CSE118 Fall 2022

Assignment 6

In this assignment you will develop a simple Swift / SwiftUI app to display hierarchical data loaded from a RESTful API to solidify your understanding of:

- HTTPS Connections
- RESTful APIs
- JSON Serialization
- Asnchronicity and Parallelism
- SwiftUI Views & Navigation
- Automated UI testing

This assignment is worth 12% 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 Xcode and open the project: **File → Open**

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

The `cse118.com` RESTful API

Endpoints are at: **`https://cse118.com/api/v1`**

You can explore the API here: https://cse118.com/api/v1/api-docs/

Note that whilst they have similar names, the end points are not always the same as those in the API for Assignment 3.

You can login with your UCSC email address (use your student ID as the password) but for the Basic Requirement you can use molly@cse118.com with a password of "molly".

You are all in a Workspace owned by me and can post messages to the one Channel within it but take care to be polite to each other and avoid anything that might cause offence. In short, be sensible.

Requirements

Basic:

Implement a SwiftUI iOS App that loads Slack-like hierarchical data from the RESTful API and passes the supplied UI tests by implementing the following:

Login:

On successful authentication, show the Workspace list.

Workspace List:

Workspaces the authenticated user either owns or is a member of. Each Workspace should show the number of Channels within it. Select a Workspace to see Channels within it.

Also show a tool-bar button to log the user out.

Channel List:

Channels in the selected Workspace, each with a count of the Messages they contain.

Message List:

Messages in the selected Channel showing the Member/user who posted it, the content, and the date posted in long form.

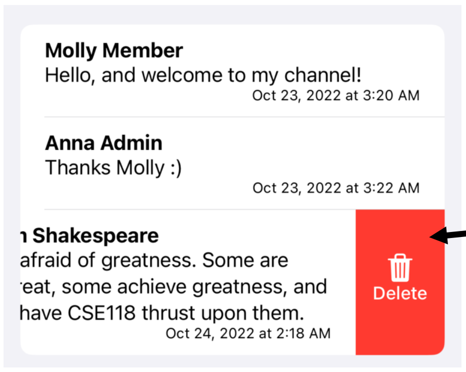
Add Message:

Allow the authenticated user to create a new Message in the current Channel.

Delete Message:

Allow the authenticated user to delete any Message they created, and Workspace Owners to delete any message in the Channels they own regardless of who created it.

In addition, prevent the authenticated user from deleting messages they did not create in Workspaces / Channels they do not own.

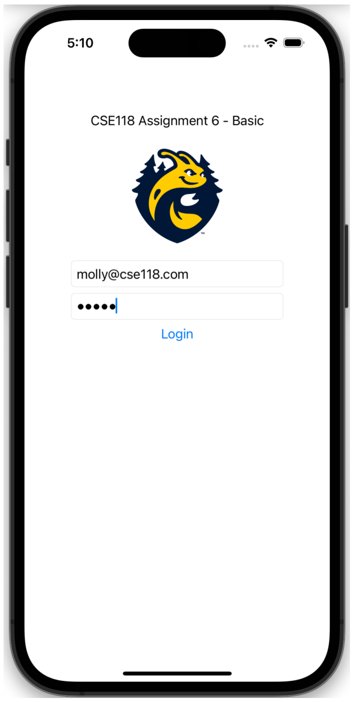

Molly Member
Hello, and welcome to my channel!
Oct 23, 2022 at 3:20 AM

Anna Admin
Thanks Molly :)
Oct 23, 2022 at 3:22 AM

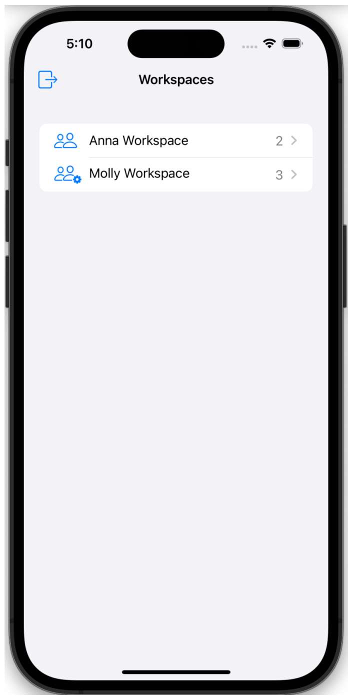
ı Shakespeare
afraid of greatness. Some are
ɾeat, some achieve greatness, and
have CSE118 thrust upon them.
Oct 24, 2022 at 2:18 AM

