

## CIS\*4030 – Mobile Computer – Exercise Thirteen: Local Data Storage

For this week's exercise you will be storing data locally for the recipe application to both store user theme preferences, and recipe additions and modifications.

**Weight:** There will be two options for weighting for this exercise: 1) you can choose to only finish the first modification (storing light vs dark mode preferences) or 2) fully finishing the application. If you choose the 1<sup>st</sup> option this exercise will be worth 4% and for the second option due to the increase work to the recipe app this exercise will be worth a total of 12% (or equivalent to exercises 13, 14, and 15). This exercise is due during class time with a two week window for submission. For this exercise since it will officially be assigned on March 12, 2024, you have until April 3, 2024 to submit this exercise.

In this week's exercise you will be modifying the recipe application to store data locally in two ways: 1) storing user theme preferences for the application with Shared Preferences (for Flutter and Android) or NSUserDefaults (Note: for those using React Native you will need to find an equivalent library to store key-value pairs persistently across your application), and 2) storing modifications and new recipes via a local data base (e.g., SQLite). Below I provide further details and requirements for each.

- 1) Storing User Application Preferences: You will be editing the settings page to add a toggle for theme preference for options between light (regular) mode and dark mode. This should change the all of the background colour, the text colours, icon colours, header or app bar colours, and the status bar colour (i.e., the wifi info, time, and other defaults provided by the system). Next, these values must be stored and loaded using some library to store key-value pairs persistently. Meaning that your application should still record changes even after closing or restarting the application.
- 2) Adding New Recipes: You will add a new page to enable the creation of a new recipes. This new page should allow for user input for all the required components of a recipe (everything from the Json file). Below are some further requirements for the new recipe page:
  - a. A floating action button must be used in order to get to the new recipe page.
  - b. There must be error checking so that no incomplete or invalid data can be entered.
  - c. The various recipe options (cooking time, ingredient cost, and recipe difficulty) must have radio buttons or single select options for all options available.
  - d. When a recipe has been successfully created you must show a toast bar to indicate that a recipe has been created.
  - e. The recipe must be stored in a local database, meaning if you close the application this recipe must still exist.

- 3) Modification of Existing Recipes: Using the same screen to add new recipes you will provide the option to modify existing recipes. To edit a recipe do this by have a option after long pressing on a specific recipe from the home page. Next, these modifications must still exist once the recipe has been close and reopened.

### **Submission**

To submit this exercise, you will need to show me the app running on a simulator during class. You must illustrate that you can switch from light to dark mode in the settings and that it will appropriately adjusting the colour scheme across the application. Next, you must be able to close the application and the light vs dark mode setting must be preserved once reopening the application. To illustrate the addition of a new recipe you must show me that the app doesn't enable the creation of invalid recipes and that the app allows you to create and display a new recipe. Next, you must show me that you can modify and recipe (you can modify the recipes in any way to illustrate this part of the exercise). Finally, you must close and reopen the application to show that both the new recipe and the modified recipes are stored locally on the app.