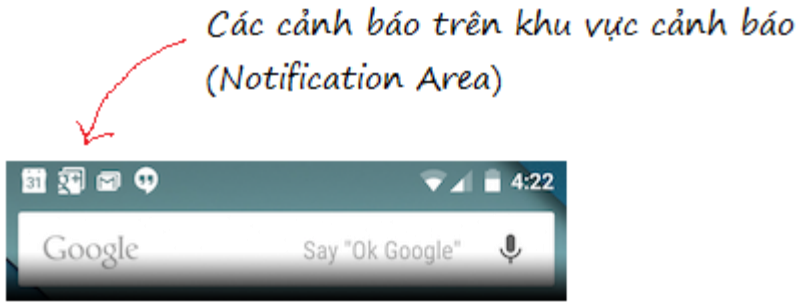
ANDROID PROGRAMMING
LESSON 11

Notifications

Nội dung

- Notifications
- Thành phần của Notification
- Ví dụ 1
- Tạo một chanel và gán giá trị importance
- Ví dụ 2
- Ví dụ 3: dùng AlarmManager và PendingIntent
- Ví dụ 4: Bài lập lịch (Note)

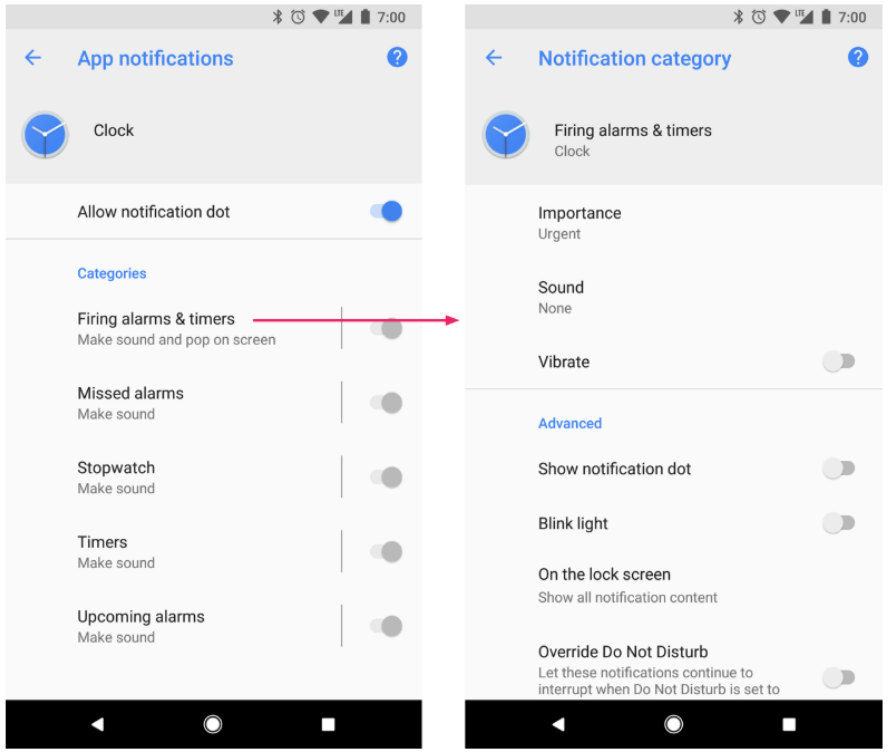
Notifications



- Notification trong Android là một user interface element, nó được hiện thị ở bên ngoài user interface của ứng dụng để thông báo cho người dùng những thông tin cần thiết. Người dùng có thể xem được những thông báo này ở màn hình khóa (đối với Android Lollipop) hoặc từ Notification Drawer.

Notification Channel

Bắt đầu từ phiên bản 8.0 (Oreo), Android bắt đầu nhóm các thông báo (notification) vào các kênh (Channel) khác nhau. Mỗi kênh sẽ có một hành vi cụ thể, và hành vi này sẽ được áp dụng cho tất cả các thông báo của nó. Mỗi kênh có một ID đại diện cho nó.



