Mobile Applications Development - 300960
Spring 2021: Assignment 1
Student Management App

Assignment deadline: Friday 10/09, 11:59pm on vUWS. You must demonstrate your assignment at the following usual tutorial in week 10

Assignment Details

For this assignment, you are required to create a simple Student Management on an Android App. The app includes a home screen that includes one or more decorative images and options (e.g. button, menu items, etc.) The App has four main functions aside from the home screen:

1. Allow user to view and manipulate students
2. Allow user to view and manipulate to-do-tasks
3. Allow user to view and manipulate exams associated to a student
4. Allow user to view and manipulate images that are associated with students

In particular, the App should

Pass level (up to 64%):

- Provide the main interface and corresponding actions where user can select an option from buttons including
  - Add a new student record: the user can a student into the list. The student information includes *student Id* (unique number for each student), *first name*, *last name*, *gender*, *course study*, *age* and *address*.
  - Edit a student record: the user can select a student from the list. The program will display all information of the student in the corresponding fields where the information can be edited. The page should also include an option to delete the current record in addition to the edit option.
- Provide the main interface and implemented actions where the user can select an option from the buttons/menus for manipulating the to-do-tasks including
  - Add a new to-do-task: user can add a task in the list. The to-do-task record includes *task name, location,* and *status* (completed and not completed). The status is initially default as not completed.
  - Update a to-do-task: the user can select a to-do-task from the list. The user can switch the to-do-task to either "complete" or "not complete".
  - View all the to-do-tasks that are grouped into two categories: "complete" and "not complete".
- Be able to return to the home page or the previous page.
- Provide an acceptable quality user interface - Are the screens easy to use? Are they laid out neatly? Do they look good?

Credit level (up to 74%):

Include all functions of the pass level, plus

- Store data in a local SQLLite database, and implement INSERT, UPDATE and SELECT on the database for both student records and to-do-tasks,
- Provide an interface and corresponding actions where user can select an image and assign the image to a student. The images can be manually stored within the app itself. The app should be able to display the assigned image when showing the student information.

Distinction level (up to 84%):
Include all functions of the credit level, plus

- Provide the main interface and implemented actions to view and manipulate exams associated with a student. When the user selects a student, in addition to the edit and the delete options, the app also includes an option for the exam manipulation. The exam manipulation can be shown on the following page, including
    - Add a new exam: the user can add an exam to the list. The exam record includes exam name, date and time, location.
    - View all the exam record, including current exams and past exams. Note: the program should check the current date/time to flag the exams.
    - Use checkboxes to delete multiple exams.

High Distinction level (up to 100%):
Include all functions of the distinction level, plus the following features

- Provide a function to get one or more photos from the camera photo gallery, and add them to the photo gallery in the app. The user can also remove one or more photos from the photo gallery.
- Provide a function to show a student's address on the map. Note: the address is included with the student record.
- Have excellent coding styles with Object Oriented structure and comprehensive comments, and screen displays should be of professional quality.

Any figures beyond the requirements are welcome and they might be considered as extra contributions.

Deliverables

You are only allowed to use Android Java to code your solution. Your program must be executable. It is an advice that you need to keep multiple versions in case of unforeseen problems. You are allowed to demonstrate your program on your laptop. You might modify the code from related source(s) with a proper citation(s) and you must be able to explain clearly your understanding. The external code should contribute less than 30% of the total program. No part of the code can be written by any other persons.

Declaration

There is no requirement for documentation. However, you are required to submit a declaration with the following claim (in a text file or world file).

DECLARATION

I hold a copy of this assignment that I can produce if the original is lost or damaged.

I hereby certify that no part of this assignment/product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.

No part of this assignment/product has been written/produced for me by another person except where such collaboration has been authorised by the subject lecturer/tutor concerned.

Submission

Both the declaration and source code should be submitted via vUWS before the deadline for documentation/checking purpose. All these files should be zipped into one file with your student id as the zipped file name. Submission that does not follow the format is not acceptable. No hard copy of the source code and email submission is acceptable.

Demonstration

You are required to demonstrate your program during your scheduled practical session in the following teaching week (or based on the advice of the lecturer/tutor). Your tutor will check your code and your understanding of the code. You will receive no marks if you fail the demonstration, especially if you miss the demo time without a special consideration. To prove that the program is written by yourself, the tutor might require you to make some minor modifications. In this case, you must do the requirement in order to prove that the code is only written by yourself. You are allowed to run your program from your laptop at the demonstration time. The feedback to your work will be delivered orally during the demonstration. No further feedback or comments are given afterward.

The program you demonstrate should be the same as the one you submit. If you fail this assignment at your first demonstration, you are allowed to improve your work and give the demonstration within one week. The maximal grade is 50% in this case. You must indicate your intention to the tutor/lecturer if you want to take the opportunity to improve the assignment. Please do not send your work to the unit coordinator or your lecturer. Your tutor is responsible to mark your work.