Platform-Based Development: Android Programming – Permissions and Notifications

BS UNI studies, Spring 2019/2020

Dr Veljko Pejović
Veljko.Pejovic@fri.uni-lj.si

University *of Ljubljana*
Faculty *of Computer and*
Information Science

Android Security

- Applications are sandboxed
  - Each app has its user ID and group ID that is unique for the device
  - Allocated resources for the user ID
- Only the most basic functionalities are available to an application, other functionalities have to be explicitly asked for

Permissions

- An app requests permissions to access:
    - User data (e.g. contacts)
    - Some cost-sensitive APIs (e.g. send SMS)
    - Some system resources (e.g. camera)
- Permissions are declared in AndroidManifest.xml
    - Predefined Strings from Manifest.permission
    - <user-permission> in Manifest indicates a request:

```
<uses-permission android:name=
"android.permission.ACCESS_FINE_LOCATION"/>
```

University *of Ljubljana*
Faculty *of Computer and*
Information Science

Permissions Types

- Normal permissions
  - FLASHLIGHT, VIBRATE, BLUETOOTH, etc.
  - The OS will automatically grant these permissions to your app

  http://developer.android.com/guide/topics/security/normal-permissions.html
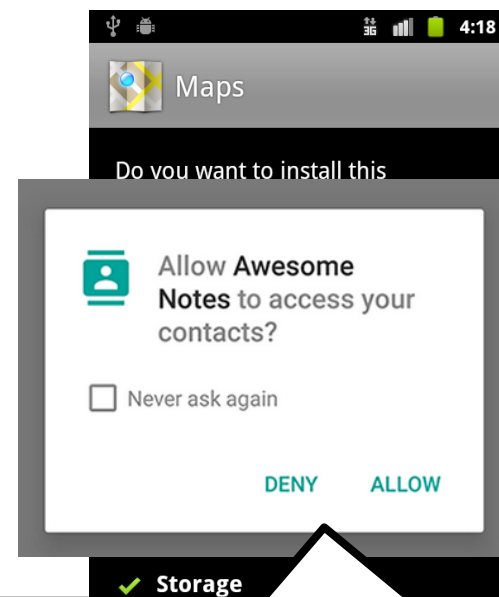
- Dangerous permissions
  - READ_CONTACTS, SEND_SMS, ACCESS_COARSE_LOCATION, WRITE_EXTERNAL_CONTACTS, etc.
  - The OS will explicitly ask the user to grant access for dangerous permissions

Handling Permissions

- Prior to Marshmallow (API 23):
  - Users must accept permissions before the app starts

- Marshmallow (API 23):
  - Permissions must be granted at the runtime, when the functionality is about to be used
  - Ask straight away, educate up front, ask in context, educate in context

    See a short explanation by Google devs:
    www.youtube.com/watch?v=iZqDdvhTZj0

Note that users can disable permissions at a later point

Custom Permissions

- Define your own permission if the app performs a privileged/dangerous function, and you don't want just any app to be able to launch yours

- In AndroidManifest.XML under <permission>

- Example:

```
<permission android:name="com.testpackage.mypermission"
            android:label="my_permission"
            android:protectionLevel="dangerous"/>
```

- Any app that wants to launch yours must request "com.testpackage.mypermission"

Component Permissions

- Permissions can be defined for individual components restricting which other component can call them
    - Activities
        - Which components can call startActivity()
    - Services
        - Which components can start or bind to a Service
    - BroadcastReceivers
        - Which components can receive and send broadcasts
    - ContentProviders
        - Which components can read and write content provider data

Permissions Design

- Do not request permissions unless you really need them
  - E.g. do you need to write data to an external storage (file) or can you keep the information in SharedPrefs?
- Show immediate benefit of granting a permission
  - E.g. your ToDo list app requires location info – show the user how she can make location-based reminders
- Use Intents to call other apps in case you don't need to handle the functionality within your app:
  - E.g. call a camera app, rather than requesting the camera permission for your app
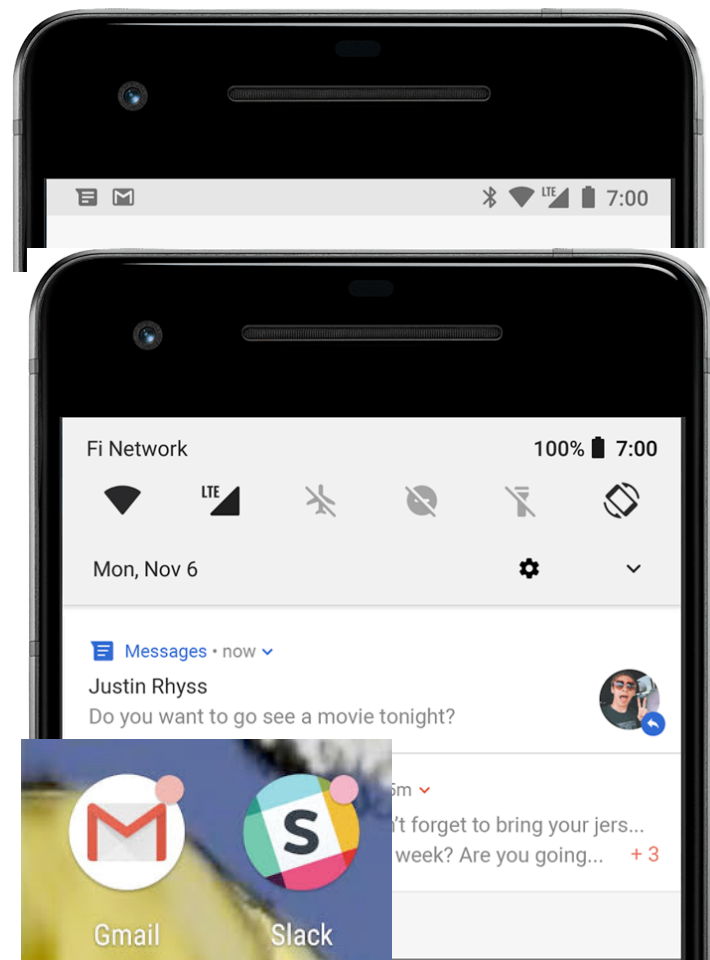
Notifications

Notifications

- Allows the app to initiate contact with the user

- Shown in the Status Bar (Notification Bar)
  - Custom icons
  - Alerts: sound, vibration, LED flashing
  - Notification drawer
  - On a locked screen
  - As a "badge"

https://developer.android.com/guide/topics/ui/notifiers/notifications

Notification Implementation

- Manage notifications via NotificationManager (or better yet via NotificationManagerCompat)
  - Post notifications with notify()
  - Clear notifications with cancel() and cancelAll()
- Build a notification via NotificationCompat.Builder

```
notification = new NotificationCompat.Builder(this)
        .setContentTitle(notificationTitle)
        .setContentText(notificationText)
        .setSmallIcon(R.drawable.icon)
        .setContentIntent(pendingIntent).build();
```

- Set a notification channel (in API 26 and above)

University *of Ljubljana*
Faculty *of Computer and*
Information Science

Notification Design

- Notifications are the only way for your app to initiate interaction with the user
  - Can be a core functionality of your app (e.g. Facebook messenger)
  - Use to show new/important information to the user
  - Use to increase user engagement
  - Use to get feedback from the user
  - Use to show that a Service is running
- Do not overuse and minimise interruptions

"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it"
Mark Weiser, 1991