Faculty of Science

Course:	CSCI 4100U: Mobile Devices
Lab Assignment:	#6
Topic:	Storage

Overview

In this lab, you will develop an application that saves its data to a file. You will need to create a new Android application. You can use Android Studio's wizard, using a Blank Activity to start.

Detailed Instructions

ShowContacts Activity
For this application, your main activity (ShowContacts) will maintain a list of contacts. For this purpose, create a simple class (Contact) which contains the relevant fields (firstName, lastName, phone). If you want an easy-to-modify list data structure, look into the ArrayList class. Below is a simple introduction on how to use it:

```
ArrayList<Contact> contactList = new ArrayList<Contact>();
Contact newContact = new Contact(1, "Bob", "Smith", "555-1234");
contactList.add(newContact);
contactList.remove(newContact);
```

The ShowContacts activity should create a layout that displays a ListView and two buttons: Add and Delete. The ListView will display the contacts of your application in the format (ID, Lastname, Firstname Phone#). When the Add button is clicked, you will display the AddContact activity, as a sub-activity. The data returned by this activity will be used to add a new element to the list of contacts. When the Delete button is clicked, the DeleteContact activity will be displayed, as a sub-activity. The data returned by this activity will be used to delete a contact from the list.

The ShowContacts activity will ensure that this application keeps its state from one execution to the next. It will do so by storing the contact data into a file, called 'contactData.txt' in the onStop() lifecycle method, and retrieve those contacts again from the same file in onStart(). The data will be stored with each contact occupying its own line, with each field separated by a single space. Do not use spaces in any of the field values when testing. Test that it works by exiting the application, and restarting.

AddContact Activity
The AddContact activity displays a form containing one text field (and a label) for each field in the Contact class, as well as an 'Add' button. When clicked, this button collects this data, attaches it to the result intent, and finishes the activity.

DeleteContact Activity
The DeleteContact activity displays a spinner containing all contacts, and a button 'Delete'. When clicked, this button determines which contact was selected from the spinner, and attaches the corresponding ID for that contact to the result, then finishes the activity.

Reading data from a file:

To read data from a file, consider using the Scanner class in Java. This class provides a nice way to pull out data from a text stream. For example, if the String variable 'line' contained one line of the text file, the following code (despite having no error checking) would collect the separate fields:

```
String line = …;
Scanner scanner = new Scanner(line);
int id = scanner.nextInt();
String firstName = scanner.next();
String lastName = scanner.next();
String phone = scanner.next();
```