Thành phần của Notification

- Small icon: setSmallIcon().
- Timestam: cung cấp bởi hệ thống, nhưng có thể override với setWhen() hoặc không hiển thị với setShowWhen(false).
- Large icon: không bắt buộc phải có, setLargeIcon().
- Title: không bắt buộc, setContentTitle().
- Text: không bắt buộc, setContentText(). Quyền ưu tiên setPriority(). Giá trị priority xác định mức độ được hiển thị trong các tình huống khác nhau trong Android 7.1 hoặc thấp hơn. (Với Android 8.0 hoặc cao hơn, gán giá trị chanel importance)

Notifications

- Để tạo một notification trong Android, sử dụng NotificationCompat.Builder. Một notification được tạo ra phải có ít nhất ba thông tin dưới đây :
  - 1 small icon, sử dụng method setSmallIcon()
  - 1 title, sử dụng method setContentTitle()
  - Nội dung của notificaiton, sử dụng method setContentText()
- Để Notification hiển thị cho người dùng, sử dụng NotificationManagerCompat :
  - Tạo một đối tượng của NotificationManagerCompat
  - Sử dụng method notify() để đẩy thông báo cho người dùng bằng cách truyền vào hai tham số : Id của notification và đối tượng Notification.

Thành phần Notification (Ví dụ 1)

```
private int notificationid=1;
Bitmap bitmap=
BitmapFactory.decodeResource(getResources(),R.mipmap.ic_launcher);
NotificationCompat.Builder builder=new
NotificationCompat.Builder(getApplicationContext())
        .setContentTitle("Tin nhan")
        .setContentText("Noi dung tin nhan")
        .setColor(Color.RED)
        .setDefaults(NotificationCompat.DEFAULT_SOUND)
        .setLargeIcon(bitmap)
        .setSmallIcon(R.drawable.ic_baseline_bus_alert_24);
NotificationManagerCompat managerCompat=
NotificationManagerCompat.from(MainActivity.this);
managerCompat.notify(notificationid,builder.build());
```

Tạo một chanel và gán giá trị importance

- Trước khi tạo ra một notification ở Android 8.0 và cao hơn, phải đăng ký notification channel với hệ thống qua một instance của NotificationChannel để createNotificationChannel().
- Chú ý hàm khởi tạo của NotificationChanel yêu cầu một giá trị importance, giá trị importance này bạn có thể lấy một hằng số từ NotificationManager.

Giá trị importance

Table 1. Channel importance levels

User-visible importance level	Importance (Android 8.0 and higher)	Priority (Android 7.1 and lower)
Urgent Makes a sound and appears as a heads-up notification	`IMPORTANCE_HIGH`	`PRIORITY_HIGH` or `PRIORITY_MAX`
High Makes a sound	`IMPORTANCE_DEFAULT`	`PRIORITY_DEFAULT`
Medium No sound	`IMPORTANCE_LOW`	`PRIORITY_LOW`
Low No sound and does not appear in the status bar	`IMPORTANCE_MIN`	`PRIORITY_MIN`

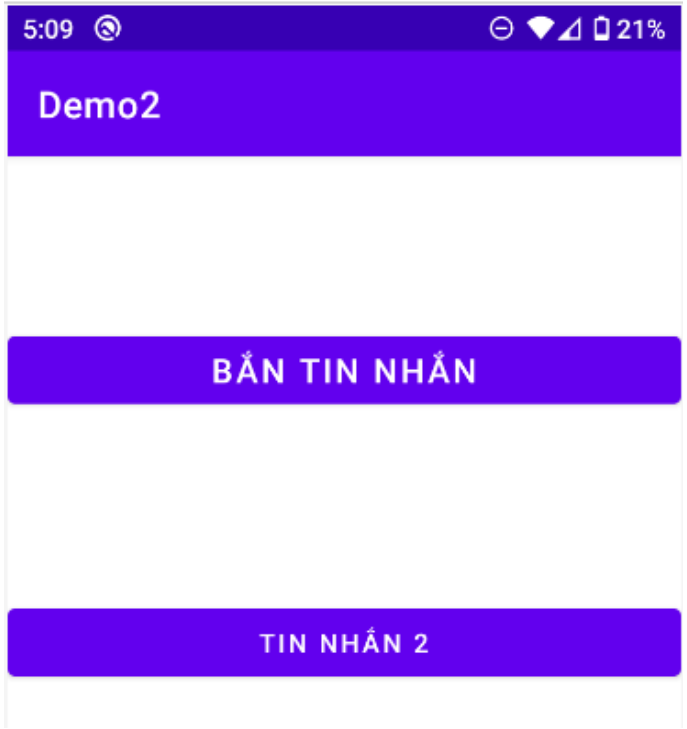
https://developer.android.com/training/notify-user/channels

