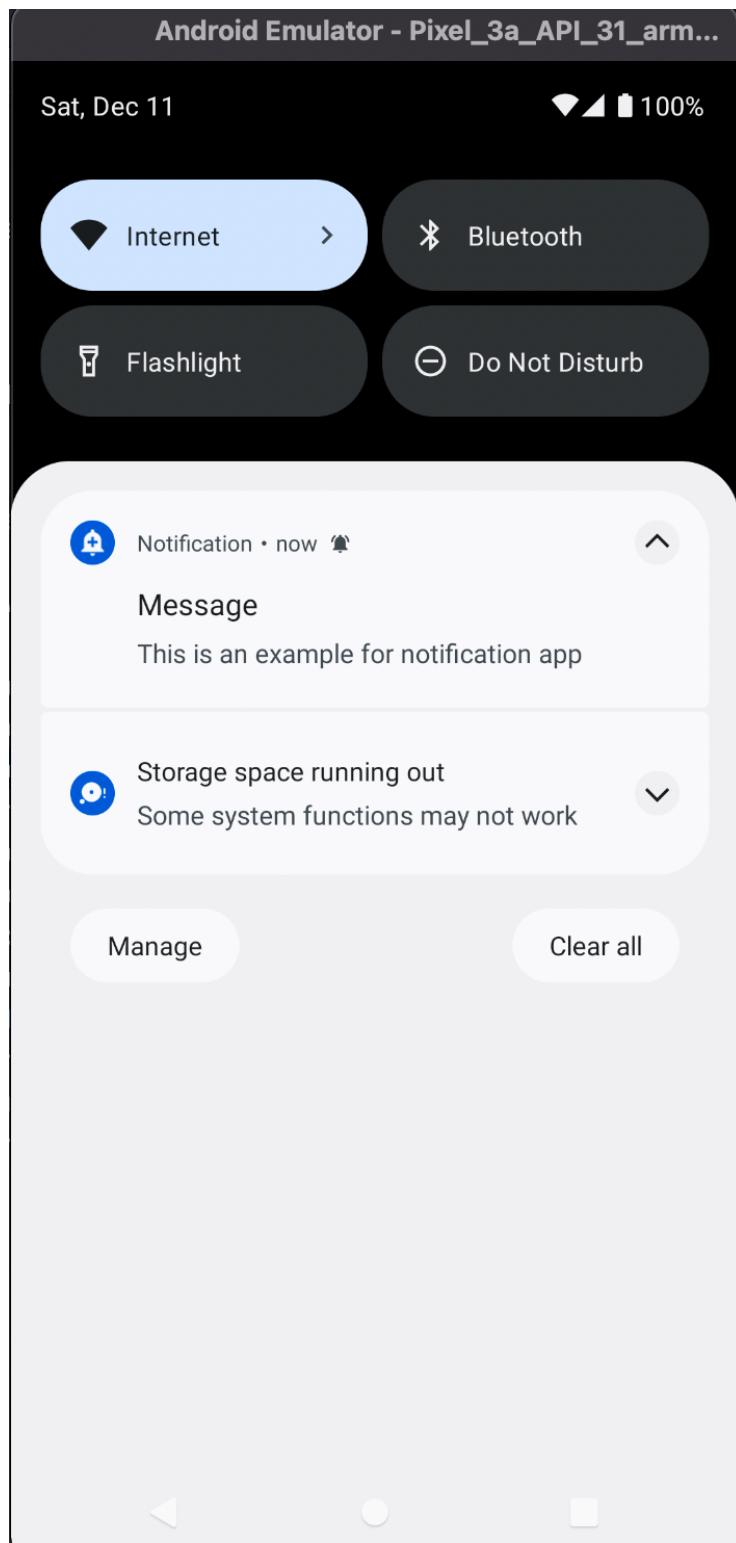
Below is the home activity landing page

This notification API is lot more useful in the case of service workers in the web and also creating an powerful progressive web applications.

