Databases - Course Project

Start Assignment

Due 18 Oct by 13:30   **Points** 25   **Submitting** a media recording or a file upload

---

Project Outline

For this course you will work on a project together with another student. This project is worth 25% of your final grade.

1. Join up with a fellow student, go to canvas and join a group. This is required for grading purposes. If you do not join a canvas group we cannot grade your project!

You will have until the last week of this course to work on the project. At which point you will deliver the following:

1. A **screen recording** presenting your software and database. In the video you must show the same source code that you upload to canvas. Give a 5 - 10 minute presentation of
   1. the choices you've made in terms of database API, programming paradigm and database system,
   2. your program's functionality,
   3. the database you created (you can present the database from the DBMS client
      If you do not want to, you don't have to talk in your video. In that case, make sure you use another way to explain what you are presenting.
2. An entity relation diagram of your database (even if you do not use a relational database)
3. A database schema of your database,
4. The source code of your system. This code will be checked for plagiarism.

The only constraints to your software are the following:

1. It must implement the functionality described below,
2. It must have a (simple) user interface, you may choose the type of UI (terminal/web/electron/javafx). Even a simple terminal UI will suffice,
3. You must use a **database system** to store the program's data. The database system must fit the requirements of the project. For instance, you **cannot use SQLite, because the described software can be used by multiple users at a time.**

You may use any programming language that the teachers can understand. If you are not sure; ask. You may use any type of database system, relational, document, graph, go wild, if you want to use some vague DBMS, make sure to discuss it with a teacher to see whether it will work. The project counts for **25% of your total grade** for this course.

The final result is based on:

1. (15 points) Did you implement all functionality while using a database as storage and data management system,
2. (5 points) Does your database schema match the requirements of the software and did you implement your schema as presented,
3. (5 points) Does your video clearly show all functionality and implementation.

For more details about how we will grade your project, take a look at the grading sheets posted in the modules.

The deadline for the project is: 18 Oct 13:30. After the deadline (in the last week) you will be asked during the lab to perform peer reviews for 3 other projects. This gives you the opportunity to study the problem from multiple angles and see how others have solved the problem.

The next labs

During the upcoming labs we'll keep practicing topics related to databases. But the labs will also act as *office hours* where you can claim time from the teacher to discuss the choices you made in the project.

Project Description

As you may remember from the software engineering course, the EU has banned centralized take-away ordering for pizza restaurants. This means that every Pizza restaurant in Europe will have to operate its own Pizza ordering service. Pizza restaurants are notoriously good in software architecture design as we all know. So, they decide that they want to separate their models and controllers from their views.

Now that you may have already written the controller during Software Engineering, we're going to add a model to the system while also expanding the system to include more functionality. Don't worry if you did not work on the assignment in my previous course, this new assignment does not rely on the old one. If you did work on it, see if you can expand upon what you built and complete the program. Creating a REST API is not part of this assignment, though it is a good solution in terms of architecture and design.

Software Requirements

First, we must be able to present a menu to the customer:

- List all the pizzas available for ordering at the restaurant.
  - Pizza prices are built up from their ingredients, which have prices and each pizza has a fixed 40% margin for profit.
  - This means that the price of each pizza is based on the *total sum of its ingredients * 1.4*
  - Even though you are probably used to being able to choose your own toppings at a pizza restaurant. For this project this is **not** a requirement. So the ingredients (i.e. toppings) of all pizza's are fixed, customers can only order predefined pizzas.
- For any given pizza show
  - all the ingredients,

- ○ its price (including 9% VAT) and,
  - ○ whether the pizza is vegetarian or not **based on its ingredients**.
- The menu also shows drinks and deserts.

Make sure to include at least 10 pizzas with at least 10 different ingredients, 4 drinks, and 2 deserts in your presentation. Also make sure that you show how you calculate the pizza prices.

Order processing:

- Allow the placement of orders for pizzas, drinks and deserts. Each order must include at least one pizza. But can include any number of products from the menu.
- Store customer information such as name, phone number and address, so we can contact them and deliver pizzas to their homes.
- Keep track of the number of pizzas a single client ordered. They receive a 10% discount on their next order after ordering 10 pizzas.
  - ○ Customers get a discount code that they can enter when ordering in this case
  - ○ You'll have to check for this code when ordering, it can be used only once!
- When a client orders a pizza, he/she should be sent a confirmation showing the products ordered and the estimated delivery time.
  - ○ After 10 minutes of preparation, the pizza goes out for delivery.

Order delivery:

- A client should be able to cancel their order if it was placed no more than 5 minutes ago. The restaurant then cancels the order and the pizza will not be delivered.
- Each restaurant has a number of delivery employees.
- Each delivery employee is assigned to the specific postal code. Multiple delivery persons may be assigned to a single postal code.
- A delivery person cannot deliver outside the area which is not assigned to them (for every delivery person there can only be single area assigned to them)
- When a pizza is ordered, that delivery employee is out for 30 minutes and cannot deliver another pizza. Until they return. you do not have to take any other kind of timing into account, they will always be available after 30 minutes.

These delivery timings do not make for efficient delivery, they merely show the minimum requirements. Suppose two clients are in the same postal order 1 minutes apart then we need two separate deliveries using the system described above.
Therefore, you are allowed to implement an improved system if you like to. Please do not work too much on creating an exact delivery system. Focus mainly on implementing the correct Database model. Making sure that the relations between tables match their intended use and allowing for ACID transactions and most importantly: **non-redundant storage.**

Even though you can use your Software Engineering project as a start for this project, this is 100% optional. You can create a new program from scratch if you'd like to. Also note that REST and API's are not part of the requirements of this project. If you want to keep using them, great! If not, also great!

To start:

1. Meet up with your fellow student. **Subscribe to a group on canvas**. **Do not create your own group**, join one of the pre-created groups in the **"People"** tab in Canvas.
2. Draw a preliminary ER diagram. Define all entities that you see in the description and how they relate. You can discuss this diagram with the teachers to make sure you start off on the right track.
3. Choose your "stack", which programming language and database system you will use. Investigate what kind of database API you'd like and how to use it. If you want to build on your work from Software Engineering you can continue with what you have created before.
4. If you have time, create the schema from your ER diagram. Define your classes and start programming

Make sure to plan ahead, split tasks if you'd like to. But also make sure you have a bit of fun creating this project.

Week 7:

The lab in week 7 is reserved for Peer-Feedback. After the deadline of this project, your group will be asked to review the work of 3 other groups. **This is mandatory to receive a grade, i.e. not posting your assigned peer-feedbacks will result in an NG for the project.** We will use the last lab to write the peer-feedback and discuss it with the other groups.