CSE118 Spring 2023

Assignment 9

In this assignment you will develop a simple React Native App based on components developed in Kotlin for Assignment 3 and Swift/SwiftUI for Assignment 6 to cement your understanding of:

- JavaScript
- React and JSX
- React Native Navigation
- React Native Swipe gestures
- Accessing RESTful APIs using `fetch`
- Background operations with `async` and `await`
- Automated testing using Jest and React Native Testing Library

This assignment is worth 10% of your final grade.

Late submissions will not be graded.

Configuration

As for Assignment 7. Make sure you are using Node.js version 18 or higher.

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.

- In Android Studio, start the Pixel 5 API 32 AVD

- In a terminal (command prompt on Windows) navigate to the installation folder and run the following command which will take a while to complete:

    ```
    $ npm install
    ```

- Then run the following command to start the react-native development server:

    ```
    $ npx start
    ```

- No wopen another terminal (command prompt on Windows) navigate to the installation folder again and enter the following command:

    ```
    $ npx run android
    ```

    The app should appear in the Android Simulator, and you can start constructing your solution by adding JSX components in the `src` folder.

- To run the tests and check code coverage:

    ```
    $ npm test
    ```

- To run check code quality and fix any trivial errors:

    ```
    $ npm run lint -- --fix
    ```
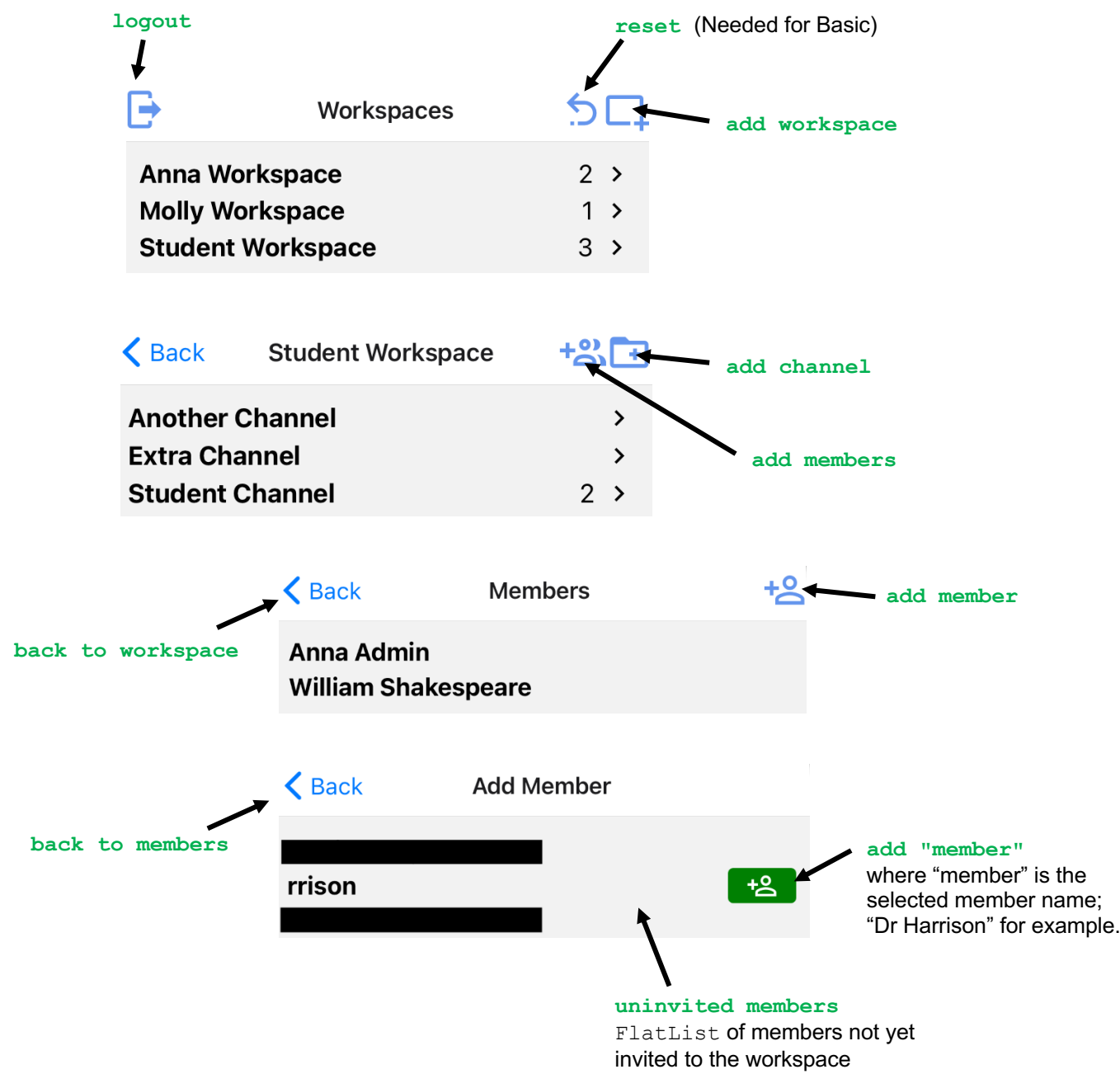
Requirements