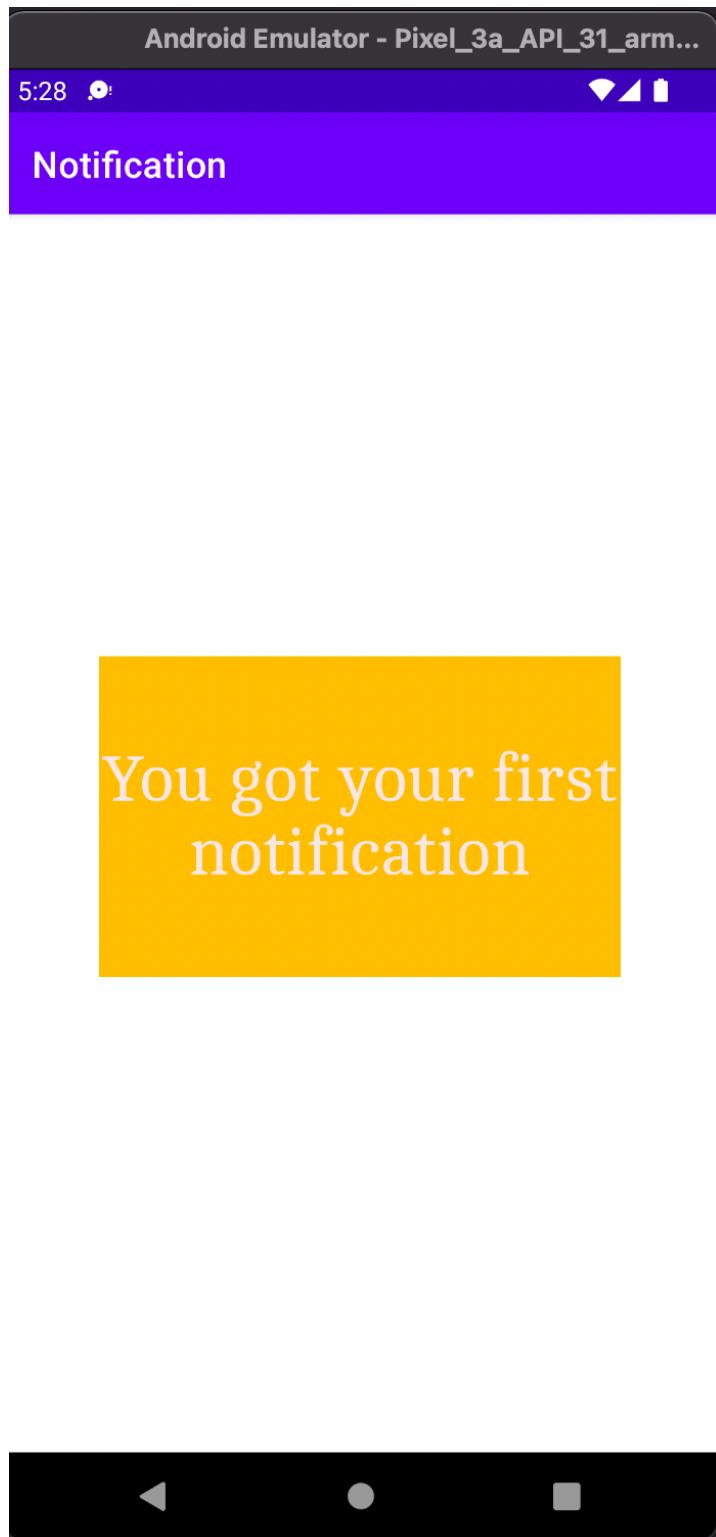
Users can also have an nice and user friendly UI with the notifications that will be sitting under quick settings.



On click event listener of this button sendNotification method will be triggered and then notify method will be triggered



