Department of Computer Engineering

CS353 - Database Systems

Spring 2021

Term Project Proposal

Group 18

Teaching Assistant:

Arif Usta

Group Members:

Melisa Taşpınar	21803668	Section 2
Yiğit Gürses	21702746	Section 1
Mustafa Hakan Kara	21703317	Section 1
Furkan Demir	21802818	Section 3

Table of Contents

1. Introduction

Online food ordering and delivery platforms are websites and applications that allow their users to order food from nearby restaurants and have their orders delivered to their location, all through the Internet. Examples of such platforms include Yemeksepeti [1] and Uber Eats [2]. Platforms like these can have a high number of users, which results in the need to manage a big amount of possibly valuable data. This data can include important information about the customers such as name, address, phone, e-mail address; as well as information about each restaurant such as address, working hours, delivery range and menu. Therefore, such platforms tend to require a database system to be able to manage such big amounts of data.

In this report, we propose a web-based application that manages a food ordering and delivery system. We first give a brief description as to what the project is, and justify our decision of using a database system by stating why and how we use it. We then specify our project requirements, which includes functional requirements, non-functional requirements and constraints. We then move onto the limitations section where we specify the list of limiting assumptions we make. Thereafter, we provide an Entity Relationship Diagram

2. Project Description

[Insert Name] is a web-based application with the purpose of managing a food ordering and delivery system. In the system, there exist restaurants, their menus, the food and beverages offered in those menus, and certain actions such as rating and reviewing deliveries. The users of the system consist of restaurant owners, customers of the restaurants, and delivery guys working for the restaurants. Customers should be able to make an order from a restaurant of their choice, in which they are allowed to select any number of items from the restaurant's menu. Here, the system is supposed to check if the customers have enough credit to make their orders. If so, the order should be processed and a free driver is assigned for the task of delivery. After the delivery process is complete, i.e. the customers' orders are delivered to their locations, customers are expected to rate and review both the restaurant and the driver.

2.1. Why We Need a Database System

A database system can be used to handle data collections that are relatively large, quite valuable, and accessed by multiple users, mostly simultaneously [3].

A food ordering and delivery system needs to store the information of their users, the orders, registered restaurants, ratings and reviews. This data can accumulate and reach a quite large amount in a short amount of time. In addition, it can contain highly important data, such as the credit card information of their users. Furthermore, such a system can be accessed by many users at the same time and, therefore, must support concurrent transactions.

Therefore, using a database system instead of an arbitrary file system can be highly advantageous.

2.2. How We Use a Database System

Our database system will store information about the users such as name, address, phone, e-mails, etc. There are three types of users, customers, restaurant owners and delivery guys, the database will provide specific functions to the users according to their type. The database will have the information about restaurants and their menus. In addition, information about the functionalities such as giving review and ratings, creating a basket and placing an order will be stored and displayed by the database.

3. Project Requirements

3.1. Functional Requirements

3.1.1. Customer

Customers should be able to:

- See a list of restaurants in delivery range
- Filter restaurants by food types

- ➢ See restaurants' menus
- ➢ See reviews and ratings of restaurants
- ➢ Filter menu items based on item categories
- ➢ Add items with chosen options to the basket
- ➢ Confirm and pay for current basket to make an order
- ➢ Add a note to their orders
- ➢ See their order status (being prepared/on the way/cancelled etc.)
- ➢ Rate and review their orders
- ➢ Order the baskets they created in an earlier order.

3.1.2. Restaurant Owner

- ➢ See and confirm/cancel orders given to their restaurant
- ➢ See the items in the restaurant menu
- ➢ Create new items and add them to restaurant menu
- ➢ Modify items in the restaurant menu
- ➢ Modify item discounts in the menu
- ➢ Respond to the reviews on their restaurant
- ➢ Set restaurant working hours
- ➢ See and change restaurant "open" status

3.1.3. Delivery Person

- ➢ See orders assigned to them
- ➢ See their ratings
- ➢ Set working hours

3.2. Non-Functional Requirements

Following are the non-functional requirements we have set for our system. Burayı 319'daki gibi spesifik mi yazsam yoksa genel mi yazsam karar veremedim değiştirebilirsiniz, ayrıca requirement ekleyip çıkarabilirsiniz aklıma gelenleri yazdım.

3.2.1. Security

E-mail addresses and passwords will be crypted and should never expose.

3.2.2. Responsiveness

The system should perform and respond to user inputs as fast as possible, especially considering the possibility of receiving several user requests simultaneously.

3.2.3. Memory Capacity

The application might have a large number of users, and, therefore, the database system might need to store large amounts of information by nature.

3.2.4. User-Friendliness

The application should be user-friendly in the sense that its user-interface components should be understandable, clear and as self-explanatory as possible.

