CSE118 Spring 2023

Assignment 3

In this assignment you will single-Activity, multi-Fragment, network connected Android App to reinforce your understanding of the following concepts:

- Accessing RESTful APIs using `HttpsURLConnection`
- Separation of responsibilities between Fragments
- Observing changes in `LiveData`
- `ViewModel` event handling
- Use of the `FloatingActionButton`
- Responding to swipe actions on `RecylerView`
- Creating Dialogs
- Using Snackbars

This assignment is worth 10% of your final grade.

Setup

Download the starter code archive from Canvas and expand into an empty folder. I recommend creating a folder for the class and individual folders beneath that for each assignment.

Start Android Studio and import the project: **File → New → Import Project**

Select the folder where you expanded the starter code and click **Open**.

The `cse118.com` RESTful API

Endpoints are at: **https://cse118.com/api/v0**

Method	Path	Description
POST	`/login`	Use your UCSC email address, and student number as your password
GET	`/workspace`	Returns all workspaces available to the logged in user
GET	`/workspace/{id}`	Returns all channels in the workspace with the supplied id
GET	`/channel/{id}`	Returns all messages in the channel with the supplied id
POST	`/message/{id}`	Creates new message in the channel with the supplied id
DELETE	`/message/{id}`	Deletes the message with the supplied id

You can explore the API here: https://cse118.com/api/v0/api-docs/

For more details, consult the handouts and recording from the Android V lecture, ask questions in Section and Office Hours, and study the 06 Ball Clubs example.

Do not use the `/ballclub` endpoints other than to run the Ball Clubs Example.
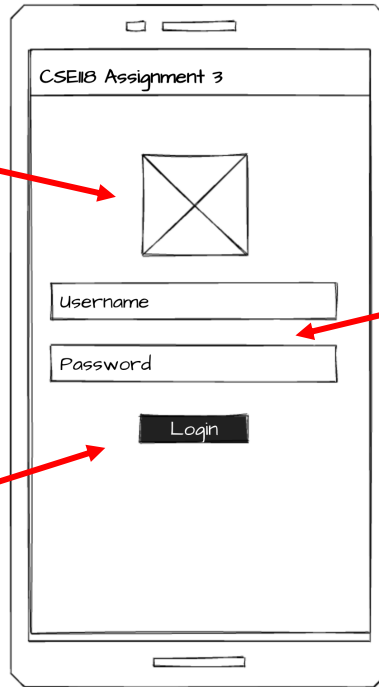
Requirements

A client has given you a set of wireframes / mock-ups for a vaguely Slack-like messaging system they want you implement for Android. Your client has several requirements…

Basic: (instrumented tests are provided for this requirement)

Show a functional login screen, before listing workspaces.

UCSC Slug Logo from the supplied drawable resource

CSE118 Assignment 3

Username

Password

Login

Entry fields are now active, the login button must not be enabled until there are at least four characters in each entry field (supplied tests define what their resource ids must be)

Clicking the Login button takes logs into cse118.com

List Workspaces returned by cse118.com.