On Click of that notification below alert layout will be opened



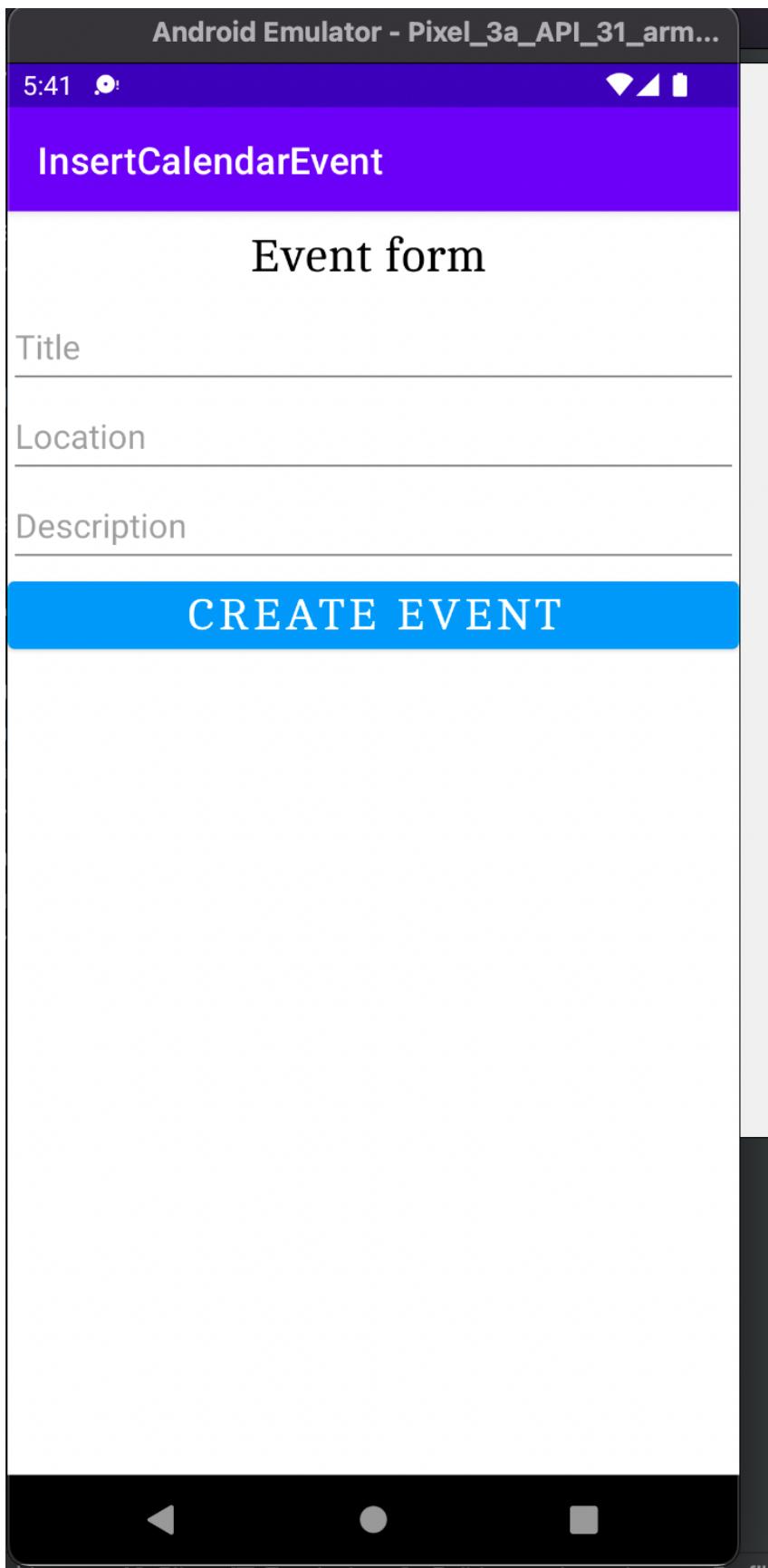
2. Display and inserting event

Creating an event with the following kotlin code

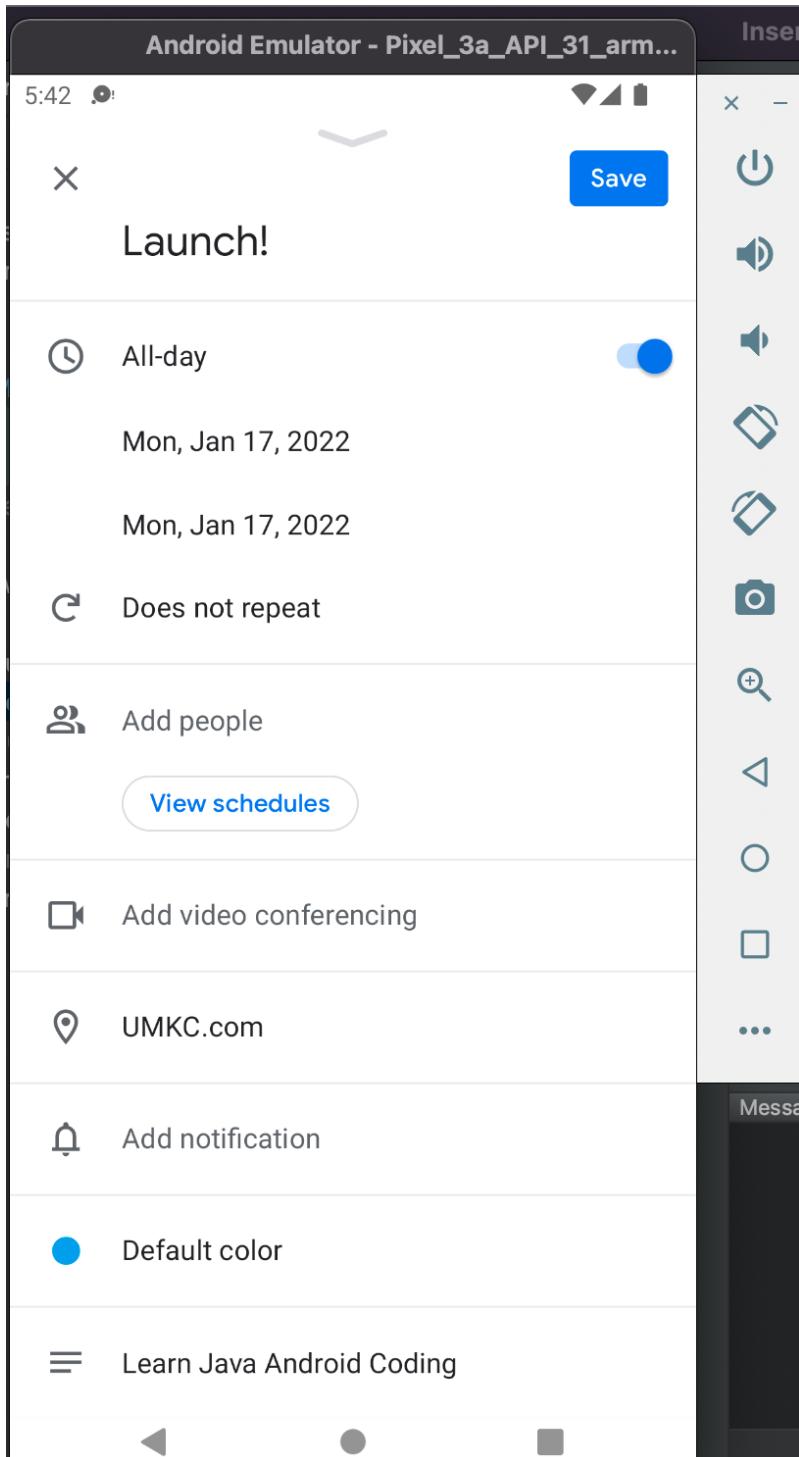
```
public void insert() {
    Intent intent = new Intent(Intent.ACTION_INSERT,
        CalendarContract.Events.CONTENT_URI);
    // Add the calendar event details
    intent.putExtra(CalendarContract.Events.TITLE, value: "Launch!");
    intent.putExtra(CalendarContract.Events.DESCRIPTION,