3.3. Constraints

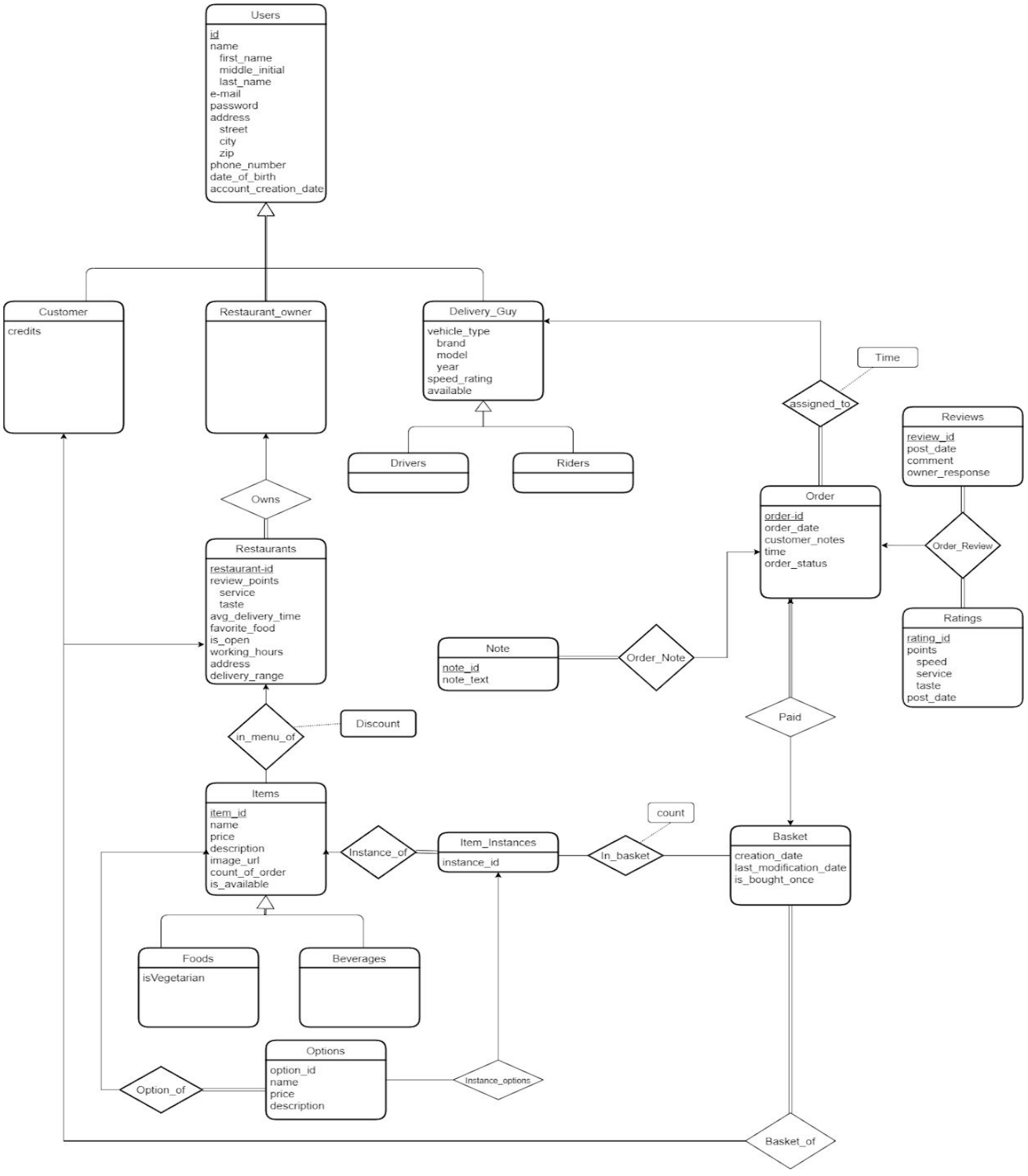
- MySQL tarzı seçimlerimiz

4. Limitations

- Users
  - A user must submit their credential information such as a name, an email. address, an address, a phone number, and a date of birth.
  - A user cannot submit more than one credential information.
- Customers
  - A customer can order from a one restaurant at a time
  - A customer can rate and review an order only once
  - A customer cannot cancel their order.
- Restaurant Owners
  - A restaurant owner cannot delete a user review
  - A restaurant owner cannot change the restaurant's ratings

- Delivery Person
  - A delivery person cannot submit more than one vehicle.
  - A delivery person cannot change their ratings.
- Restaurants
  - A restaurant can have only one owner.

5. Entity Relationship Diagram



6. Website

Here is a link to the repository for our project, where we will be publishing our reports and source code: [Insert GitHub Link]

7. Conclusion

…

References

[1] *Yemeksepeti*. https://www.yemeksepeti.com/en [Accessed: Feb. 21, 2021]

[2] *Uber Eats*. https://www.ubereats.com/ [Accessed: Feb. 21, 2021]

[3] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concept*s, 7th ed. New York: McGraw-Hill, 2019.