

Android University - Fall 2020

Week 1: Layouts, Scrolling Feeds, Network Requests

Online course guidelines



Make sure to mute yourself



Turn on your webcam



Use chat room during class, discussion forum outside of class



Keep an eye on Slack:

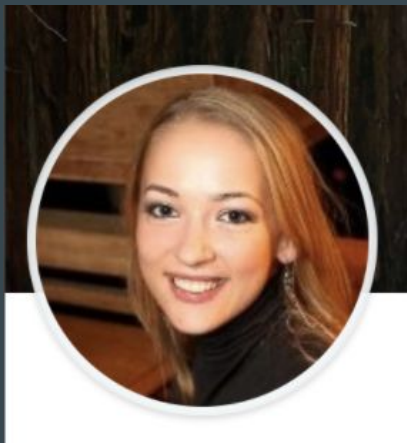
android-remote-fall20 (general announcements)

android-help-fall20 (help)



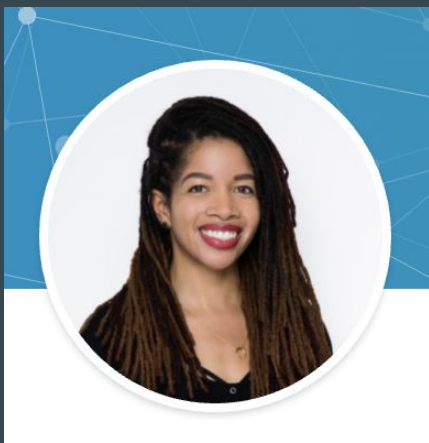
We'll take breaks during a session!

Instructors intro



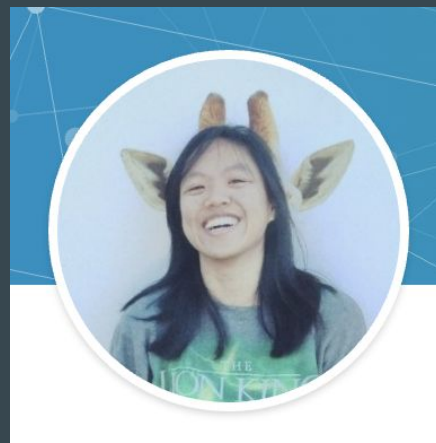
Marina

Week 1-2



Deonna

Week 3-4



Caren

Week 5-6

TAs intro



Lovpreet Hansra (TA)(TF) ○



Anthony Tran ○



Giancarlo Garcia Deleon



Nicolas Lara Fonseca ○

General course structure

- Adapted from semester long on-campus course
 - original course meets twice a week
 - if you like this course you can bring it to your campus!
- Current class meets once a week on Saturday
 - we go over Android concepts
 - we have live coding 'labs'
 - weekly assignment is due at midnight on Friday (right before next session that runs on Saturday)


How to succeed in this course

- Remember that coding is HARD!
- Allocate 5-10 hours per an assignment (tip - block time in your calendar)
- Android Studio can be frustrating
- Set goals, find accountability partner / study buddy
- Reach out for help (Slack, chat rooms, discussion forum, connect with other students)

Course portal

CodePath Courses


Courses ▾ Cohorts ▾ Locations ▾ Quick Links ▾ Organizer Links ▾

 Marina Tanasyuk ▾

Android University

Overview

Submitting Assignments

Unit 0 

Unit 1

Unit 2

Unit 3

Unit 4

Unit 5

Unit 6


Overview


Assignment



Lab

Hints

Reading


 Tech Fellow



 Grading

 Submit your [Flicks Part 1](#) assignment by **Friday, September 25th at 12:00am PDT** using the **Submit** button on the [Assignment tab](#) 

Unit 1 - Layouts, Scrolling Feeds, and Making Network Requests

Getting Started!

 **Are you just getting started?** If you haven't already, be sure to start with [Unit 0 - Getting Started](#) before jumping into this unit!

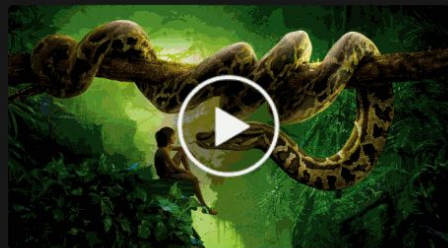
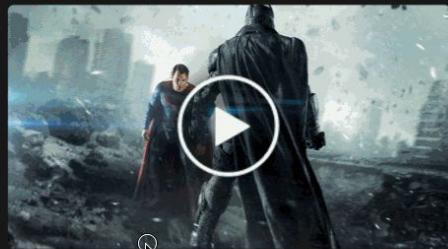
This includes reviewing the [Android Studio Starter Guide](#), watching the  [Course Orientation](#) and watching our  [How to Succeed in Android Development 4-part Series](#).

