Databases Project

For this course your group will develop a database and an application that uses it to fulfill a set of needs. Your project will solve a problem, and you'll design it with such problem in mind. In order to implement your solution, you'll follow the following steps:

1. *Decide the group composition and the project's topic.* Your group will be made of 4 people (exceptionally 3) and your topic should have something to do with biomedical engineering.
2. *Define the project's features and draw a UML use case diagram.* The diagram must have at least 2 actors and needs to be accompanied by a brief text document (PDF) that explains your system's intent and list all its features.
3. *Draw an E-R diagram that captures the project's needs.* It must have at least 5 entities and 1 many to many relationships.
4. *Create the relational database's tables from the E-R diagram.* You can create the diagram in a spreadsheet, but later you'll need to write its DDL in SQL.
5. *Create the DDL for your relational database.* This includes any CREATE and INSERT needed for the initial setup of your database. Try your DDL in a real DBMS (using DB Browser is a good idea) and save it for future use.
6. *Define the items and flow of the UI of your project.* All projects must provide a user interface that implements all the features defined in the document and diagram created for step 2. You can create a text menu-based UI or a GUI. Whatever you choose, create a mockup or skeleton that provides all the desired functionality.
7. *Prepare and present your mid-semester presentation.* You'll prepare a 15 to 20-minute presentation that explains steps 1 to 6. Use PowerPoint.
8. *Create the DML that will be used by your project.* Each option in your UI that needs to access the database will do so through a SQL query (SELECT) or update (INSERT, UPDATE, DELETE). This will be your DML. For each feature (or UI option, they're both the same) write down and test a SQL query.
9. *Draw a UML class diagram of your application.* By now you should have all the needs and features of your application defined. You need to create a class diagram (with at least 2 packages and 1 interface for accessing the DBMS) that describes your application. Use this diagram to plan how you're going to tackle the next step, which is…
10. *Program!* Code all your classes. After finishing this step, you should have a working piece of software that fulfills all the needs stated in step 2… with one exception: the login subsystem. All programming must be done collaboratively using GitHub.
11. *Expand your UML class diagram to include the login subsystem.* Some of the new classes must use JPA and follow the ORM paradigm. This should be its own package.
12. *Program!* Code the new classes and integrate them with your existing application and database. By now, you should have a working application with different users and roles.
13. *Define the XML subsystem's DTD.* Your application must import/export at least 2 entities (from the diagram created in step 3) from/to XML. To do that, the first step is defining a schema for the input/output XML files using DTD.
14. *Define the XML subsystem's HTML interface.* Your application must be able to generate a webpage with the database's information for at least 2 entities (from the diagram created in step 3). To do that, the first step is defining a transformation for the XML files using XSLT.
15. *Expand your UML class diagram to include the XML subsystem.* Some of the new classes must JAXB and the provided code. This should be its own package.
16. *Program!* Code the new classes and integrate them with your existing application and database. The one thing missing is…
17. *Prepare and present your final presentation.* You'll prepare a 15 to 20-minute presentation that demos your project. Use PowerPoint. **Congratulations! You've finished!**

Database project checklist

Tick the box when you have finished the task! Tasks in italics are optional and raise your score.

Documentation

☐ Write a summary of your system's intent and a list of your system's features
☐ Prepare the mid-semester presentation
☐ Prepare the final presentation
☐ *Write a design document of your system, explaining why you did what*

Diagrams

☐ Draw an E-R diagram of your database (at least 5 entities and 1 many to many relationships!)
☐ Draw an UML diagram of your system(at least 4 packages and 1 interface!)

Structure

☐ Develop a database manager that uses JDBC
☐ Develop a database manager that uses JPA (for the login subsystem)
☐ Develop an XML manager using JAXB (for the XML subsystem)
☐ Components must interact via objects or primary keys
☐ Create a menu-based UI separated from your database manager or…
☐ *…create a GUI separated from your database manager*

JDBC

☐ Create all database tables
☐ Insert new data into at least 2 tables
☐ Show all elements of at least 2 tables
☐ Search by 1 column other than the primary key (of your choice) in at least 1 table
☐ Update data in at least 1 table
☐ Delete data from at least 1 table
☐ Insert data into at least 1 table with 1 or more foreign keys
☐ *Use binary objects (like images) in at least 1 table*

JPA

☐ Annotate at least 2 entity classes that share an association with JPA (for the login subsystem)
☐ Make at least 1 create operation using JPA
☐ Make at least 1 read (that involves search) operation using JPA
☐ Make at least 1 update operation using JPA
☐ Make at least 1 delete operation using JPA
☐ *Properly encrypt your users' passwords*
☐ *Use inheritance with JPA*

XML

☐ Create a DTD schema
☐ Use XSLT to produce a HTML, *extra points if it's cool and/or uses CSS2!*
☐ Marshall and unmarshall at least 2 entities using JAXB
☐ *Export/import your whole database to/from XML using JAXB*

Finishing touches

☐ Make sure your system fulfills the purpose stated in the intent document and squash all bugs
☐ Put your project (and documentation) in GitHub and send the link to your professor
☐ *Write a user's manual*