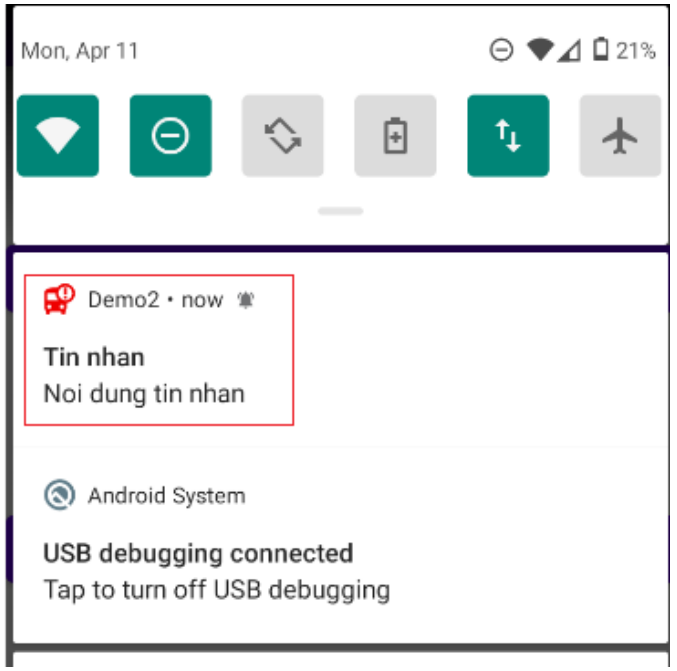
```
private void createNotificationChannel() {
        // Create the NotificationChannel, but only on API 26+ because
        // the NotificationChannel class is new and not in the support library
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            CharSequence name = getString(R.string.channel_name);
            String description = getString(R.string.channel_description);
            int importance = NotificationManager.IMPORTANCE_DEFAULT;
            NotificationChannel channel = new NotificationChannel(CHANNEL_ID,
name, importance);
            channel.setDescription(description);
            NotificationManager notificationManager =
getSystemService(NotificationManager.class);
            notificationManager.createNotificationChannel(channel);
        }
    }
```

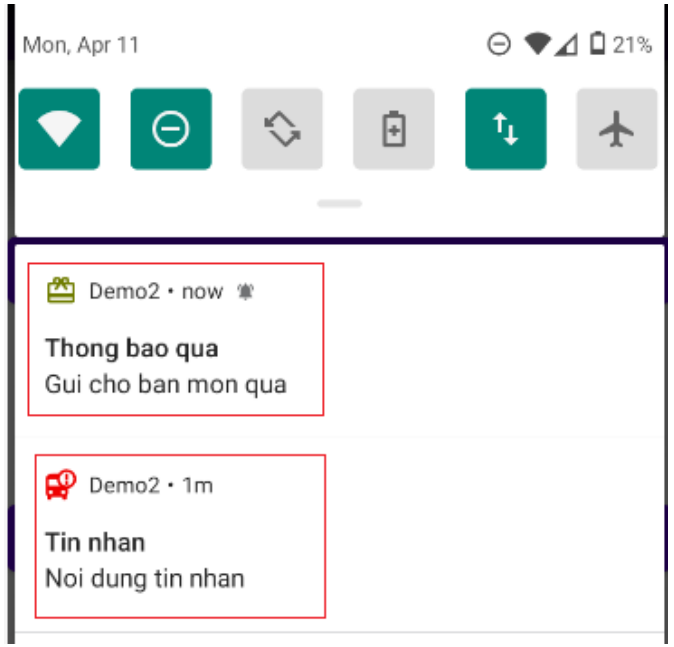
Hiển thị notification

```
Bitmap bitmap=
BitmapFactory.decodeResource(getResources(),R.mipmap.ic_launcher);
        NotificationCompat.Builder builder=new
NotificationCompat.Builder(MainActivity.this,
                MyApplication.CHANNEL_ID)
                .setContentTitle("Tin nhan")
                .setContentText("Noi dung tin nhan")
                .setColor(Color.RED)
                .setDefaults(NotificationCompat.DEFAULT_SOUND)
                .setLargeIcon(bitmap)
                .setSmallIcon(R.drawable.ic_baseline_bus_alert_24)
                .setCategory(NotificationCompat.CATEGORY_ALARM)
                .setAutoCancel(true);
        NotificationManagerCompat
managerCompat=NotificationManagerCompat.from(getApplicationContext());
        managerCompat.notify(notificationid,builder.build());
```