        value: "Learn Java Android Coding");
    intent.putExtra(CalendarContract.Events.EVENT_LOCATION,
        value: "UMKC.com");
    Calendar startTime = Calendar.getInstance();
    startTime.set( year: 2021, month: 12, date: 17);
    intent.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME,
        startTime.getTimeInMillis());
    intent.putExtra(CalendarContract.EXTRA_EVENT_ALL_DAY, value: true);
    // Use the Calendar app to add the new event.

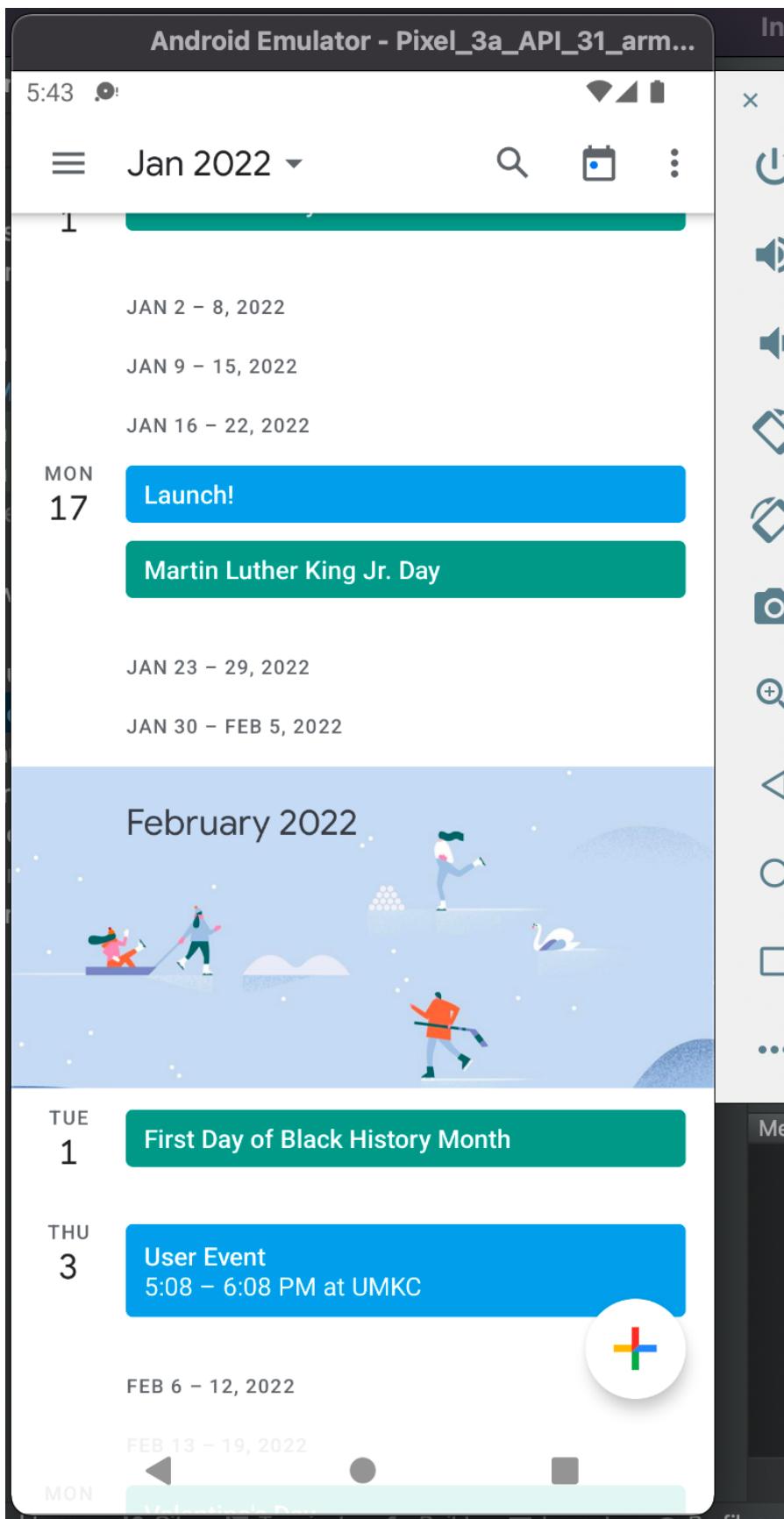
    startActivity(intent);
}
```

Home Activity:

