**Concordia University**

**COEN/ELEC 390**

**Winter 2025**

**Technical Assignment 2**

**Deadline:** March 4**,** 2025 @ 11:55 pm

**Late Submission:** -20% for every 1 day late after due date

WARNING

You should do this assignment entirely by yourself, individually, using only materials provided in this document and by the Teaching assistants.

Objective

Design and implement an android mobile application with description given below. Using SQLite to save data and read data in an MVC structure with the use of DialogFragment. By the end of the assignment, you will end up with a simple application to manage a database of simple profiles for users, with a profile consultation history feature.

Application Description

Two Activities: mainActivity, and profileActivity. The mainActivity has an action button which opens a DialogFragment allowing the operator to add user profiles to the database. The mainActivity displays the list of all user profiles, showing only the profile name. Each list item is clickable, clicking on a list item opens the profileActivity and display the details of the user profile entry clicked. Because the data in the profiles may be sensitive, the profileActivity also displays a list of all previous time & date where the profile was opened.

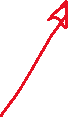
Approximate Mockup of app Navigation

Graphical user interface, application, email

Description automatically generated Graphical user interface, application

Description automatically generated Table

Description automatically generated



Requirements

General

* Two Activities: mainActivity and profileActivity.
  + profileActivity is a child Activity to mainActivity (up navigation is provided).
* A DialogFragment to add profiles to the database
  + Opened from the mainActivity using the Floating Action Button

MainActivity

* mainActivity has a textview at the top which displays:
  + the total number of profiles stored in the database.
  + how the database entries are displayed in the listview (by ID (increasing order) or by Name (alphabetical by Surname))
* mainActivity has a Floating button that opens an “Insert Profile” DialogFragment.
* mainActivity has a ListView that displays all the profiles stored in the database.
  + Each line corresponds to one profile
  + Each line shows either the profile name or the profile ID
  + Each line starts with the line number of the entry in the listview
  + Every item in the list is clickable. When an item is clicked, go to profileActivity.
* The mainActivity will have a toolbar with an action to toggle between two modes:
  + **Profile name Display mode:**
    - Default display mode
    - Each listview entry is a line number and the profile “Surname, name”.
    - The listview entries will be ordered alphabetically using the Surname first.
    - display mode can be toggled to “By ID” using an action in the action bar.
  + **Profile ID Display mode:**
    - display mode can be toggled using an action in the action bar.
    - Each listview entry is a line number and the profile ID number.
    - The listview entries will be ordered in increasing order using the ID number.
    - display mode can be toggled to “By Name” using an action in the action bar.

InsertProfile DialogFragment

* Insert Profile Dialog Fragment has:
  + 2 EditTexts for profile Name and Surname (2 strings, no special characters).
  + 1 EditText for profile ID# (8-digit integer, 10000000 to 99999999, **no duplicate ID# allowed!**)
  + 1 EditText for profile GPA (1 float number, 0 to 4.3)
  + 2 buttons for save profile and cancel.
  + Save button:
    - saves the input of the edit texts as a new profile in the database
    - returns to main activity and reloads the listview.
    - Does not save if some fields are empty or information entered is invalid.
    - A toast notifies the user about missing/invalid entries.
  + Cancel button: closes the dialog and returns to main activity.
* See Dialog Fragment example from Tutorial 3-4 for an example

ProfileActivity

* profileActivity has:
  + TextView(s) that display all the information of the profile: (Must show the profile corresponding to the entry clicked from the listview in the previous activity).
    - Name
    - Surname
    - ID#
    - GPA
  + Textview header to the Access History listview.
  + ListView displaying the timestamps, formatted as **yyyy-mm-dd @ hh:mm:ss**, indicating when the profile was created, opened, closed & deleted.
  + Delete button which deletes the profile that is currently opened in the profileActivity then go back to the mainActivity and reloads the listview of the profile.
  + A toolbar with an Up navigation to the mainActivity
* When the profile is created, a “created” entry is added to the access table.
* Each time a profile is opened from the mainActivity, a new access “open” entry is added to the access table.
* Each time a profile is closed, a new access “close” entry is added to the access table.
* Deleting a profile does not delete the access entries in the database, it adds a “profile deleted” entry with a timestamp.

DatabaseHelper

* DatabaseHelper provides:
  + Creation of an SQLite database with two tables
  + Helper functions to
    - Add entries in both tables
    - Read entries in both tables
    - Delete entries in only the Profile table
  + All add/read functions should return Access or Profile objects
* You must create the Access and Profile Classes to use their objects with the helper.
  + Use Setters/Getters and an appropriate constructor function
  + See Course.java class file from Tutorial 3-4 for an example

Things to help you with the assignment.

Your database will have the following tables:

**Profile table:**

* ProfileID **primary key**
* Name
* Surname
* Profile GPA
* Profile creation date

**Access table:**

* AccessID **primary key**
* Profile ID number
* Access type (created, opened, closed, deleted)
* Timestamp (time & date)

To select all the Access entries for one profile, you need to select Access entries where **Profile ID** in table “Access” is equal to the **Profile ID** from “Profile” table.

When creating Access entries for a profile in the Access table, you need to insert them with a **Profile ID** corresponding to the profile in the Profile Table, as follows.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Profile Table** | | | |  | **Access Table** | | | |
| **Profile ID** | **Surname** | **Name** | […] |  | **AccessID** | **Profile ID#** | **Access Type** | **Timestamp** |
| 10000001 | Asimov | Isaac | […] |  | 1 | 10000001 | created | 2021-06-06 @ 16:13:15 |
| 20000002 | Burgess | Anthony | […] |  | 2 | 20000002 | opened | 2021-06-09 @ 04:25:32 |
| […] | […] | […] | […] |  | 3 | 10000001 | opened | 2021-06-09 @ 04:29:37 |

In the above example, the first and third row in the Access table belong to the first entry in the Profile table. The second row in the Access table belong to the second entry in the profile table, because of the corresponding **Profile ID**.

Assignment submission and procedure

**Original work**

**This is an individual assignment. Your submission must be your own work.**

**You may use tutorial and online sources to understand, but you must write your own code. Do not directly use code (copy/paste) from internet sites, or from any other individual. As a rule of thumb, avoid having both IDE and code references open side by side. Look at the reference, identify the important elements and switch to the IDE to work on your code.**

You must submit your assignment before midnight on the due date using moodle Assignment Submission **in the submission link tagged with the tutorial section you are registered in (very important, a wrong submission might be considered a late submission)**. The file submitted **must be a .zip** **(no .rar)** file named **StudentID\_Ass1** containing **the entire android project folder**.

**Before submitting your code make sure you clean the project OR use the following project export procedure (the export procedure seems better than cleaning the projet).**

**Points may be deducted for uncleaned/improperly exported projects.**

**Use this to clean a project:** Android Studio --> Build --> Clean

**Use this to export a project:** Android Studio --> File --> Export -> Export to Zip File

Graphical user interface

Description automatically generated

**Evaluation criteria and grading scheme**

Meeting the requirements and use cases 80%

Using MVC design 15%

Clean code: well commented, proper naming, easy to read and understand. 5%

Possible bonus for quality UI design (clean, looks good/professional).

**If the project submitted does not compile and run the student will receive a grade of 0! So, make sure even if the assignment is not completely done that you submit an application that can be built and run. We will not grade none compiling code.**

**Also make sure to uninstall your project from the emulator and test it as a fresh install with no prior data. Some crashes only happen on fresh installs of the app.**