

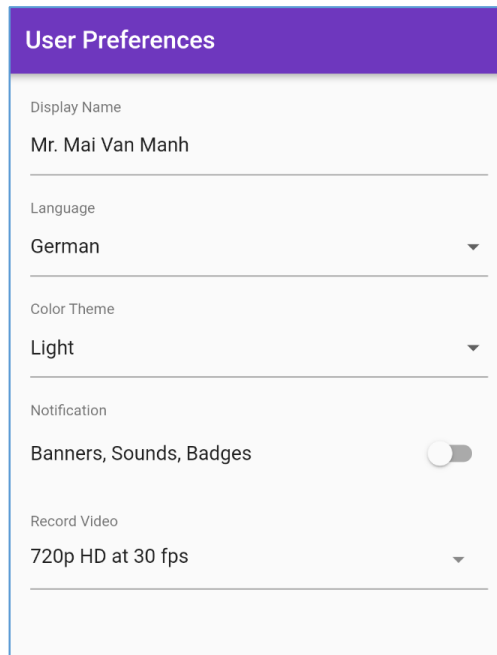
CROSS-PLATFORM MOBILE APP DEVELOPMENT

(503107)

LAB 8

EXERCISE 1

Based on the given source code, implement the functionality to store user settings using the [shared_preferences](#) package. The application allows users to set the settings information in various ways such as switch, dropdown button, alert dialog and modal bottom sheet. Make sure you save the data as soon as the user makes a change. If done correctly, the user's settings data will be displayed the next time the application is opened.

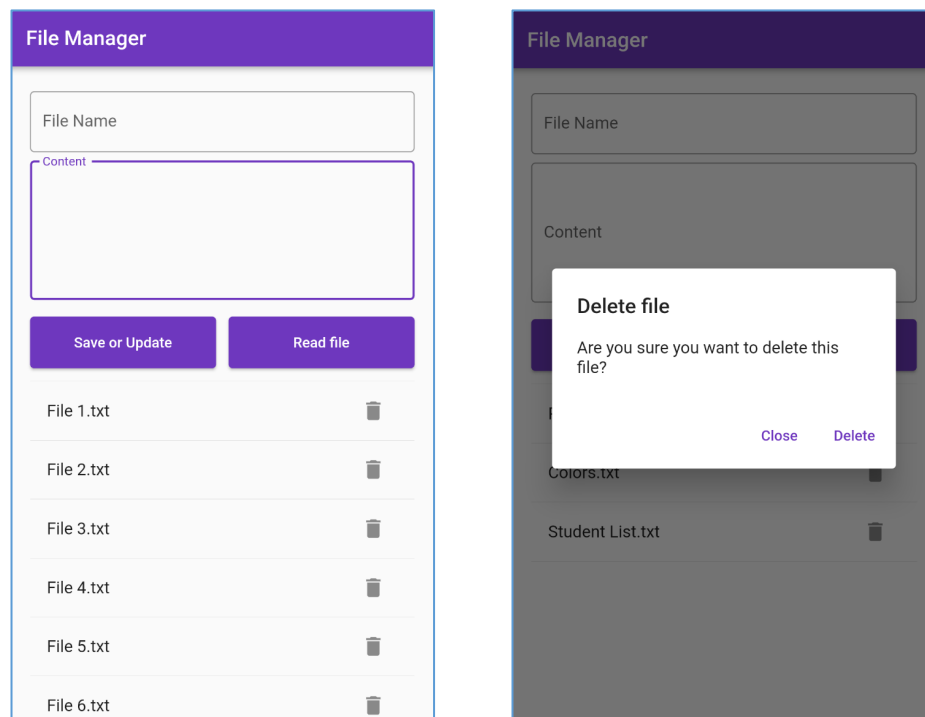


The image shows a mobile app interface for 'User Preferences'. It has a purple header with the title 'User Preferences'. Below the header, there are five settings sections, each with a title, a value, and a horizontal line separator. 1. 'Display Name' with the value 'Mr. Mai Van Manh'. 2. 'Language' with a dropdown menu showing 'German'. 3. 'Color Theme' with a dropdown menu showing 'Light'. 4. 'Notification' with a toggle switch for 'Banners, Sounds, Badges', which is currently turned off. 5. 'Record Video' with a dropdown menu showing '720p HD at 30 fps'.

EXERCISE 2

Write a basic file manager application that allows the user to enter a file name, file content and then save it to the directory returned by the `getApplicationDocumentsDirectory()` method from the `path_provider` library.

When the save button is pressed, if the file does not exist, a new file will be created. If the file already exists, the contents will be overwritten. To read an existing file, the user enters the filename and presses the Read file button.



Every time a file is created, it will also be displayed in the listview below. After successfully creating a file, the contents of the textboxes will be cleared, the filename textbox will be automatically focused to be ready for a new file to be added.

Users can also click on the file name to be able to view the file content quickly. When you click on the delete button next to it, the file will be deleted.

If errors occur (such as when reading a file that does not exist) then the application should display the appropriate message (using snackbar or alert dialog).

EXERCISE 3

Continue working on the note management exercise from last week's lab. Add dialogs to make the application more interactive. Let's apply SQLite to storing the list of notes instead of just storing the notes in memory as at the moment.