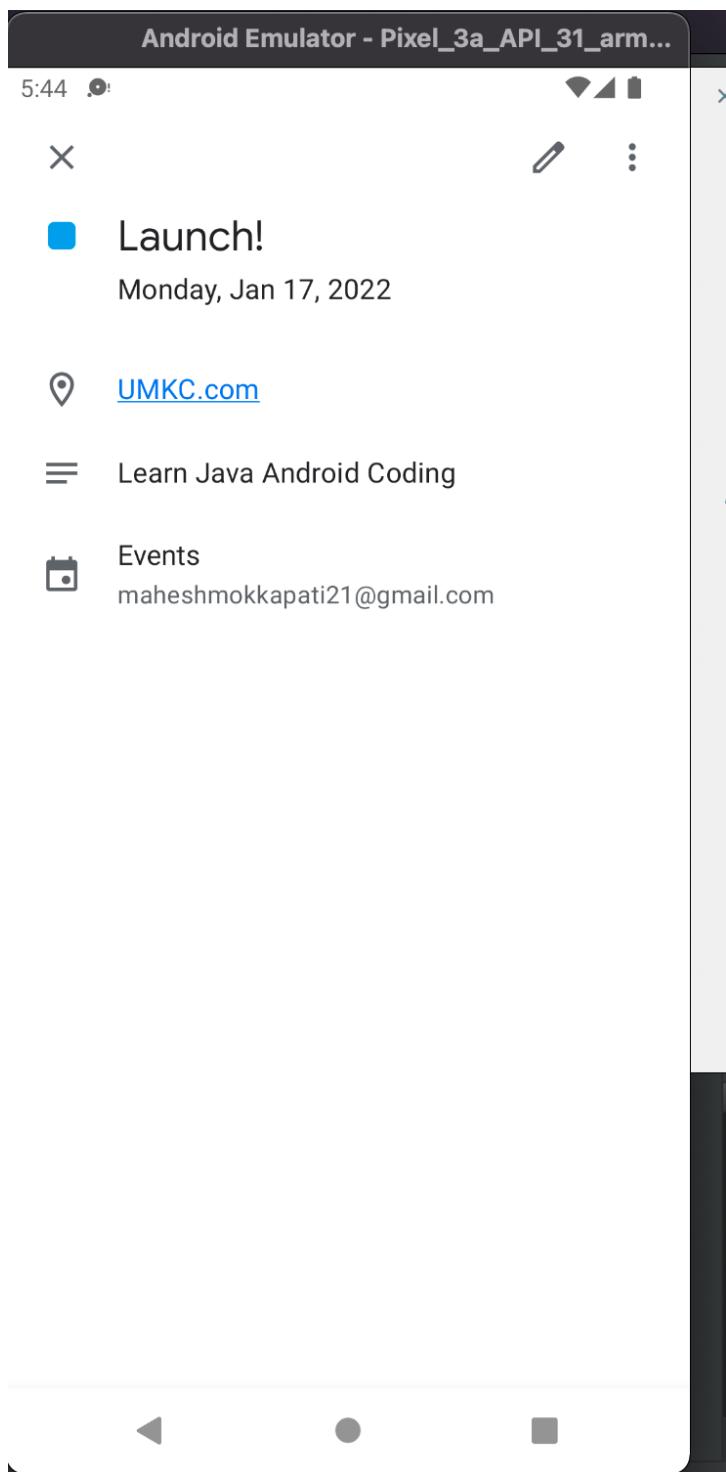


On Successful launch of the event, calendar application will be opened with sync from existing google calendars





Displaying the calendar event



Android Emulator - Pixel_3a_API_31_arm...

4:59

Save

WEB!

All-day

Does not repeat

Add people

[View schedules](#)