Accessibility Label = `Delete`

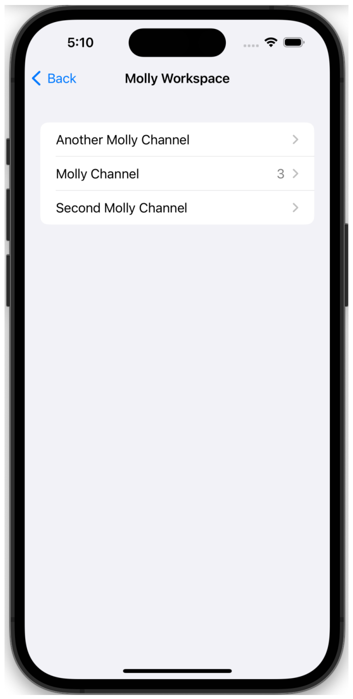
Navigation hierarchy is show on the following page.



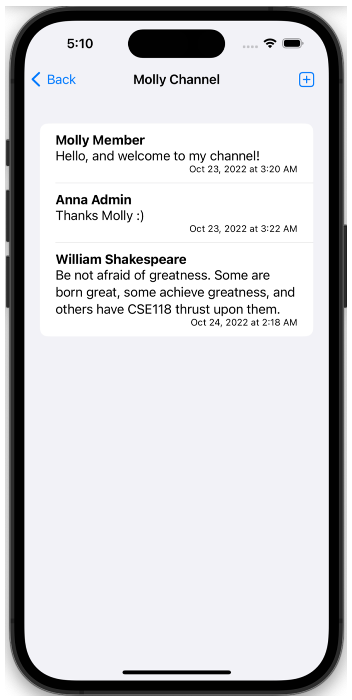
Login



Workspace List



Channel List

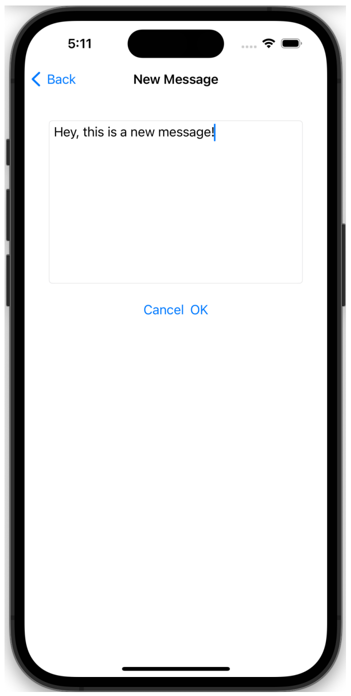


Molly Member
Hello, and welcome to my channel!
Oct 23, 2022 at 3:20 AM

Anna Admin
Thanks Molly :)
Oct 23, 2022 at 3:22 AM

William Shakespeare
Be not afraid of greatness. Some are born great, some achieve greatness, and others have CSE118 thrust upon them.
Oct 24, 2022 at 2:18 AM

Message List



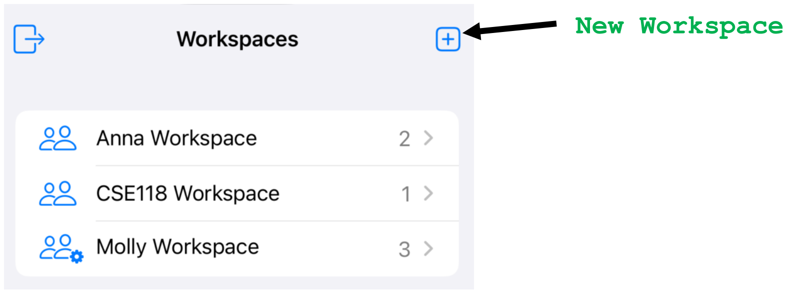
New Message

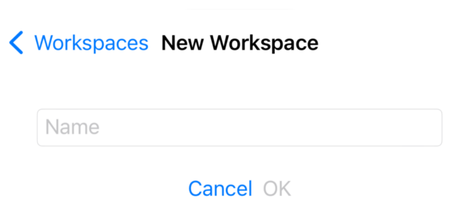
Advanced:

Enhance the App's functionality to allow the following operations:

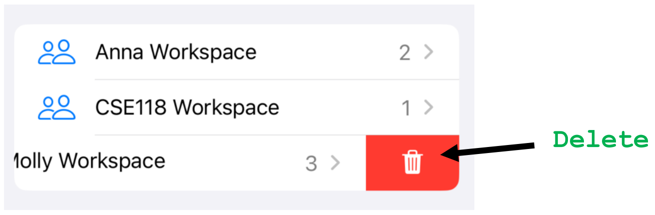
Accessibility Labels not obvious from displayed text are show in `Green Courier New`

- Create Workspaces

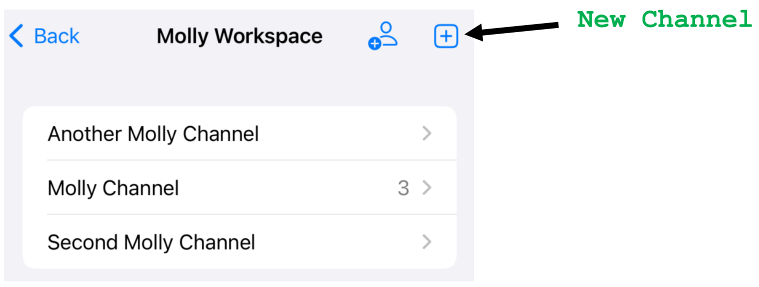


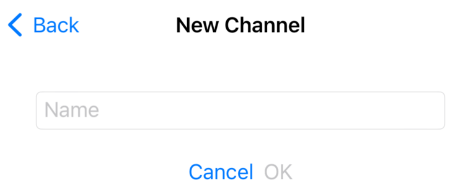


- Delete Workspaces

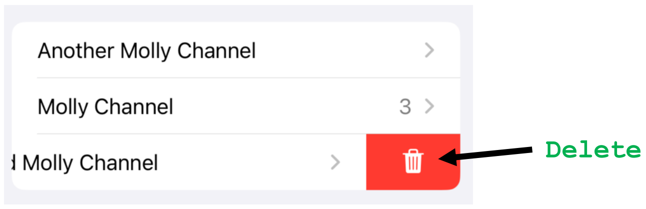


- Create Channels in Owned Workspaces





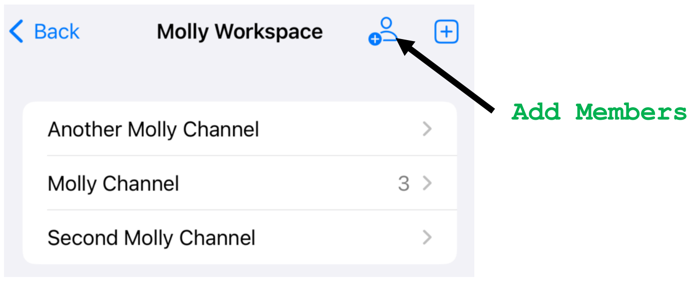
- Delete Channels


Another Molly Channel

Molly Channel 3

Molly Channel

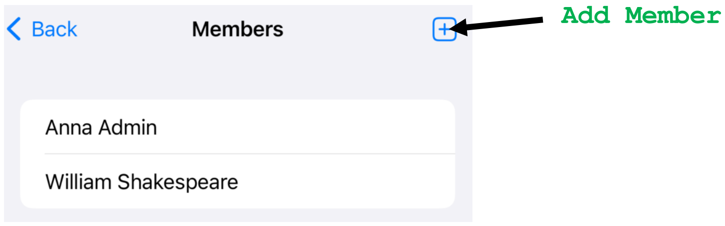
- Add members to Owned Workspaces



Another Molly Channel

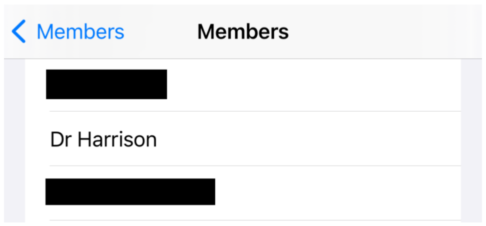
Molly Channel 3

Second Molly Channel



Anna Admin

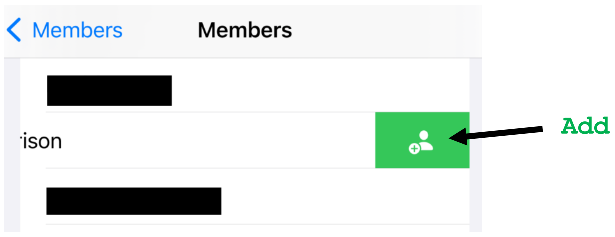
William Shakespeare



[redacted]

Dr Harrison

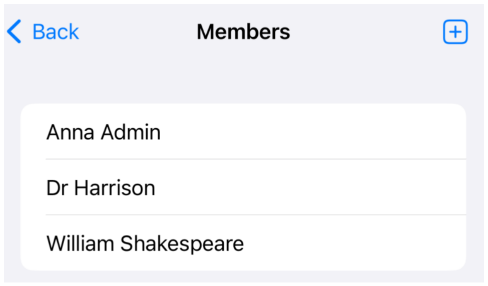
[redacted]



[redacted]

ison

[redacted]

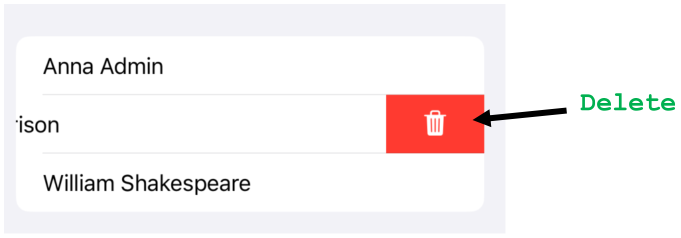


Anna Admin

Dr Harrison

William Shakespeare

- Remove members from Owned Workspaces



Stretch:

Your App should exhibit 100% class, method, line, and branch coverage when the provided Basic tests and your Advanced tests (if any) are executed.

Note you must write your Advanced tests in `AdvancedUITests.swift`, if you add additional test files your solution may fail to compile in the grading system

What steps should you take to tackle this?