Flicks App - intro



The Rift

The satellite has vanished and the only clue is a trail leading to a small village nestled on the outskirts of forest line. As the team begins to investigate they quickly discover that something else came back from space, something not of this world. As...



Flicks App - what we need to know



The Rift

The satellite has vanished and the only clue is a trail leading to a small village nestled on the outskirts of forest line. As the team begins to investigate they quickly discover that something else came back from space, something not of this world. As...

1. How to create different screens for your app
2. How to build UI / Layout for each screen
3. How to surface a list of similarly looking items (each item has a poster image, a title and a description)
4. How to fetch and persist data for this list

Common components of Android apps

- Show data
- Create content and take users through different flows (aka features)
- Fetch and persist data
- Polished user interface



Android Concepts - week 1

Activity

Layout

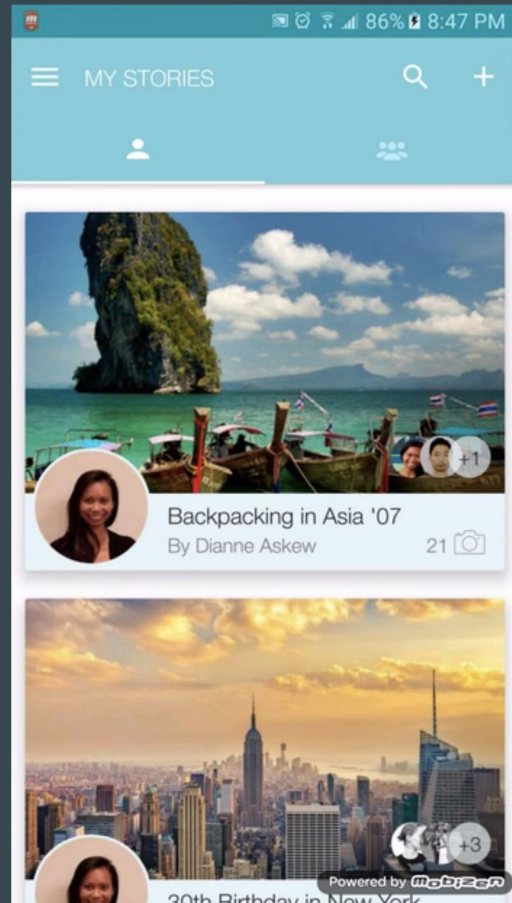
RecyclerView

Layout Manager

AsyncHttpClient

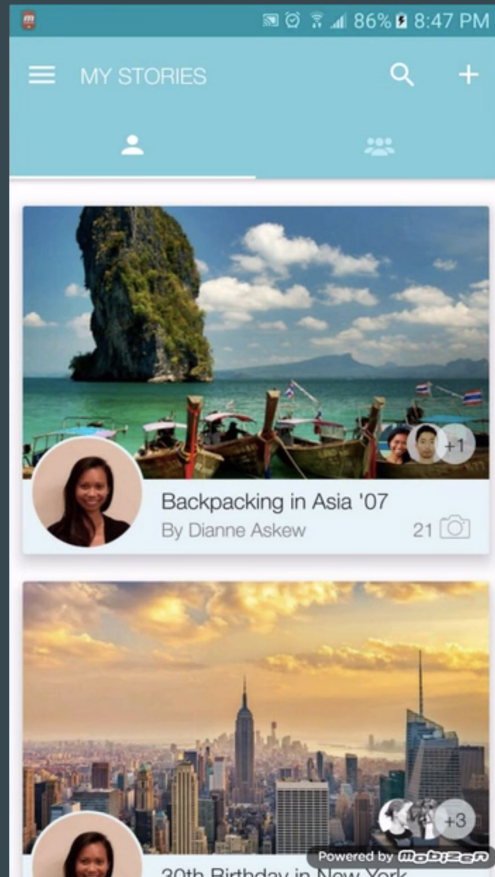
Activity

- Screen = Activity
- Activity is where you write code to surface different content, handle user interactions and more!



