

IS503 – Project Description

For your term project, you will select a project topic, design a database, and prepare queries that would be used by the application for your topic. 1 to 3-person project groups are allowed. Your project topic must be chosen from below, but the information given related to these project topics are merely for inspirational use. You are free to enrich and completely change the properties of the project ideas given that you are able to come up with deliverables that are qualified and complex enough. A user interface is not needed, think as if you are only responsible with the database layer of an application. The aim of this project is not to create a working application but to teach you how to design an application which utilizes databases.

Project deliverables:

First phase of the Project (Deadline: 8 December 2019): Project proposal

Inform me about your group and the project title in ODTU-Class “Project Group Information”.

15 points: ER/EER Diagram

40 points: Schema Design and Implementation

15 points: Full Report (Deadline: **12 January 2020**)

30 points: Demonstration of the project (Deadline: **13 January 2020**)

The demo program will be announced later on ODTU-Class.

Project Proposal:

You should choose one of the project cases described below. The group info (group name if you have any, member names) and project topic (one of the topics below) should be decided and submitted in ODTU-Class.

You may prefer to enrich/revise the case studies below. However, your final report should include all the details of the project that you are implementing. If you do not submit your case study with these constraints in the final report, then we will not be able to evaluate your E-R design.

Case 1: A Ridesharing Application

Platforms such as **Uber**¹ help people get to their destination or earn some money by becoming drivers for hire. Registered users can state their location and destination, see the estimated fare, browse closest drivers that are available, choose them and hire them. The chosen driver drives them to their destination. In return, they get a percentage of what the user pays through the application. Customers and drivers can also rate each other that affects their ratings which can be seen by other people through the application. It is possible to split the fare between multiple users through the application or schedule rides beforehand as well. Rides can be cancelled, and their prices can dynamically change based on the demand and supply. Design a ridesharing application that would have similar features.

Case 2: A Ticketing System for Events

Ticketing systems such as **Biletix**² help people follow upcoming events and buy tickets for them. They can browse an event schedule, filter the events, read reviews of the past events (of the chosen musician, for example), read information related to the venue, look at the seating arrangements of venues, choose a section, and buy seats. These seats can have individual reviews (made by other users) that comment on their view angle quality and share an image taken from that seat. Seat prices can change based on the event, section, and even seat. If all seats are sold out, interested users can get in the queue that would be used to fill seats that were cancelled by users who had bought them. Design a ticketing system that would have similar features.

DELIVERABLES AND CONSTRAINTS IN THE PROJECTS:

ER/EER Diagram:

You should provide your project's ER/EER diagram. You are supposed to show entities, weak entities and relations with min-max constraints.

Schema Design and Functionality of the System:

You should provide the schema design (tables, primary and foreign keys). Your tables should have meaningful table and column names. Your queries should make sense and they should be queries that could be actually used by an actual application.

In your database, you are expected to work with **at least 4 tables**. **If your group constitutes 2 or 3 members, then a minimum of 8 and 12 tables should be utilized, respectively**. The number of tuples should be at least 60 in total (for all the tables).

The constraints of the project are as follows:

- One table per group member must have insert, delete, and update operations satisfying integrity constraints. For example, if your group consists of two members, you should perform these operations on two tables.
- For every member of the project team, develop two complex queries, one with aggregate functions, and one that involves multiple tables to be joined, preferably nested queries.
- Include a derived attribute, date or time information in your database design and perform at least one query comprising these values.

¹ <https://www.uber.com/us/en/ride/how-it-works/>

² <https://www.biletix.com/about/TURKIYE/tr>

- Error and exception handling (for example by using triggers and foreign key constraints) should be done and demonstrated at the demo phase.
- For each member of the project, a view that makes sense (it should be beneficial for analytics purposes and such) should be implemented.

Project presentation – Demo:

The following items will be checked:

- Insert, delete, update operations for the selected tables and whether integrity is enforced
- Derived attribute
- One aggregate query for each student in the team
- One complex join query for each student in the team
- One view operation for each student in the team
- Error checking

Report:

You should prepare a final report that includes all the deliverables.