UMKC

The day before at 11:30 PM

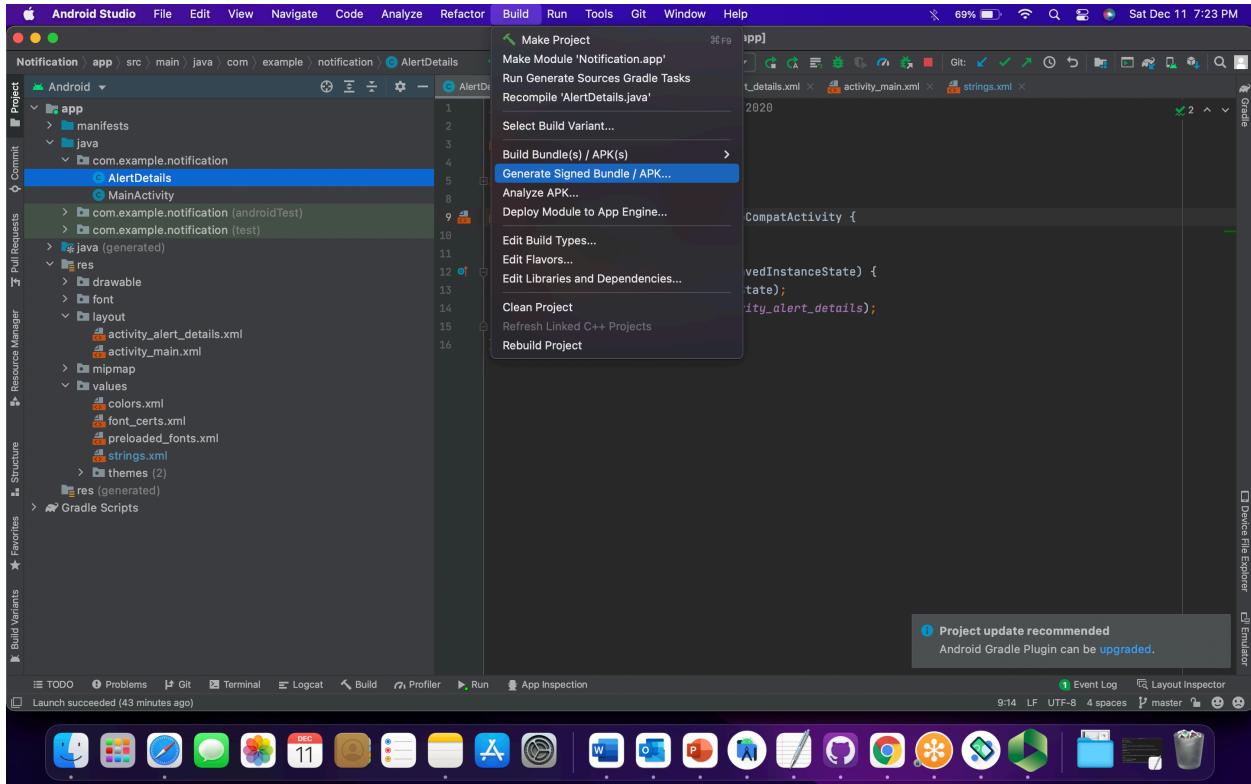
Add notification

Add video conferencing

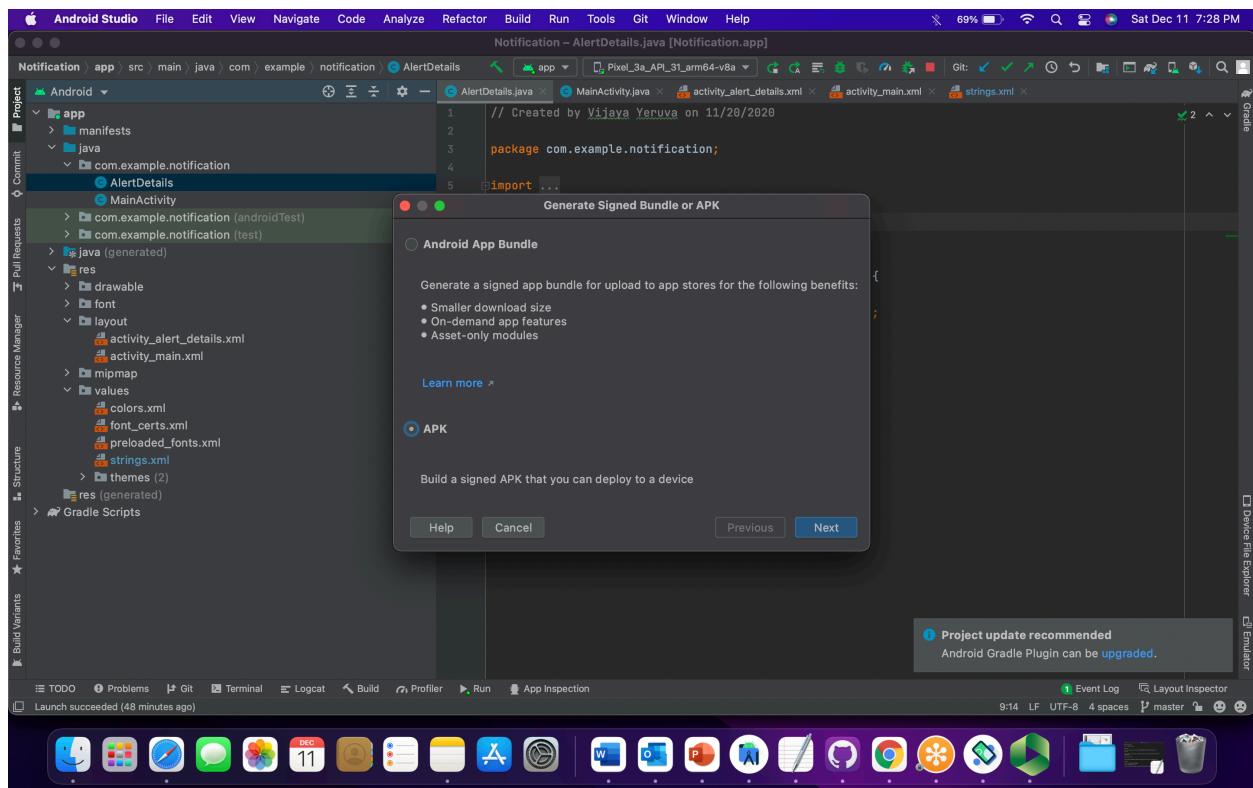
Default color

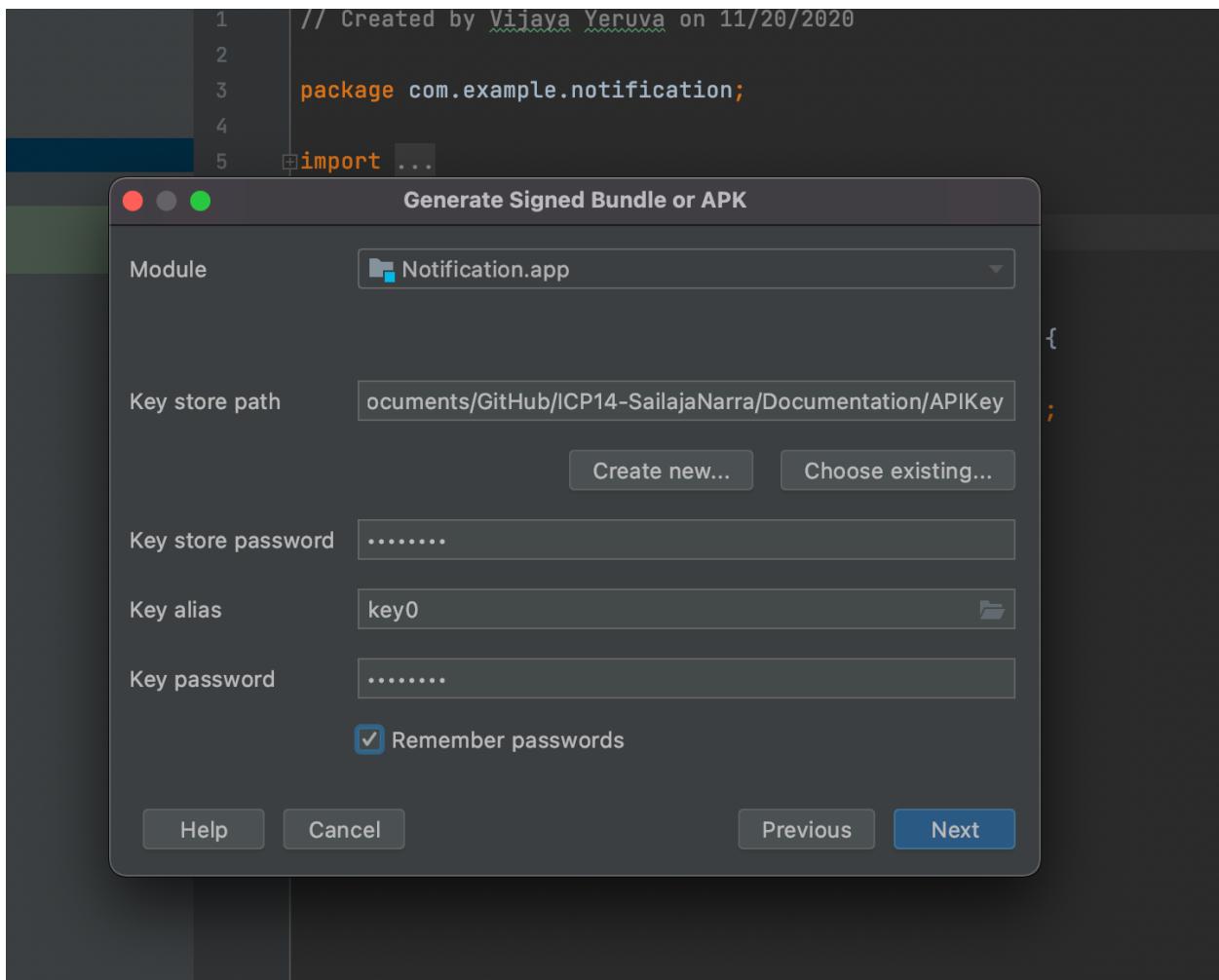