Basic:

As for Assignment 6. Study the Basic tests for this assignment to see what the accessibility labels for the navigation-back and add-message buttons should be.
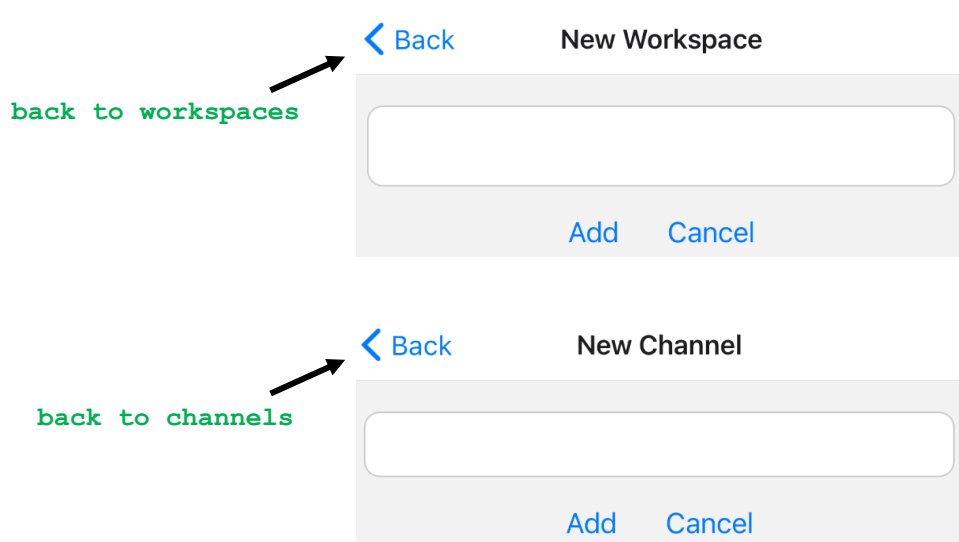
Advanced:

As for Assignment 6. Take care to honor the required accessibility labels for adding Workspaces and Channels and Members to Channels:

`logout`

`reset` (Needed for Basic)

Workspaces

`add workspace`

Anna Workspace	2 >
Molly Workspace	1 >
Student Workspace	3 >

‹ Back   Student Workspace

`add channel`

`add members`

Another Channel	>
Extra Channel	>
Student Channel	2 >

‹ Back   Members

`add member`

Anna Admin
William Shakespeare

`back to workspace`

‹ Back   Add Member

`back to members`

rrison

`add "member"`
where "member" is the selected member name; "Dr Harrison" for example.

`uninvited members`
`FlatList` of members not yet invited to the workspace

For "swipe-to-delete", accessibility labels should be:

Workspaces:	`delete "workspace"`	where "workspace" is the selected workspace name
Channels:	`delete "channel"`	where "channel" is the selected channel name
Messages:	`delete message`	

And for back-navigation when adding Workspaces and Channels:

Accessibility labels for swipe-deleting Workspaces and Channels are the same as for Messages, as are the text on Add and Cancel buttons, consult the Basic tests for details.

Stretch:

Your App should exhibit ***no linter errors*** and 100% class, method, line, and branch coverage when the provided Basic tests and your Advanced tests (if any) are executed.

What steps should you take to tackle this?

Use your understanding of what worked well and what didn't in Assignment 8 as the basis of the steps you will take here. Certainly, get the basic requirement out of the way quickly and submit to Canvas before starting work on Advanced.

How much code will you need to write?

A model solution that satisfies all requirements has approximately 1,000 lines of code including style definitions but not 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:

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

What to submit

In a terminal (Command Prompt on Windows), navigate to the folder you extracted the starter code into and have been working in then run the following command to create the submission archive:

```
$ npm run zip
```

** UPLOAD `Assignment9.Submission.zip` **TO THE CANVAS ASSIGNMENT AND SUBMIT **