< User Name

Workspace 1
32 Channels

Workspace 2
5 Channels
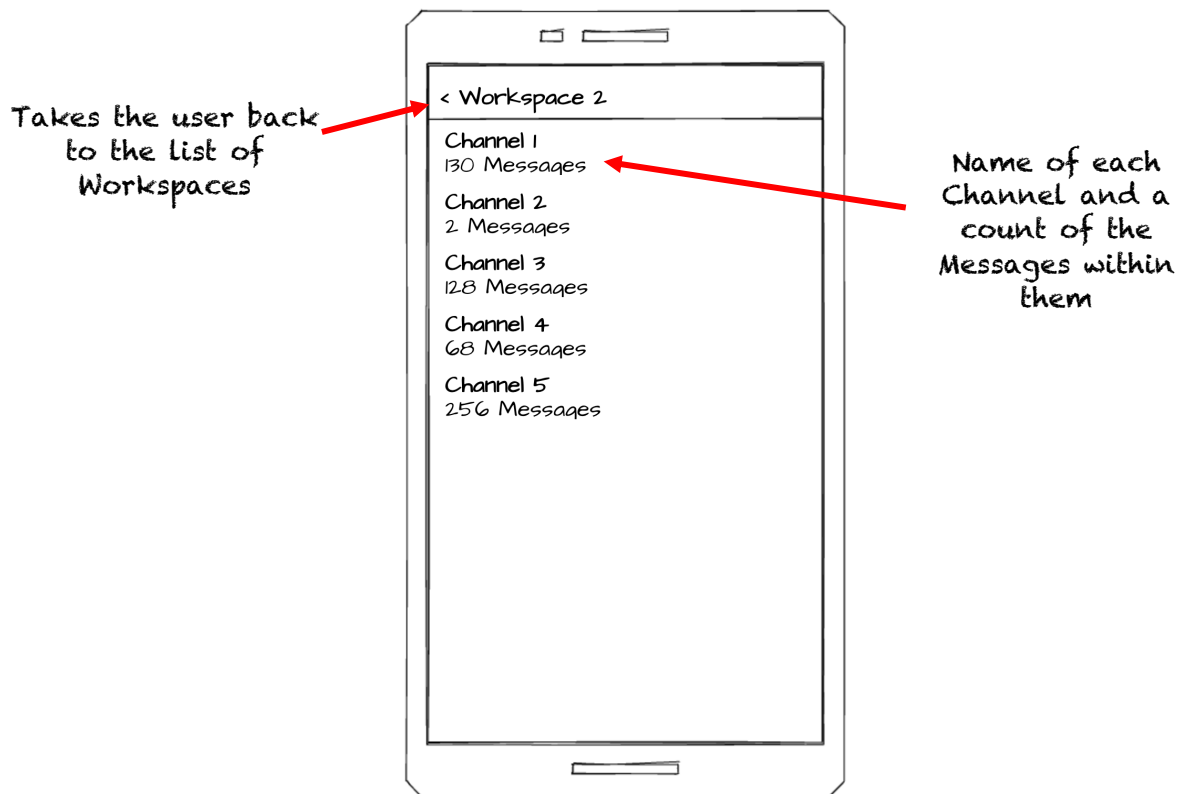
Workspace 3
16 Channels