Layouts

- Layout = XML file
- XML = markup language
- Add, position and customize views
- Views = Buttons, TextViews, ImageViews, and more



Bridging Activity and Layout

MainActivity.java

activity_main.xml

Handles user interaction

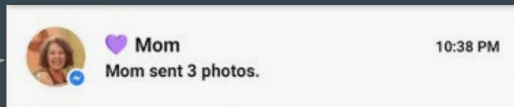
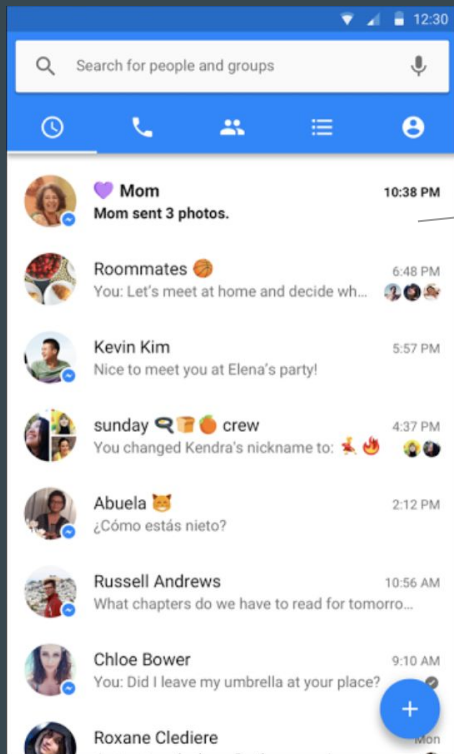
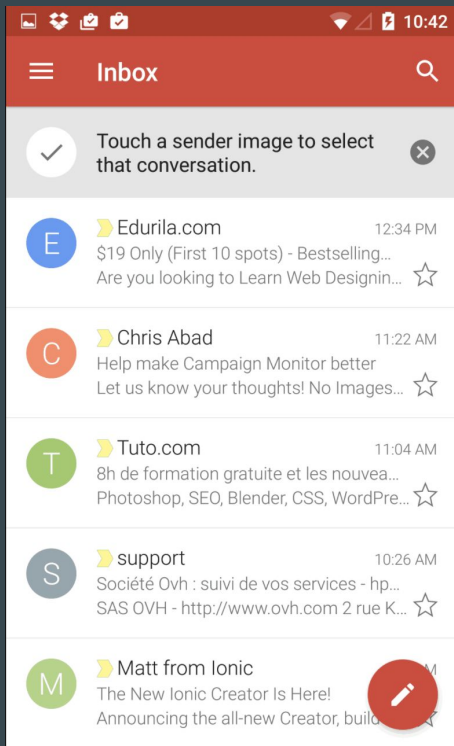
Sets up views (aka
programmatically -
advanced)

