CS194A

Android Programming Workshop

Lecture 7: Nov 3, 2021
Rahul Pandey
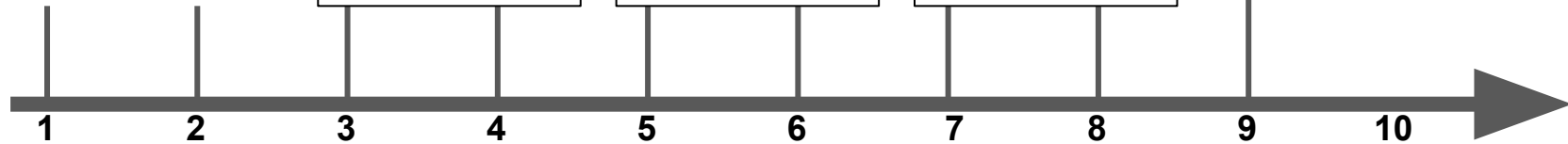
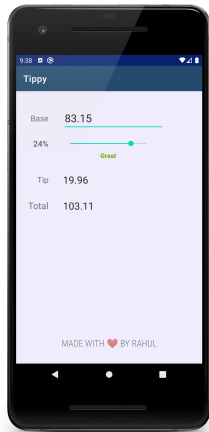# Outline

- Logistics
- Activity lifecycle
- Permissions
- Networking/APIs
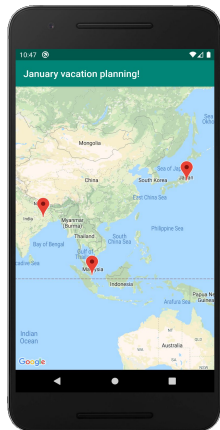
Outline

- **Logistics**
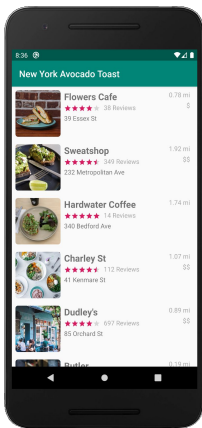- Activity lifecycle
- Permissions
- Networking/APIs

Tippy

Base   83.15

24%   Great

Tip   19.96

Total   103.11

MADE WITH 🧡 BY RAHUL

January vacation planning!