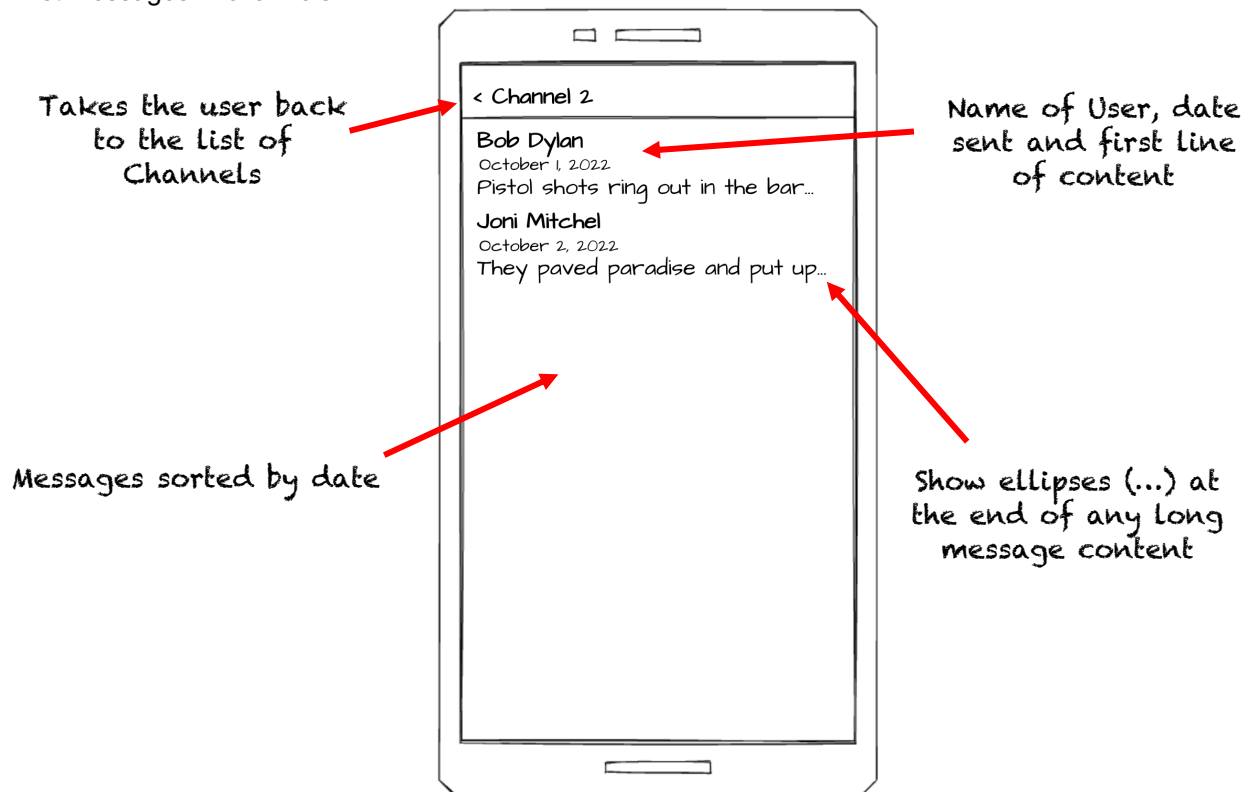Logged in user's name appears at the top of the list of workspaces
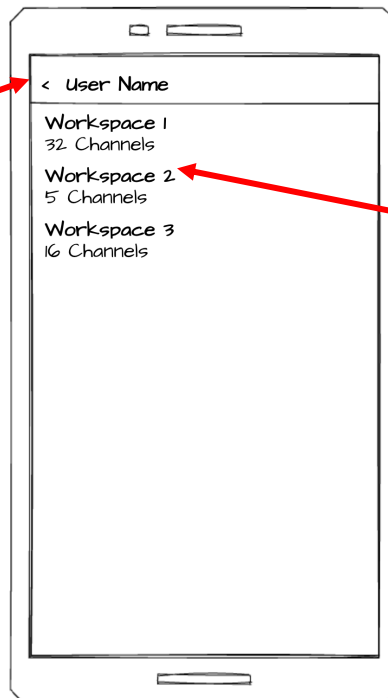
Name of each Workspace and a count of the Channels within them

When a Workspace is selected, show a list of channels in that Workspace.

Takes the user back to the list of Workspaces

< Workspace 2

Channel 1
130 Messages

Channel 2
2 Messages

Channel 3
128 Messages

Channel 4
68 Messages

Channel 5
256 Messages

Name of each Channel and a count of the Messages within them

List messages in channels:

Takes the user back to the list of Channels

< Channel 2

Bob Dylan
October 1, 2022
Pistol shots ring out in the bar…

Joni Mitchel
October 2, 2022
They paved paradise and put up…

Name of User, date sent and first line of content

Messages sorted by date

Show ellipses (…) at the end of any long message content

When a message is selected, show its contents:

Takes the user back to the list of Messages

Name of User

< Bob Dylan

October 1, 2022

Date sent

Pistol shots ring out in the barroom night Enter Patty Valentine from the upper hall She sees a bartender in a pool of blood Cries out, "my God, they killed them all" Here comes the story of the Hurricane The man the authorities came to blame For somethin' that he never done Put in a prison cell, but one time he coulda been The champion of the world Three bodies lyin' there, does Patty see And another man named Bello, movin' around mysteriously "I didn't do it" he says, and he throws up his hands "I was only robbin' the register, I hope you understand"
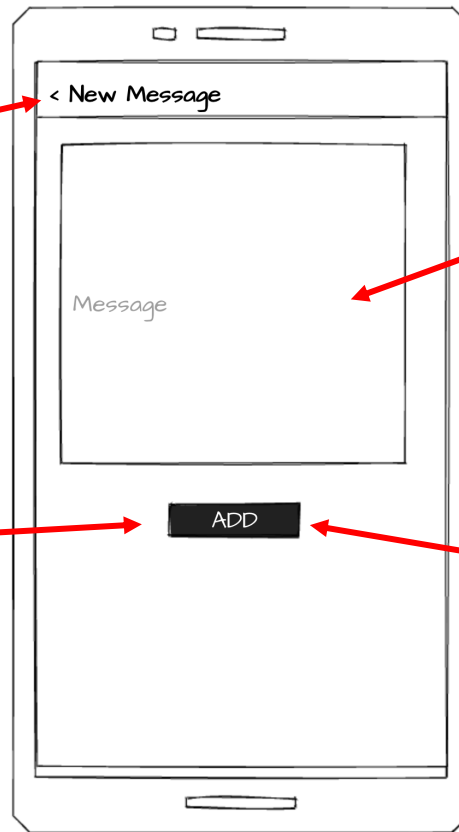
Content of message