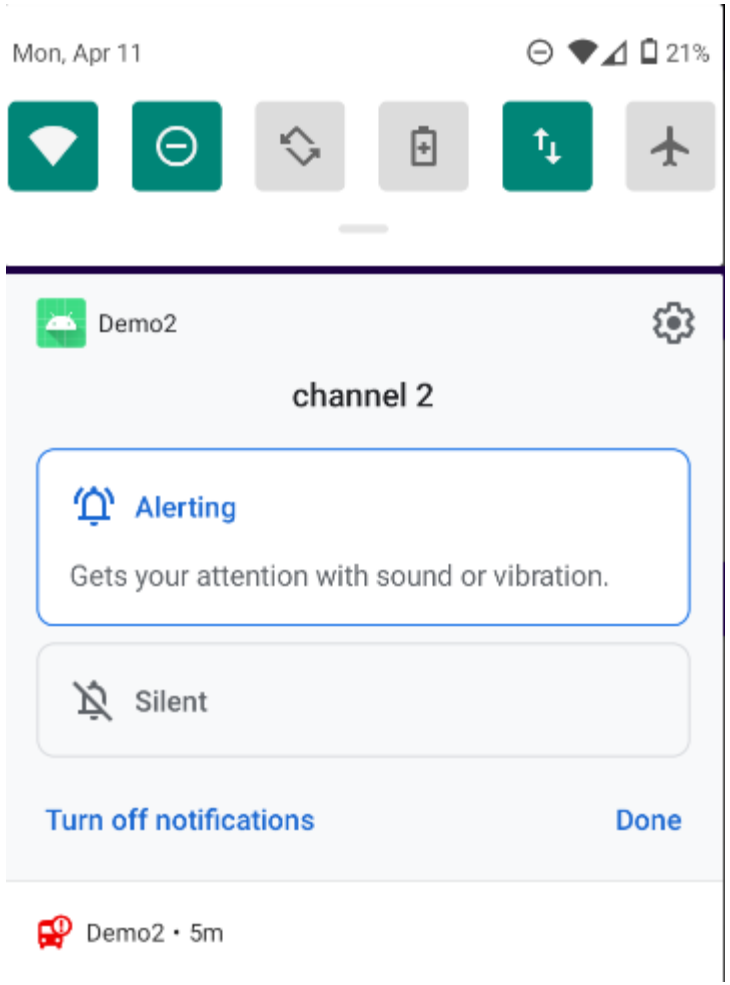
Ví dụ DemoNotification (2)