New York Avocado Toast

Flowers Cafe   0.78 mi
★★★☆☆ 38 Reviews $
39 Essex St

Sweatshop   1.92 mi
★★★★☆ 349 Reviews $$
232 Metropolitan Ave

Hardwater Coffee   1.74 mi
★★★★★ 14 Reviews
340 Bedford Ave

Charley St   1.07 mi
★★★★☆ 112 Reviews $$
41 Kenmare St

Dudley's   0.89 mi
★★★★☆ 097 Reviews $$
85 Orchard St

Assn 1: Tip
Calculator

Assn 2:
Google Maps

Assn 3: Yelp
Clone

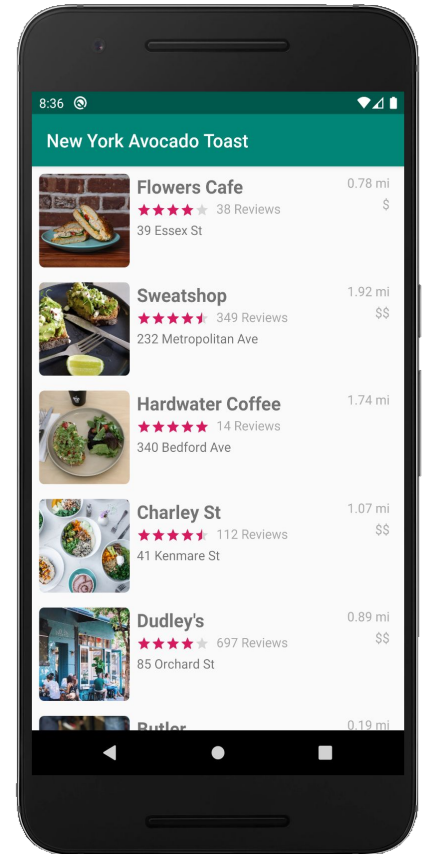Industry panel
discussion

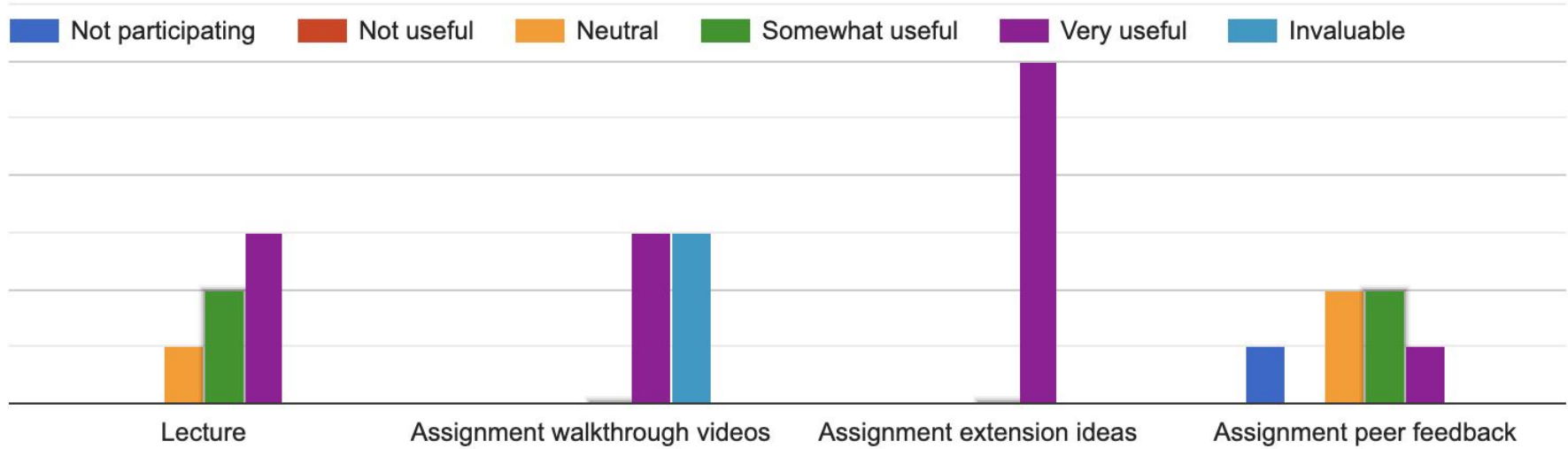1   2   3   4   5   6   7   8   9   10

Week #

Assignment #3- Yelp Clone

- The last assignment!
- Gives you practice with Retrofit and networking
- Many extension possibilities
- Due Sunday, November 14

Mid-quarter survey readout

Code review

- Are there logic errors or edge cases that weren't considered?
- Is the code written idiomatically so that it follows best practices for the language and can be maintained in the future?
- Are there alternative (potentially cleaner) approaches to implementing the same functionality?

- Company/team socialization!

Outline

- Logistics
- **Activity lifecycle**
- Permissions
- Networking/APIs

Activity can be in a number of states

- States:
  - **Resumed**: activity is in foreground
  - **Paused**: activity is partially obscured by another activity. Activity cannot receive user input or execute code
  - **Stopped**: activity is hidden/in the background. Things like member variables are maintained
  - **Destroyed**: Resources of the activity are reclaimed by the Android system. E.g. back button press
- Android system will notify you when a state transition happens
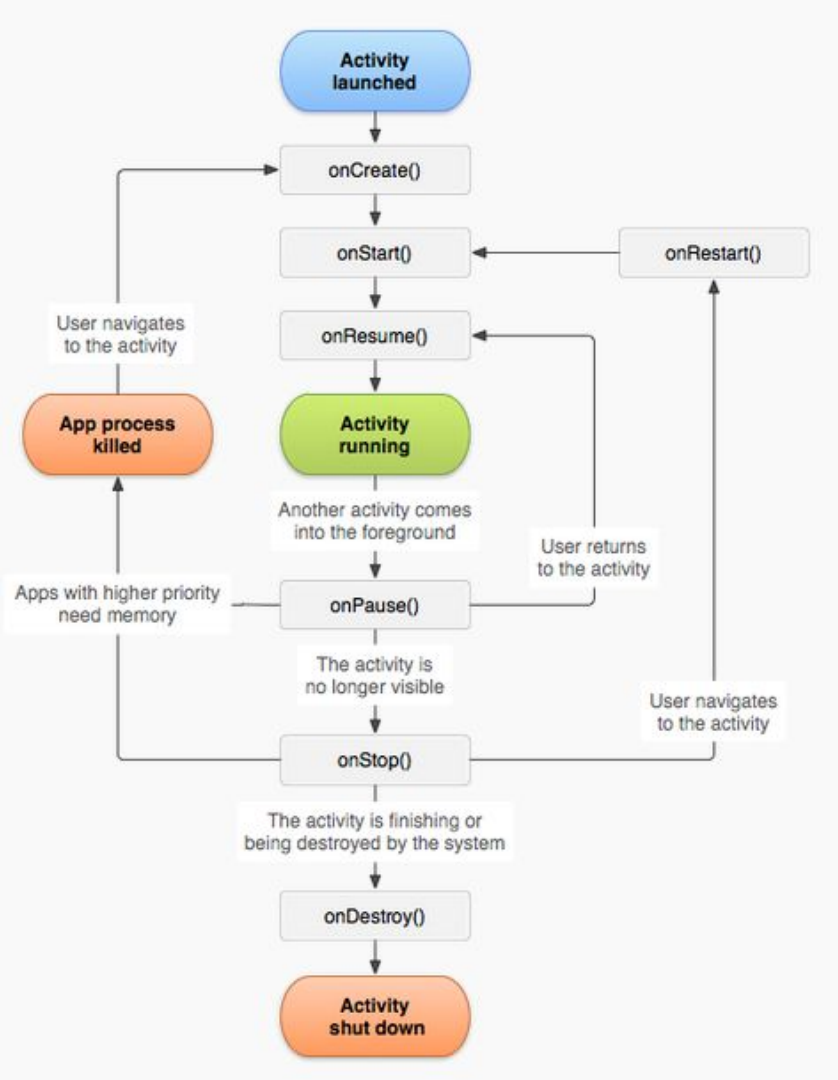
Outline

