Coursework 3: Mini Project

Demonstration: Week 24, during scheduled class

Report: Friday of week 24, 7 April 2017

Assessment will be based on the following:

1. Demonstration of the Program Developed and answering questions – 70%

This part will be assessed based on **program functionality** (40%), **work done in class/attendance** (10%), **understanding** (20% of the work done, which will be assessed according to your answers asked by the lecturer when checking your program. You must receive a minimum of 8/20 in understanding to pass the coursework.)

2. Report Submission – 30%

You should use the following format to report your work (Please Submit Online):
The Coursework report should contain the following sections (out of 100, 30% contribution):

i. **Introduction** (5%)
ii. **Database Design** (10%)
    1. ERD
iii. **Software Development**
    - **Design** – 10%
        - Class diagram and explanation
    - **Implementation** (45%)
        - Show some important snippets of code (half a page max) and explain.
        - Object Oriented Concepts
        - UI Concepts
        - JDBC Concepts
iv. **Testing, Results, and Discussions** (15%)
v. **Conclusion and future work** – 10%
vi. **References (5%)**

The Task
You will need to develop a GUI-based Desktop Application composed of Components forming part of the `javaFX` Package which is able to connect a Database and make use of the JDBC API.

The Scenario
You are requested to write a Desktop Application that acts as a Room Booking (and potentially Timetabling) System for Middlesex University. The output should be tables showing information on rooms, bookings, programmes, lecturers etc.

You are not expected to develop code to calculate the best timetable.

The application should provide the following functionality.

1. A TCP based client-server system
2. A database system residing at server side. Processing should be done at server side.
3. A thin client for administration and accessing the server.

4. Administrators:
    a. Multiple admin login with synchronisation.
    b. Adding users (e.g. lecturers) who can only view but cannot change anything.
    c. Register/remove new Programs, Modules, and Rooms.
    d. Register new Admin & delete Admin.
    e. Show rooms booked/available
    f. Search for room booking based on the requester.
5. Users to view their bookings only.

You should decide what tables are needed. As guidance, you will need the following tables:

Programmes: A table of academic programs listing the modules taught on the program, the Department and the Faculty it is part of.

Modules: A table of modules with credit points, cps (15, 20, 30, 40, or 60), module leader, student numbers on each module and teaching/learning hours for lectures, seminars, laboratories, studio, and workshops, as appropriate. Faculty, and Department. There must be a total of one teaching/learning hour for each 10 cps.

Rooms: With number, type (lab, lecture theatre, seminar room, workshop, studio, and specialist), size, availability, module booked for, requester, booked by. For type you can use an integer (e.g. 1 for lecture room, 2 for seminar room, 3 for lecture/seminar & lab etc.)

Rooms Booked: Date and time of booking. Block booking possible. You will need this because a room can be booked for different time slots.

Users: There will be two types of users, one with admin powers and the other type to view his/her bookings only. Hence columns such as user ID, Name, Surname as well as a Boolean field to distinguish between administrators and ordinary users will be necessary. When an ordinary user accesses to the system, some fields should be disabled.

The specifications for tables are for guidance only; feel free to add more tables and/or columns, and merge tables if needed to.

Room, program, module information can be **specific to the Campus** you study at. However, be as realistic as possible.

Challenge: Anyone implementing one or more of the following will have 20% added to lab average mark. If lab average is more than 80% and mini is project fully implemented, than grade 1 will be guaranteed.

i. Adding optimal timetabling
ii. Replacing TCP communication with web sockets
iii. Adding user authentication using password hashing and salting.