Sets up views

Thoughts / Questions 

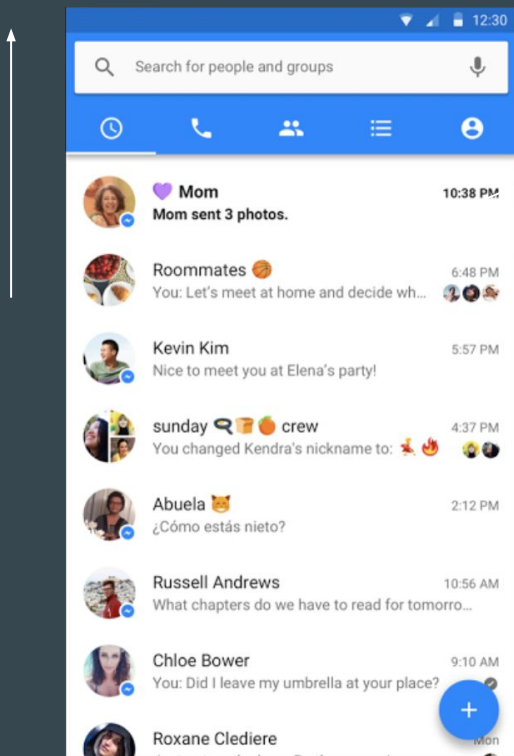
5 min break 🤫

RecyclerView - what and why

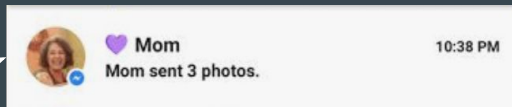


Parsing data for layout
Creating classes
Inflating views in the layout

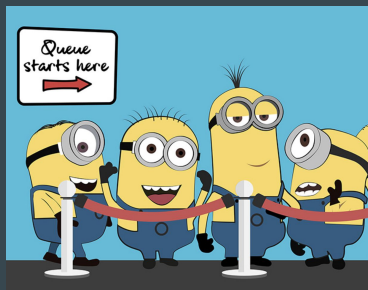
RecyclerView - how



1. Layout instance scrolls out of view

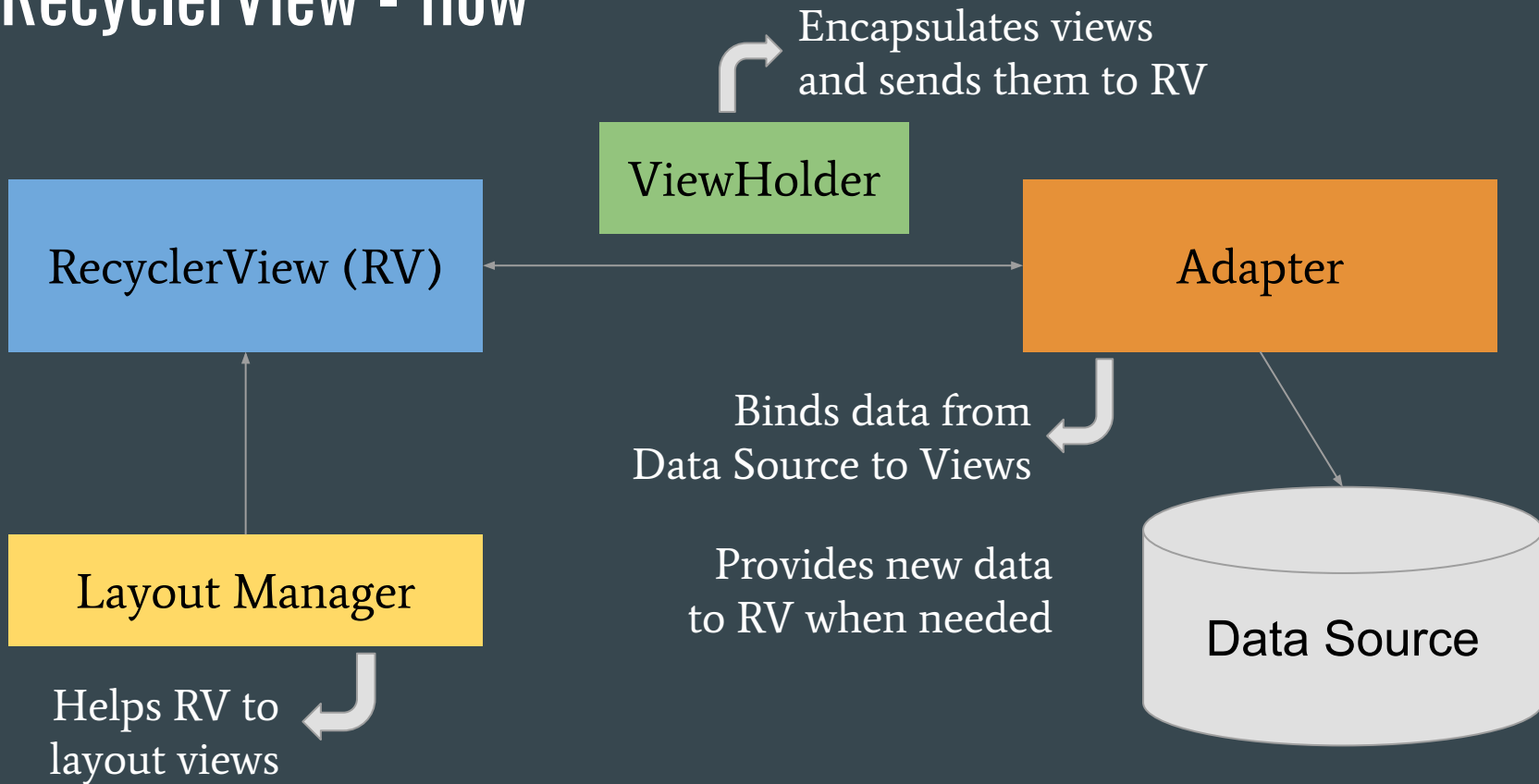


2. Placed in queue

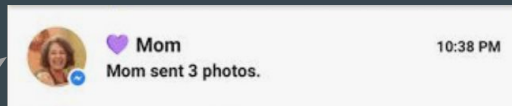
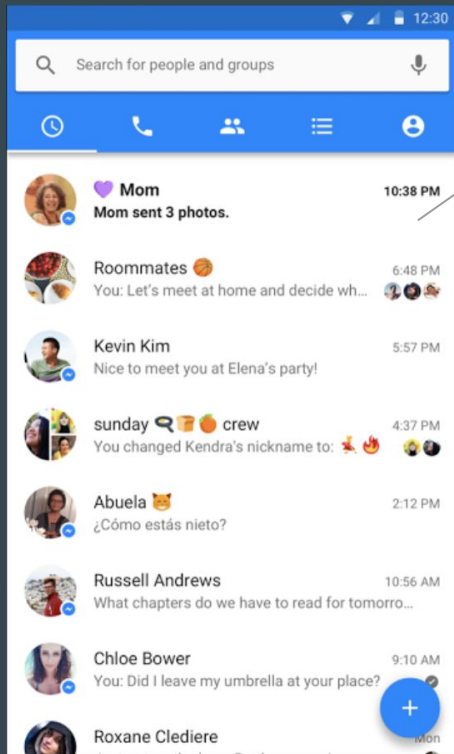


3. Filled with new content and scrolls in again

RecyclerView - flow



Data Source - Data Models



```
public class Message {  
  
    String id;  
    String senderName;  
    String messagePreview;  
    String timeSent;  
  
    public Message(String id, String senderName, String messagePreview, String timeSent)  
    {  
        this.id = id;  
        this.senderName = senderName;  
        this.messagePreview = messagePreview;  
        this.timeSent = timeSent;  
    }  
}
```

Live Coding (RecyclerView)

Making network calls - intro

In order to display a list of movies that are currently playing (for instance), we'd need to access data stored on the internet.

APIs allow us to access all sorts of different data.

Flicks app API 'GET' call:

https://api.themoviedb.org/3/movie/now_playing?api_key={your_api_key}

Making network calls - JSON response

```
{
  "results": [
    {
      "vote_count": 2150,
      "id": 335983,
      "video": false,
      "vote_average": 6.6,
      "title": "Venom",
      "popularity": 225.576,
      "poster_path": "\/2uNW4WbgBXL25BAbXGLnLqX71Sw.jpg",
      "original_language": "en",
      "original_title": "Venom",
      "genre_ids": [
        878
      ],
      "backdrop_path": "\/VuukZLgaCrho2Ar8ScL9HtV3yD.jpg",
      "adult": false,
      "overview": "When Eddie Brock acquires the powers of a symbiote, he will have to release his alter-ego \"Venom\" to save his life.",
      "release_date": "2018-10-03"