- Logistics
- Activity lifecycle
- **Permissions**
- Networking/APIs

Permissions

- Defines what your app can and cannot access- designed to protect the user's privacy.
- E.g. access the camera, see your contacts, get your location
  - [Full list of permissions](#)
- **Normal permissions** are automatically granted to the app
- **Dangerous permissions** must be accepted by the user

Permissions model- change in 2015

- Pre-6.0: Permissions granted at install time
- Starting with 6.0 (Android Marshmallow): Permissions granted at runtime when needed
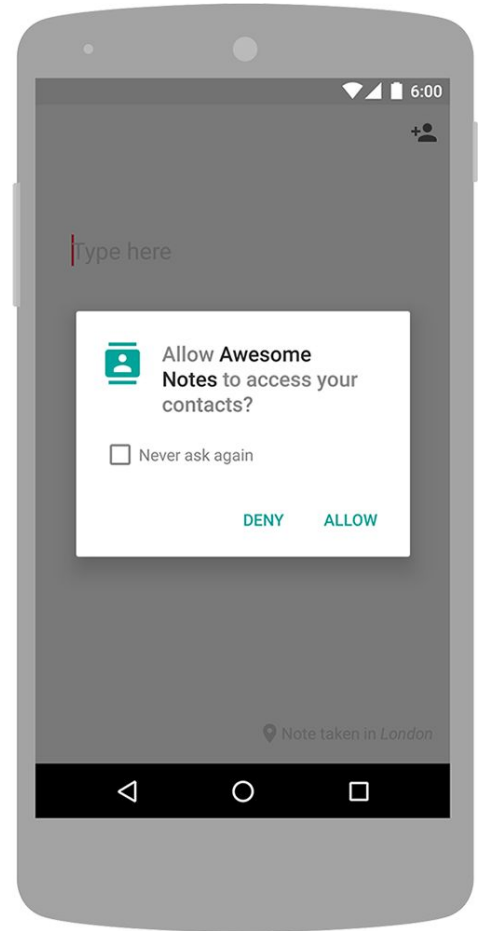
```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.stanford.rkpandey.week3">

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

    <application...>

</manifest>
```

```
if (ContextCompat.checkSelfPermission(this, Manifest.permission.READ_CONTACTS) !=
PackageManager.PERMISSION_GRANTED) {
    // Permission is not granted
    // Check if we should show an explanation
    if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.READ_CONTACTS)) {
        // Show an additional explanation to the user *asynchronously*
    } else {
        // No explanation needed, we can request the permission.
        ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.READ_CONTACTS),
    REQUEST_READ_CONTACTS)
        // REQUEST_READ_CONTACTS is an app-defined int constant. Handle the result in
    onRequestPermissionsResult.
    }

} else {
    // Permission has already been granted
}
```

My Maps <mark>highlights</mark>

Luna Yang

My Maps **highlights**

Megan Worrel

My Maps <mark>highlights</mark>

Meng-Jin Lin

Outline

- Logistics
- Activity lifecycle
- Permissions
- **Networking/APIs**

Networking

- Most interesting apps use the internet to retrieve or send data
- Our application needs to know where to get the data and how to use it
- **Important**: Do not make network requests on the main (UI) thread
  - Long-running, flaky operation which should not occur on the UI thread
  - Avoid jank in your app

Web services

- **Client**: your mobile application
  - Makes a query to a URL
- **Server**: a computer which listens for requests on that URL
  - Responds to the query as appropriate (e.g. database query)
- **API:** defines how the client and server communicate
  - Language of data transfer: JSON, XML, HTML, something else
  - Example [request] to a movie API

API: defines how software apps communicate
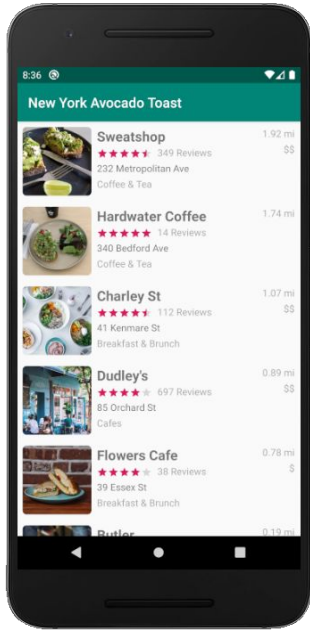
I want to search for restaurants in New York that serve avocado toast.

Here's a list of businesses:
1. Flowers Cafe
2. Sweatshop
3. ……………………

Networking library

- You'll almost always use a library for networking
- AsyncHttpClient
- Retrofit
- Volley
- Ion

Prep for next week

Start the **Yelp clone**