

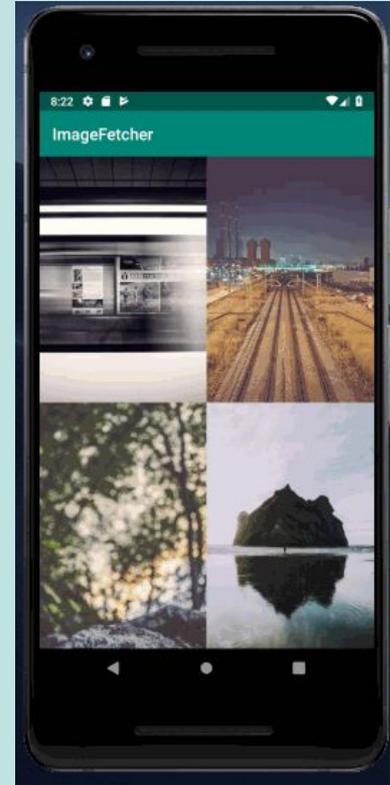
CSc 317

Service, API, JSON



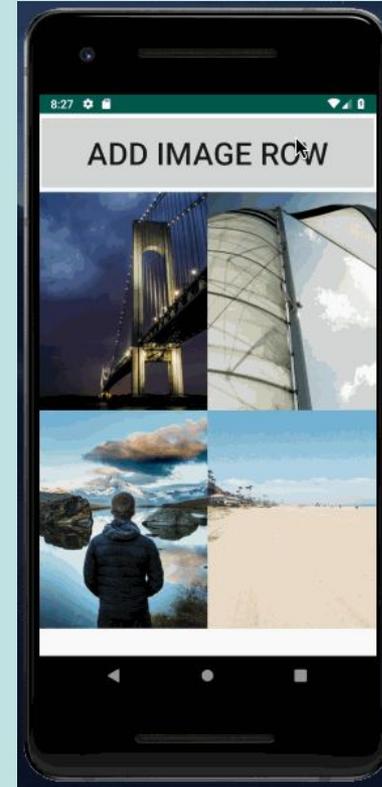
ImageFetcher

- From last time:
 - An Intent service that would download images at an interval, and would sent a message with the bitmap
 - ***Does it continue to run in background?***



ImageFetcher Improvements

- Change the imagefetcher so that:
- Starts with 4 images (two rows)
- Has button to add row of 2 images
- Continues to work in background
- Don't display action bar



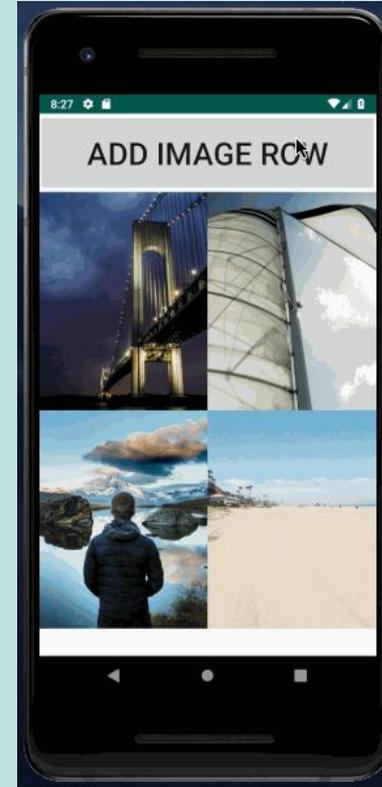
ImageFetcher, Step 1 (2 mins)

- Comment out **VERSION A** of the XML, un-comment out **VERSION B**
- Note the difference



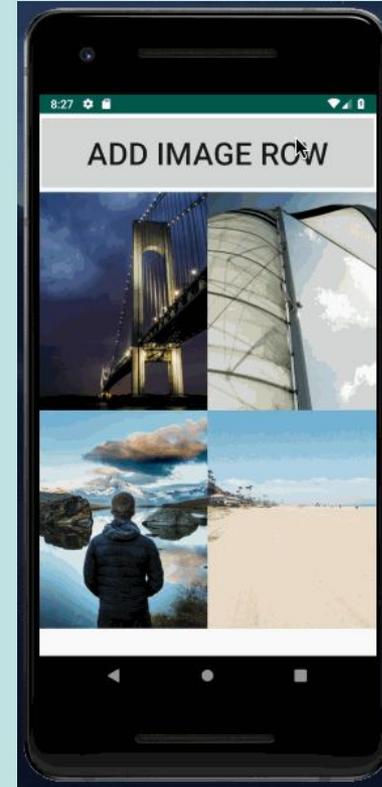
ImageFetcher, Step 2 (10 mins)

- Implement **addRowOfImageViews**
 - See the comment . . .
- (1) Create two new image views (iv1 and iv2)
 - (2) Add the image views to the imageViews ArrayList
 - (3) Set the ImageView.ScaleTyle to CENTER_CROP
 - (4) Call the function to update all image view Dimensions
 - (5) Create a new horizontal LinearLayout (ll)
 - (6) Add the two ImageViews to the LinearLayout (ll)
 - (7) Add ll to the outer LinearLayout with id R.id.outer_linear



ImageFetcher, Step 3 (7 mins)

- Implement `updateAllImageViewDimens`
 - See the comment . . .
- (1) Create a new `LinearLayout.LayoutParams` and set
 - (2) The width to be half of the screen width
 - (3) The height to be a fraction of the height, based on the number of rows
 - (4) (optional) set the margins to 0
 - (5) set the layout params
 - (6) call `invalidate()` on the `ImageView`



ImageFetcher, Step 4 (5 mins)

- Change `getNextId`
- See the comment . . .

Update this function so that it does not just cycle through four image IDs. Instead, it should cycle between `imageViewIdBegin` and `imageViewIdEnd`. You can use `imageViewIdNext` to keep track of the next `imageView` id to update.