    },
    {
      "vote_count": 881,
      "id": 424694,
      "video": false,
      "vote_average": 8.3,
      "title": "Bohemian Rhapsody",
      "popularity": 134.625,
      "poster_path": "\/LHulwtNaczFPGFDTrjCSzeLPTKN.jpg",
      "original_language": "en",
      "original_title": "Bohemian Rhapsody",
      "genre_ids": [
        18,
        10402
      ],
      "backdrop_path": "\/pbXgLEYh8rIG2Km5IGZPnhcnuSz.jpg",
      "adult": false,
      "overview": "Singer Freddie Mercury, guitarist Brian May, drummer Roger Taylor and bass guitarist John Deacon take the music world by storm when they form the rock 'n' roll band Queen in 1970. Hit songs become instant classics. When Mercury's increasingly wild lifestyle starts to spiral out of control, Queen soon faces its greatest challenge yet \u2013 finding a way to keep the band together amid the success and excess.",
      "release_date": "2018-10-24"
    }
  ]
}
```

Making network calls - threads

Android app has only one Main thread.

All user interactions are handled on Main thread. That's why it's also called UI thread.

Network calls can take a while to complete.

Use background thread for anything other than interacting with UI, otherwise you might run into an ANR (Application Not Responding) dialog.

Managing threads yourself can be a very daunting process and is quite error-prone.

Making network calls - AsyncHttpClient

```
AsyncHttpClient client = new AsyncHttpClient();

client.get("https://www.google.com", new JsonHttpResponseHandler() {

    @Override
    public void onSuccess(int statusCode, Header[] headers, JSONObject response) {
        // this method will get called if the network request succeeds
    }

    @Override
    public void onFailure(int statusCode, Header[] headers, String responseString, Throwable e) {
        // this method will get called if the network request fails
    }

});
```

Thoughts / Questions 

5 min break 🤫

Live Coding (Lab)