Start by defining serializable Structures for the Login Credentials, Login Response and Workspace data types defined by the RESTful API.

Next, create a Login View to enter the email and password, use molly@cse118.com with password “molly”.

Once you can successfully log in, write a Workspace List View to list the Workspaces Moly either owns or is a member of. i.e. Workspaces owned by other users who have given Molly access.

Next, create Channel List and Message List views, finally adding a New Message View where the authenticated user can create a message.

Once you are passing all the Basic tests with 100% code coverage, move on to the Advanced requirement writing tests as you go.

Note that completing the Basic requirement with 100% code coverage is worth 60% on this assignment. Take care when implementing Advanced functionality to pay close attention to your coverage.

How much code will you need to write?

A model solution that satisfies all requirements has approximately 1,300 lines of code, including tests for the Advanced requirement.

Grading scheme

The following aspects will be assessed:

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

a.	Basic	(40%)
b.	Advanced	(40%)
c.	Stretch	(20%)

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:
```
$ Compress-Archive -Path . -DestinationPath Assignment6.zip
```
Linux:
```
$ zip -r Assignment6.zip *
```
Mac:
```
$ zip -x "*.DS_Store" -r Assignment6.zip *
```

**** UPLOAD** `Assignment6.zip` **TO THE CANVAS ASSIGNMENT AND SUBMIT ****