Advanced:

On the list of messages in a channel, show a `FloatingActionButton` that when clicked allows the user to create a new message in the current channel.

< Channel 2

Bob Dylan
October 1, 2022
Pistol shots ring out in the bar...

Joni Mitchel
October 2, 2022
They paved paradise and put up...

When clicked, user can create a new message in the current channel.
Must have a resource id of "fab"
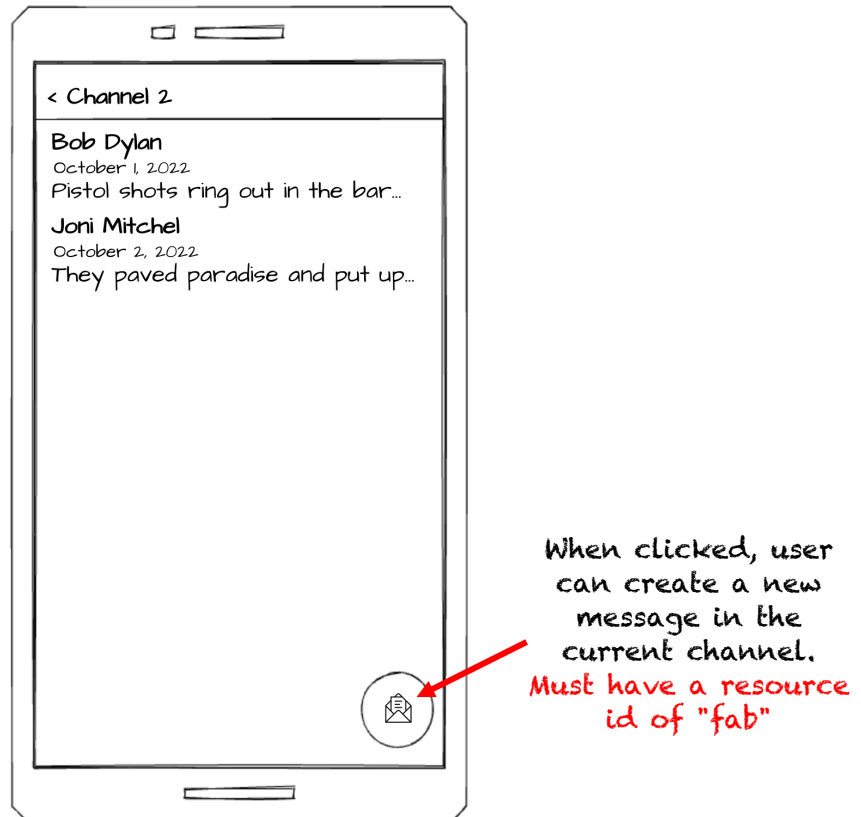
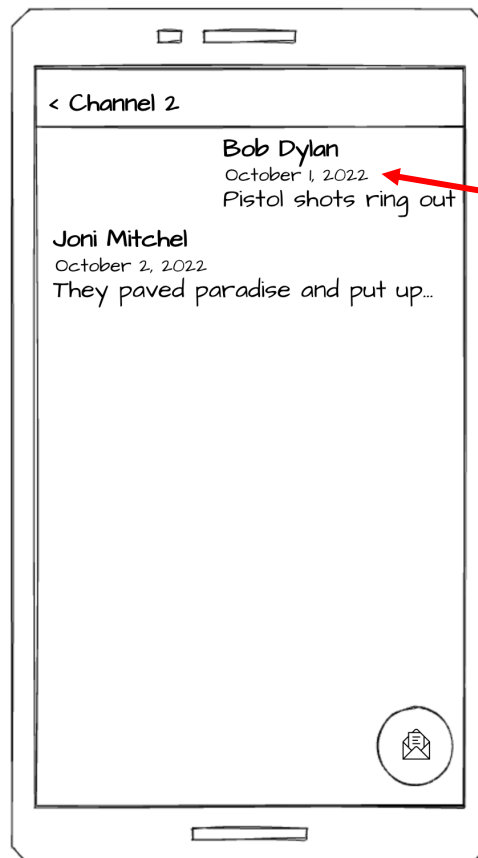Takes the user back to the list of Messages