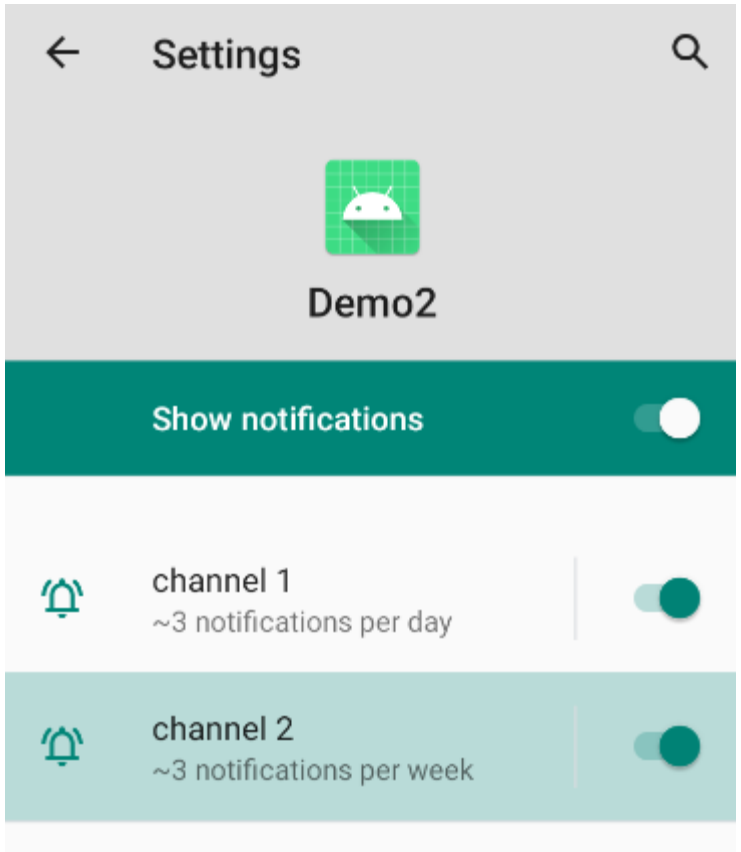






Các channel


Mon, Apr 11
21%
Demo2
channel 2
Alerting
Gets your attention with sound or vibration.
Silent
Turn off notifications
Done
Demo2 • 5m


Settings
Demo2
Show notifications
channel 1
~3 notifications per day
channel 2
~3 notifications per week

AlarmManager và PendingIntent

- AlarmManager là một cầu nối giữa ứng dụng và alarm service của hệ thống Android. Nó có thể gửi một bản tin broadcast tới ứng dụng ở thời điểm đã được lên lịch trước đó. Sau đó ứng dụng có thể thực hiện bất kỳ tác vụ nào cần thiết ở thời điểm đó.
- AlarmManager, chúng ta nghĩ ngay tới ứng dụng báo thức. Ta đặt một lịch đến đúng 8h sáng thì chuông reo để còn đi làm.
- Cùng với sự trợ giúp của PendingIntent và Intent, một bundle đóng gói các thông tin cần thiết sẽ được gửi cùng broadcast hệ thống tới ứng dụng khi thời gian đặt lịch đến.
- Tuy nhiên, từ phiên bản Android M thì Google đã có sự thay đổi với AlarmManager để hạn chế việc tiêu thụ pin quá mức. Alarm sẽ không còn được ưu tiên bật chính xác vào thời điểm đã đặt, mà tùy thuộc vào tình hình tài nguyên máy thời điểm đó mà thời gian bật alarm sẽ có sai lệch 1 chút.

Thiết lập AlarmManager

- **Tạo BroadcastReceiver để nhận broadcast từ hệ thống**
  - bạn tạo một class kế thừa từ BroadcastReceiver
  - Sau đó đăng ký receiver trong **AndroidManifest.xml**
- **Tạo alarm**
  - alarmManager=(AlarmManager)getSystemService(ALARM_SERVICE);

