Assignment 2: Desktop Application Development

(version 1 – 2022-11-15)

IMPORTANT: This assignment must be done individually.

Read **Section 1 & 2** to understand the assignment requirements and **Section 3** to know how to submit your work.

1. Description

As required from the stakeholder, you will change the CourseMan program from Assignment 1 to create CourseMan2 program with a graphical user interface using Java's Swing API. Similar to CourseMan, CourseMan2 still uses the `CourseManProg` class to coordinate other components and to start the program from the command line. In this assignment, you will implement the following features:

- the application main window
- the function to manage students
- the function to manage modules
- the function to manage enrolments
- the function to display an initial enrolment report & an assessment enrolment report

2. Requirements

2.1. Data requirements

The data requirements of this program are described in Assignment 1. The required data classes are: `Student`, `Module`, `CompulsoryModule`, `ElectiveModule`, `Enrolment`. In this assignment, the program must save the data objects to an SQLite database version 3 (see *Section 3* for the required name and location of this file). You shall create suitable database tables for the above data classes, as well as primary and foreign keys to model the data relationships in this program.

The number of tables and the design of tables may vary, but the database design must result in no data duplication and should be able to maintain data integrity.

2.2. The menu

The program's main GUI has a menu bar for the user to use. This menu bar has four menus:

- File – This menu has one menu item:
  - Exit – This menu item terminates the program.
- Student – This menu has two menu items:
  - New student – This menu item opens a window for user to add a new student.
  - List students – This menu item opens a window which displays a table of students.
- Module – This menu has two menu items:
  - New module – This menu item opens a window for user to add a new module.

- List modules – This menu item opens a window which displays a table of modules.

- Enrolment
  - New enrolment – This menu item opens a window for user to enrol a Student into a Module. There should be two drop-down boxes for selecting Student and Module respectively.
  - Initial report – This menu item opens a window which displays a tabular list of enrolments *without* grades.
  - Assessment report – This menu item opens a window which displays a tabular list of enrolments *with* grades.

2.3. The windows

1) The application's main window should be an empty frame with just the menu. You can put your own copyright text, user guides or your picture on this window.
2) The 'New student' window should allow user to enter student details through text fields (except for the student's id, which should be auto-generated).
3) The 'List students' window displays a table of students.
4) The 'New module' window should allow user to enter details about a module (except for the module's id, which should be auto-generated). There should be a drop-down box for user to select Compulsory or Elective module. Switching to Compulsory module will hide (or remove) the input for department name; switching to Elective module will show (or add) the department name input again.
5) The 'List modules' window displays a table of modules.
6) The 'New enrolment' window should have two drop-down boxes for user to select student and module, and text fields for user to enter internal mark and exam mark.
7) The 'Initial report' window displays a table of enrolments with these columns
   a. ID: the enrolment's integer id from database.
   b. Student ID
   c. Student name
   d. Module code
   e. Module name
8) The 'Assessment report' window displays a table of enrolments with these columns:
   a. ID: the enrolment's integer id from database.
   b. Student ID
   c. Module code
   d. Internal mark
   e. Exam mark
   f. Final grade

2.4. Other requirements

1. Update the `CourseManProg` class, which is the main program class, to start the program's main GUI. Don't forget to remove unnecessary parts in your code.
2. Preprare your database with these data:
   (a) At least 5 modules (3 compulsory and 2 elective) and 5 students
   (b) At least 10 enrolments (containing a mixture of compulsory and elective modules)

3. Submissions

1. You must create a top-level package named `a2_sid` for the program where `sid` is your student id.

   (*) *Your submission must contain only one package. Put all necessary classes in this package.*
2. You must store your database in the same folder where you store your source code, in other words, in the `a2_sid` package folder. The database file should be named `database.sqlite3`. As a result, its path should be `src/a2_sid/database.sqlite3`.

   Also store any images/pictures that you use in your program inside your package folder.

   (*) *This is not the recommended way in practice, but it is for the convenience of marking your assignment submission.*
3. You must submit a single `zip` file named `a2_sid.zip` containing the `a2_sid` package to LMS by the due date. For example, if your student id is `1912345678` then your zip file must be named `a2_1912345678.zip`.

IMPORTANT: failure to name the file as shown will result in reduced marks!

NO PLAGIARISM, strict penalty of marks reduction will be applied!