CS 202 Project
Fall 2019

Deadline: 20 December 2019 Friday, 23:55 p.m.

In your project, you will develop a Web application called "**Hospital Management System**". In order to complete this project, you first need to research and find out the specifics of Hospital Management Systems. During your research, you need to think about the question "What actions can be taken in a Hospital System?" thoroughly. You need to determine the DB dynamics and mandatory actions. The list of some example actions are given below. You can extend your projects' coverage even further; but these specifications are a **MUST.**

The project consists of **two phases**.
In the first phase you will form your project groups, design and implement the database for this application. You have to form ***groups of 3*** for this project. Your report should include UML Diagram of your Database, necessary DML and DDL statements, and a brief explanation about your design decisions.
In the second part, you will implement the web application and its underlying components. Your application should connect and access to database.

The mandatory requirement of the hospital management system:

- ❏ There must be separate user types. *At least* 3 for patient, doctors and managers.
- ❏ Users can create a patient account from the home page with a sign-up functionality.
- ❏ Both medical staff and patients will login to the system with their own password in order to use the system.
- ❏ All passwords should be encrypted/hashed.
- ❏ Medical staff accounts must already be present in the DB or a DB Manager should only be able to add them - in other words, you cannot create a doctor account from the Web System.
- ❏ Doctors can list the availability of the rooms in the system
- ❏ Nurses are only able to view room availability, but nothing about patient information.
- ❏ Doctor's can specify the days or hours that they won't be available if and only if there are not any appointments during those periods.
- ❏ Both doctors and patients can see their past and upcoming appointments on separate pages and filter both of them can filter them in terms of days. (e.g show me my upcoming appointments in 10 days)
- ❏ When patients search for a doctor, they must have the ability to filter doctors in following parameters:
  - ❏ Field of expertise (i.e. Orthopedia, Cardiology)
  - ❏ Available in hour A in X day
  - ❏ Available anytime between hours A-B and days X-Y
- ❏ Patients can also filter their appointment information in terms of the area of expertise of the doctor which the appointments is made to.
- ❏ When a patient makes an appointment, that hour of that doctor must be available, in other words, your system **MUST** not let patients book an unavailable time.

❏ The patients can cancel their appointments if and only if there are more than 24 hours left.
❏ You date systematics must be implemented using datetime variable type.
❏ The user can list or view his all appointment information.
❏ Management should be able to retrieve data about the usage of facilities. Like;
  - Which department uses facilities the most
  - Patient statistics. Like
    - Number of patients on a time period
    - From which department they received medical care

Please note that, you should do your search on the hospital management system, and add new features to this system.

The specifications of the two phases are as follows:

DB design and implementation phase:

- Find out the information requirements of the Hospital Management System. Determine the constraints and domains. Determine the entities and relations.
- Identify the properties of the entities and their domains.
- Determine the identifier of each entity (i.e., find the primary key).
- Draw the Entity-Relationship (E-R) diagram.
- Decide what the base relations are.
- Draw the Functional Dependency diagrams.
- Make sure your relations are in 3NF.
- Decide what the referential integrity constraints are (identify foreign keys).
- Decide which deletion integrity rules to use (restrict, set to NULL, or cascade).
- Identify user views.
- Considering frequent access, come up with a physical database schema.
- Create the database on a DBMS.
- Specify DDL statements in SQL (internal, external, conceptual level).
- Specify the queries (needed for the transactions you determined in step 1) using SQL. Write SQL insertion, deletion, modification and select statements (DML statements).

Web application design and implementation phase:

- A simple web interface is adequate.
- Have an interface for both patients and medical staff
- You are recommended to use the following technologies:
  - ○ Java (JSP, and JDBC) + Apache Tomcat Server (for JSP)

Remarks:

• You should write your own SQL queries manually. You are not allowed to use a framework that automatically generates SQL statements and not allowed to use frameworks that might abstract the database interaction such as ORM frameworks. Some of them are listed here:
https://www.wikiwand.com/en/List_of_object-relational_mapping_software

• In addition, you should sanitize inputs to avoid SQL injection.

Submission:

- **Part 1**

  You will submit your Design Report on LMS by **8 December**. Your submission should only consist of a single .zip file which contains:

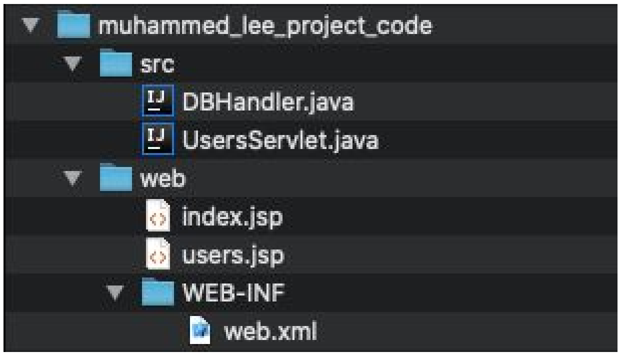
  - Your Design report in ***PDF*** format. Your file name should look like:
    muhammed_lee_project_design_report.pdf
  - Your DML SQL Scripts. Your file name should look like:
    muhammed_lee_project_dml.sql
  - Your DDL SQL Scripts. Your file name should look like:
    muhammed_lee_project_ddl.sql

  Your zip file name should look like:
  muhammed_lee_project_design.zip

- **Part 2**

  You will submit implementation of your project and associated files on LMS by **22 December**.

  - You will only submit the working version of your code and nothing else
  - You are allowed to only submit a single .zip file. No other file type is not accepted (i.e. rar)
  - Your .zip file should only contain project folders and these folders should be directly inside the .zip and not in a folder inside the .zip file.
  - **Do not** add directories that contain auto-generated compiler output folders like *bin*, *tmp* and *out.*
  - Your file layout may look like shown below when extracted; that is equivalent to "extract to folder", not "extract here".
    This does not mean that it has to look exactly like this ***as you will have more components!***



```
▼ muhammed_lee_project_code
    ▼ src
        DBHandler.java
        UsersServlet.java
    ▼ web
        index.jsp
        users.jsp
        ▼ WEB-INF
            web.xml
```

  - Before sending the .zip file, **make sure that it works when it is extracted** from the .zip file!

It is mandatory to attend presentation of your project.

DO NOT FORGET TO USE YOUR OWN NAME

Failing to comply with the guidelines ***will*** result in a **penalty**.