PendingIntent

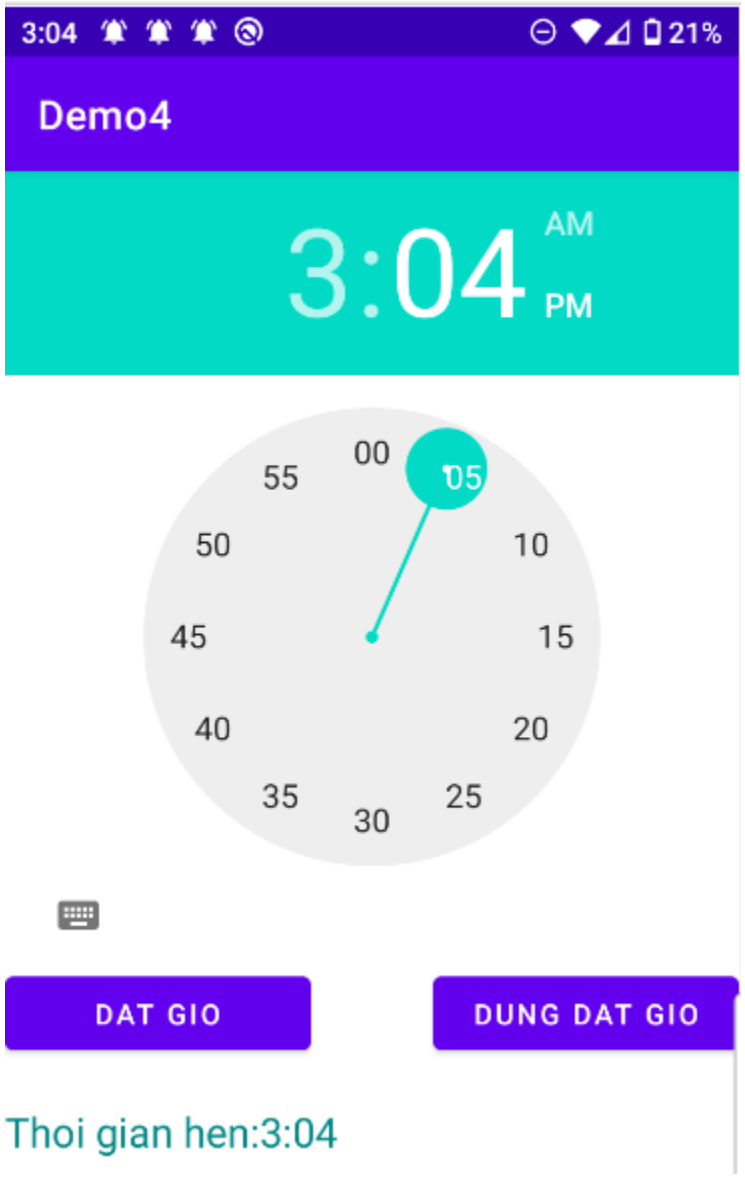
- PendingIntent là 1 Intent đặc biệt, những Intent khác chỉ tồn tại khi được khởi tạo cho đến khi kết thúc nhiệm vụ của mình nhưng PendingIntent lại có thể tồn tại trong suốt ứng dụng và ngay cả khi đã thoát ứng dụng.
- Nó thường được sử dụng trong các trường hợp thực hiện những tác vụ không biết khi nào kết thúc hoặc cần được sử dụng trong suốt ứng dụng bất kỳ Activity nào :send SMS, broadcast/receiver, send message (Skype, Viber app)...