< New Message

Content of new message Must have an resource id of "content"

Message

Disabled until 16 or more characters are in the message content

ADD

When clicked, adds a new message in the current channel, returns to the previous screen and shows a snackbar stating "Message Created"
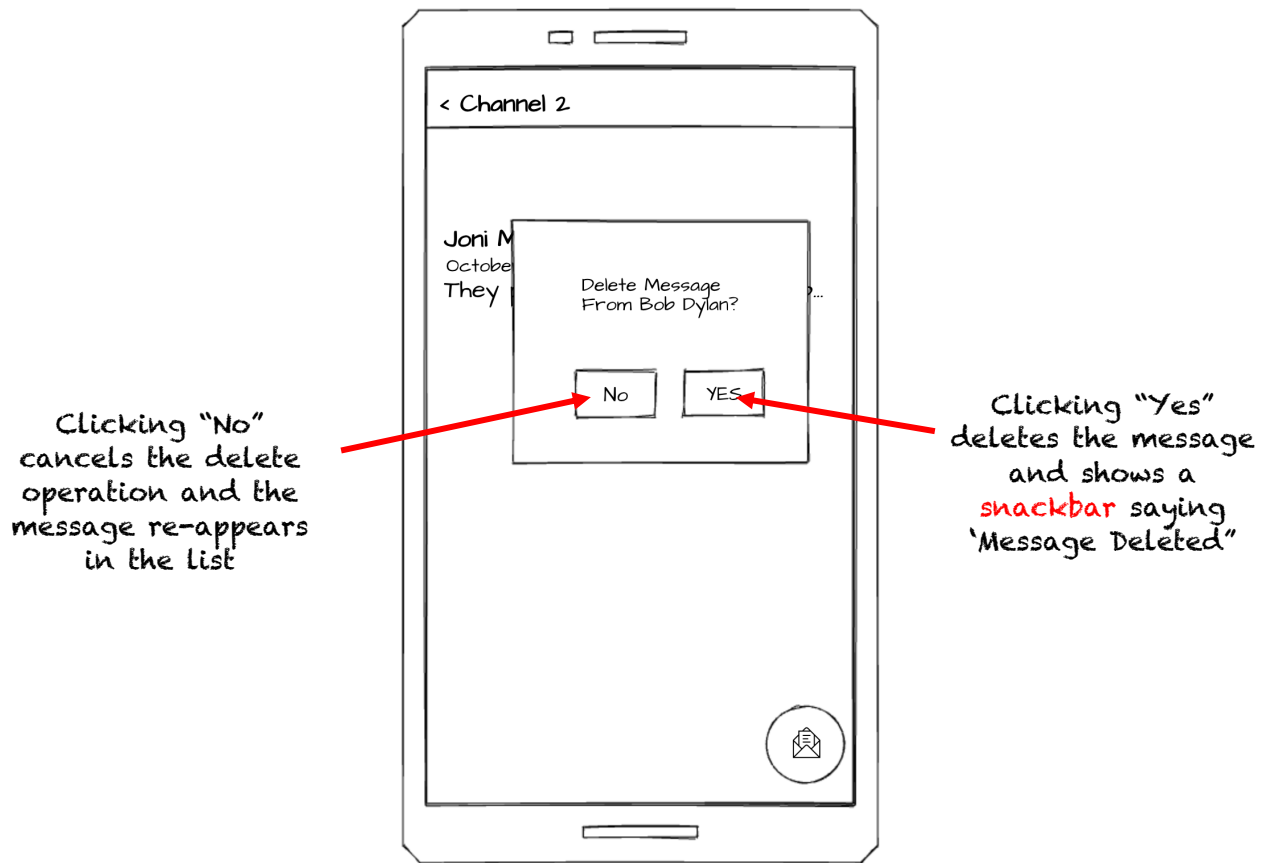
Stretch:

Delete a message when the user swipes right on it:

< Channel 2

Bob Dylan
October 1, 2022
Pistol shots ring out

Joni Mitchel
October 2, 2022
They paved paradise and put up...

As the user swipes right, message moves in that direction

When the user completes the swipe, show a confirmation dialog:

Clicking "No" cancels the delete operation and the message re-appears in the list

Delete Message From Bob Dylan?

No   YES

Clicking "Yes" deletes the message and shows a snackbar saying 'Message Deleted"

ONLY DELETE MESSAGES YOU CREATED!

Extreme:

Write Automated tests for Advanced and Stretch Requirements that pass when executed against the model Solution.

To achieve this the only View IDs you should rely on are `R.id.recyclerview`, `Ri.id.fab` and `R.id.content` other views should be found by the text they display.

What steps should you take to tackle this?

Whilst the order in which you put your code together is entirely up to you, a plausible initial development schedule might include the following steps:

1. Define Serializable Data Classes for:

- Member
    - Name
    - Email
    - Role (user or admin)
    - Access Token (returned by cse118.com when you log in)
- Workspace
    - ID
    - Name
    - Channels (how many channels are in the workspace)
- Channel
    - ID
    - Name
    - Messages (how many messages are in the channel)
- Message
    - ID
    - Poster (name of the person who posted it)
    - Date
    - Content

3. Create a ViewModel to store the logged in member and workspaces.

2. Create a Login Fragment and handle a login request by connecting to cse118.com.

4. On successful login, show a list of workspaces.

5. Create a Channel Fragment and handle the user selecting a workspace by showing the channels within it after making a call to cse118.com to find them

6. Create a Message Fragment and handle the user selecting a channel by showing the messages within it after making a call to cse118.com to find them.

7. Create a Message Detail Fragment and handle the user selecting a message by showing full content.

Once you have these steps complete, you've satisfied the Basic Requirement and can move on to Advanced.

How much code will you need to write?

A model solution that satisfies all requirements has approximately 1100 lines of Kotlin and 500 lines of XML.

Grading scheme

The following aspects will be assessed:

1. (100%) **Does it work?**

    a. Basic   (60%)
    b. Advanced   (15%)
    c. Stretch   (15%)
    d. Extreme   (10%)

2. (-100%) **Did you give credit where credit is due?**

    a. Your submission is found to contain code segments copied from on-line resources or created by code generation tools and you failed to give clear and unambiguous credit to the original author(s) in your source code You will also be subject to the university academic misconduct procedure as stated in the class academic integrity policy. (-100%).

    b. Your submission is determined to be a copy of a past or present student's submission. (-100%)

    c. Your submission is found to contain code segments copied from on-line resources that you did give a clear an unambiguous credit to in your source code, but the copied code constitutes too significant a percentage of your submission:

        ○ < 25% copied code   No deduction
        ○ 25% to 50% copied code   (-50%)
        ○ > 50% copied code   (-100%)

What to submit

On the console (PowerShell on Windows), navigate to the folder you extracted the starter code into and run the appropriate command to create the submission archive:

Windows:
```
$ gradlew.bat clean
$ Compress-Archive -Path . -DestinationPath Assignment3.Submission.zip
```
Linux:
```
$ ./gradlew clean
$ zip -r Assignment3.Submission.zip *
```
Mac:
```
$ ./gradlew clean
$ zip -x "*.DS_Store" -r Assignment3.Submission.zip *
```

** **UPLOAD** `Assignment3.Submission.zip` **TO THE CANVAS ASSIGNMENT AND SUBMIT** **