Project: Database Application Development

Due date: Last day of classes (in the assignment box and electronically),
Value: 10% of final marks.
Price: best project gets a copy of the text book.

In this assignment, you will apply much of the application-oriented material you learnt to create a small database-backed application. The focus of this project is to do conceptual and logical database design and to create a JDBC-based code that accesses an Oracle database.

Description of the domain — MicroHard Inc. needs to store information about its employees, departments, projects, coop-students that it hires, and children that depend on the employees. Consider the following information about MicroHard:

- Employees have an SIN, a name, a salary, an address, and a phone number.

- Projects have a project number, a start date, an end date, and a budget.

- Each project is managed by one employee who acts as the project leader. Employees can manage several projects.

- Each project has one or more coop-students working on it.

- One or more employees participate in each project as co-workers. Employees can participate in several projects.

- An employee may have another, more senior employee – an advisor – who advises him – an advisee – on various items within MicroHard Inc.

- Employees work in one or more departments, and for each one of the department they work in, there is a start date indicated.

- Departments have a number, a name, and are located in a building with a given name.

- Departments are run by an employee (head) since a certain date. Each department must have one such head, and no department can have more than two heads.

- Employees have children that depend on them. Children have a name and an age. Each child must be uniquely identified by his name only by using the information about one of his parents.

- Coop-students have an SIN, a name, an address, an age, a phone number, and a field of study (e.g. CSI, EE, SENG, etc).

- There is an employee who must mentor work done by a coop-student on a project. Coop-students can work on several projects, and must have a mentor for each one of them.

Based on the information given above, do the following.

1. **(10 points)** – Draw an ER diagram for the MicroHard database. Doing so, indicate the various attributes of each entity and relationship set; also specify the key and participation constraints for each relationship set. Specify any necessary weak entity sets, looping relationship sets, aggregations, and hierarchies. (Submit this on Oct 28)

2. **(10 points)** – Translate your ER diagram into a relational database schema by writing an Oracle SQL script. Use appropriate constraints in your SQL script. (Submit this on Oct 28)

3. **(60 points)** – Write JDBC code that performs database access. The interface to your system should offer a reasonable functionality:

   (a) A user can search for employees by their SINs, names, and phones. He can perform similar operations with departments, projects, and coop-students.

   (b) A user can run appropriate queries on the different tables that constitute the database schema. Find such appropriate queries and provide an interface for them.

   (c) A user logs in and should remain within the system at most 1 hour.

   (d) The interface to your system should be appealing.

   (Submit this on the last day of classes)

4. **(Optional – 3% to be added to the final grade if you do this)** – Furthermore, design a (very simple) web-based presentation tier using (static) HTML pages, and a middle tier using one of the technologies available: e.g. a (very simple) CGI script, Java servlets, JavaServer pages, or something else.

Some Resources: Links to JSP, Servlets, XML, JDBC, Oracle tutorial, HTML, etc:

```
http://www.cs.wisc.edu/~dbbook/openAccess/www_resources.html
http://www.java.sun.com/products/jdbc
http://www-db.stanford.edu/~ullman/fcdb/oracle.html
http://www.utoronto.ca/webdocs/HTMLdocs/Book/Book-3ed
```

To submit:

- Your ER diagram

- Your SQL script

- Your Java code

- Any further CGI code written or the like

- A printout of the output of the system that clearly shows that what you did works.