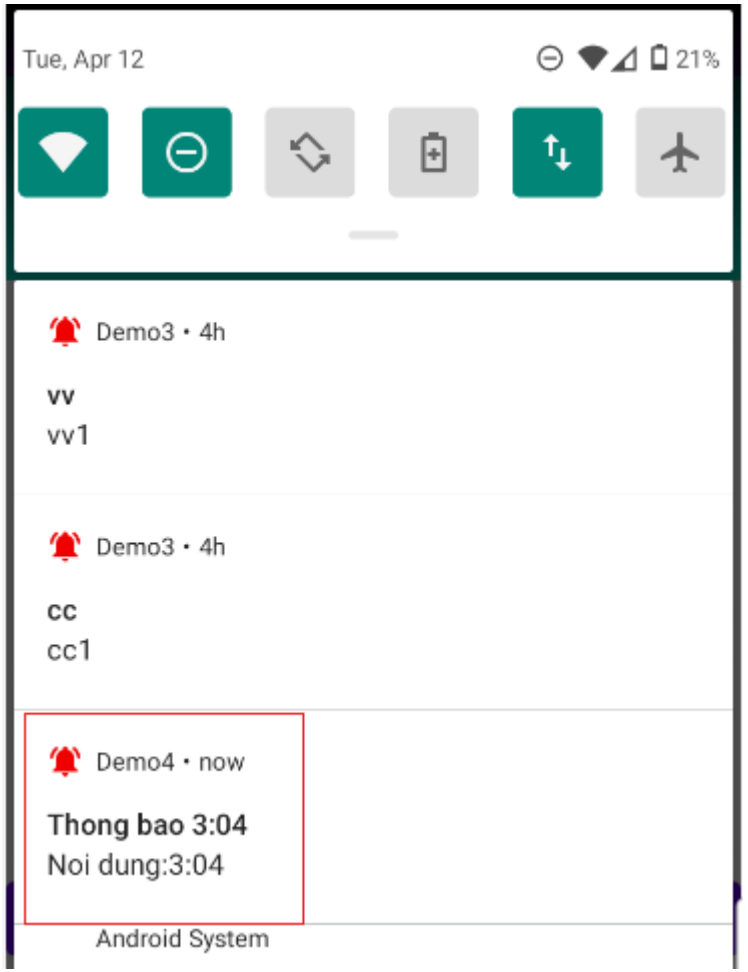
- AlarmManager sẽ gửi một broadcast tới BroadcastReceiver. Theo như tài liệu chính thức của Google có mô tả thì hàm getBroadcast() dùng cho AlarmManager.
- pendingIntent = PendingIntent.getBroadcast(MainActivity.this, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
- alarmManager.set(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(), pendingIntent);

Hủy một AlarmManager

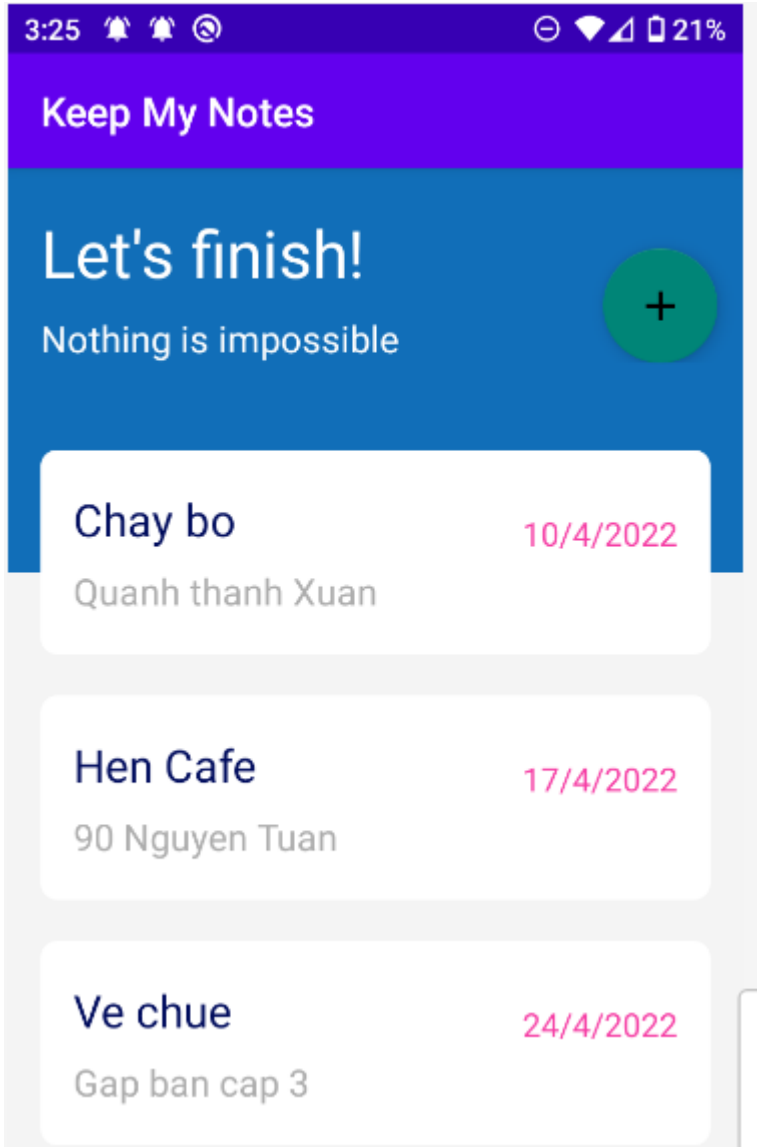
- alarmManager.cancel(onAlarmListener);
- Tuy nhiên, nếu chỉ muốn hủy chính xác một alarm thì cần phải truyền chính xác requestCode mà đã tạo trước đó.