Web and mobile programming

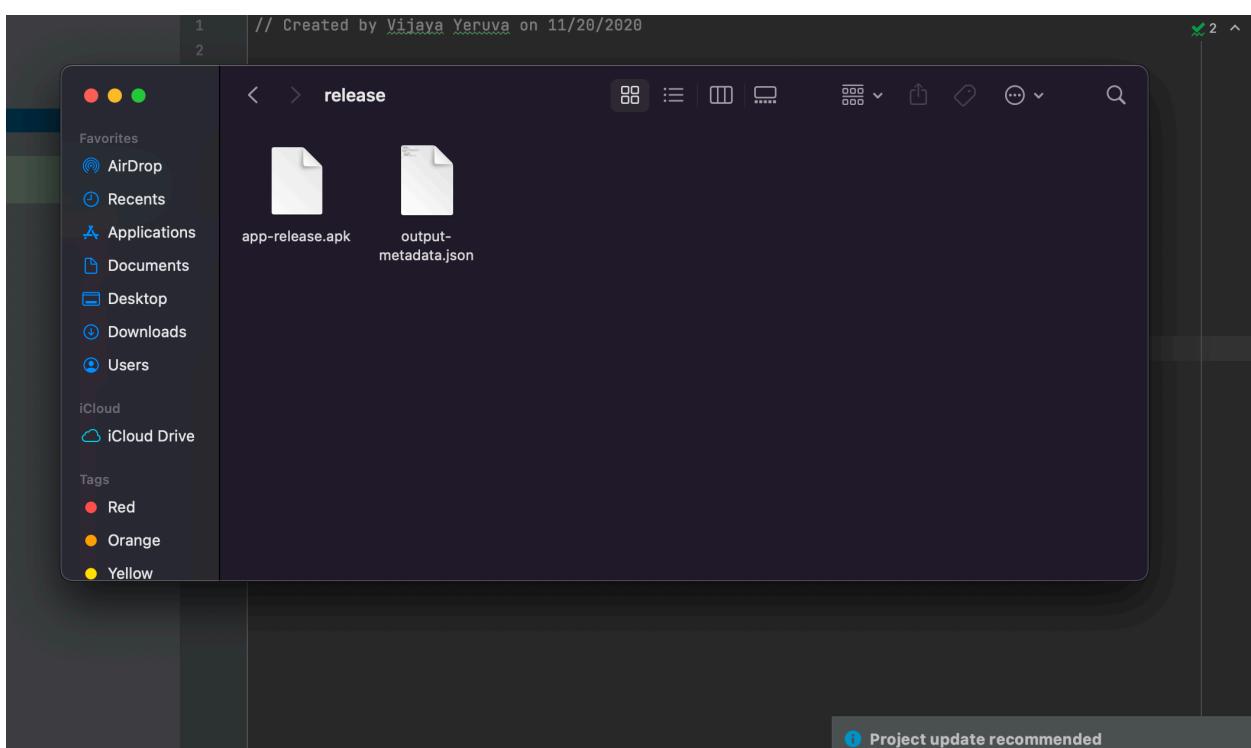
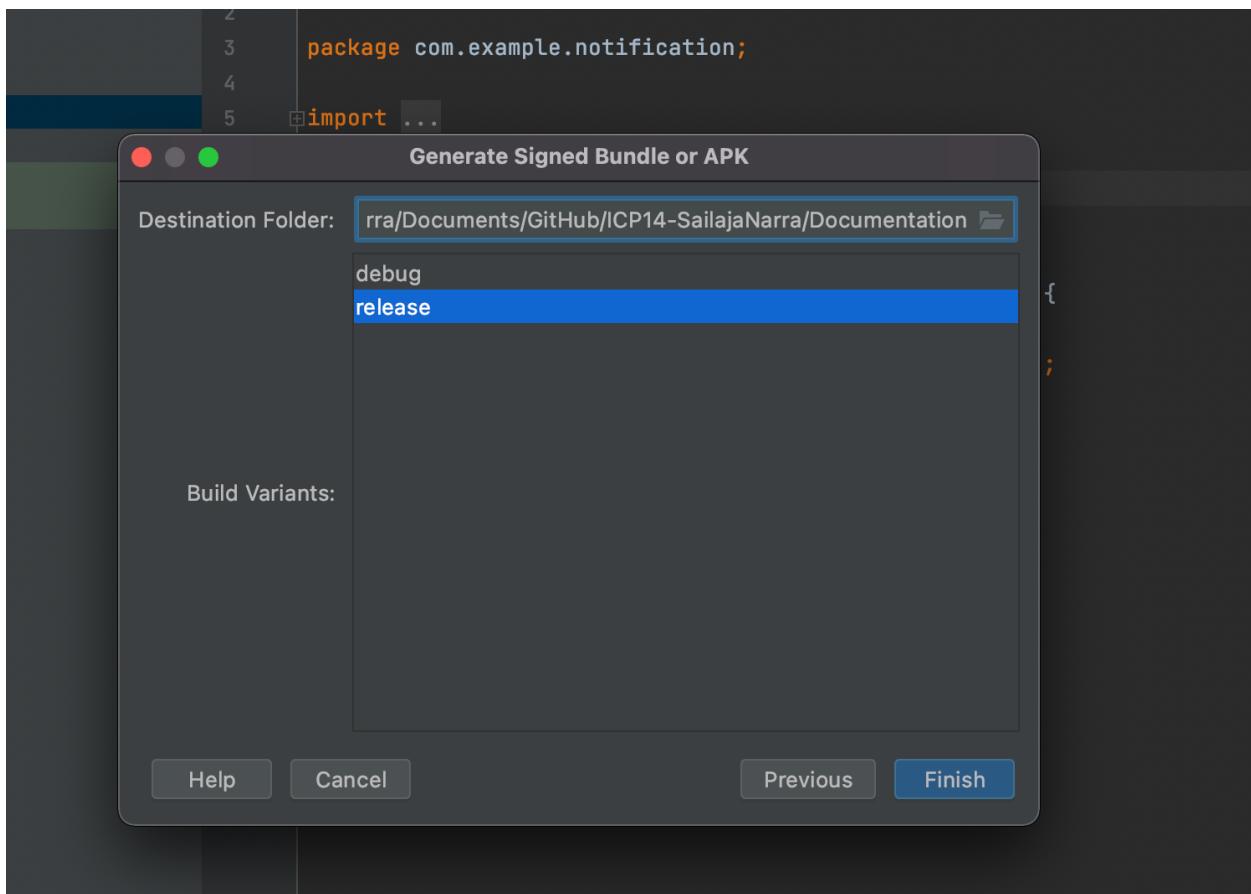
Generating the APK:



Under the build there is an option to generate the SDK/APK file. With this APK users can directly install the application into their devices with the compatibility settings turned off.







With this approach APK has been created successfully with the release version.