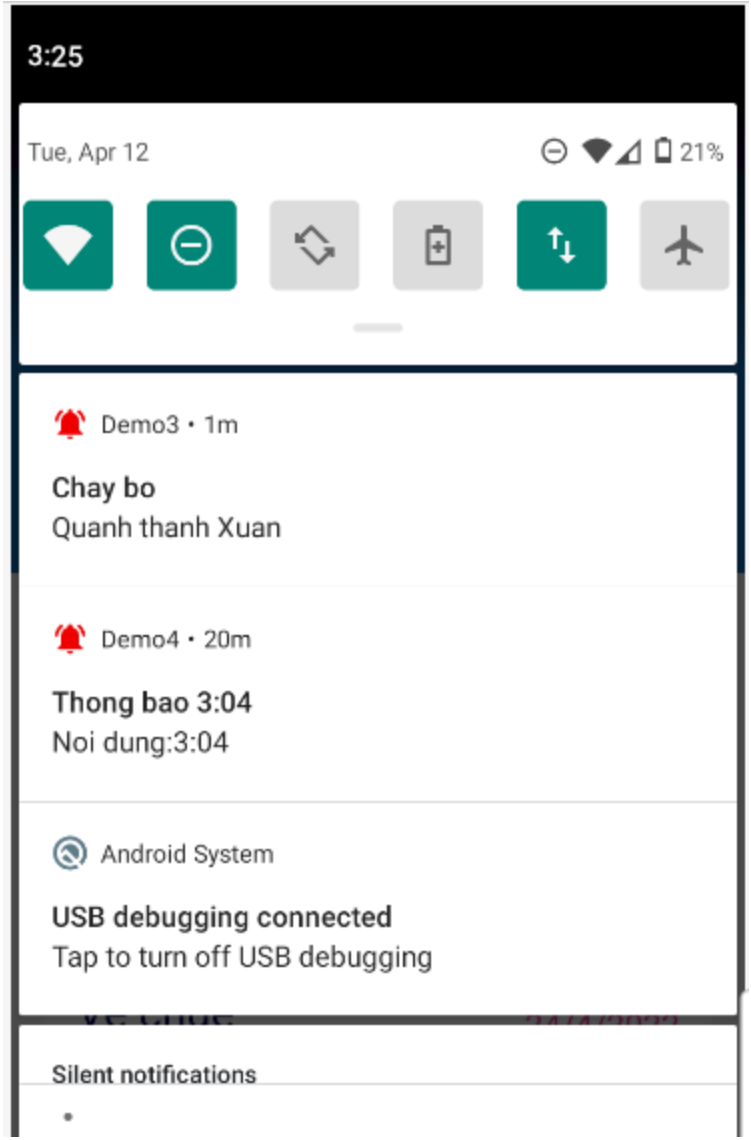
Ví dụ 3


3:04
21%
Demo4
3:04
AM
PM
00
05
10
15
20
25
30
35
40
45
50
55
DAT GIO
DUNG DAT GIO
Thoi gian hen:3:04


Tue, Apr 12
21%
Demo3 • 4h
vv
vv1
Demo3 • 4h
cc
cc1
Demo4 • now
Thong bao 3:04
Noi dung:3:04
Android System

Ví dụ 4:bài toán lập lịch





- End of Lesson 11

Thank you!