CS353 Database Management Systems Project

Project Proposal

Food Ordering and Delivery System

Group 19

Batuhan Özçömlekçi, 21703297

Yusuf Ziya Özgül, 21703158

Musa Ege Ünalan, 21803617

Mustafa Göktan Güdükbay, 21801740

Contents

1. Introduction

We aim to develop a database application for a food ordering and delivery system. This report describes our application, explains the reasons for using a database, and presents our functional and non-functional requirements. We will give a conceptual design (E/R Model) of the database we will implement in our application.

2. Project Description

The food ordering and delivery system will be an online application for customers to order from registered restaurants. There are three types of users: customers, restaurant owners, and delivery personnel. All users will have a unique username and login with their username and password. Users will have additional information such as their name, email, and birthdate. Customers can make orders, review them, and mark restaurants as their favorite restaurants. Customers will be able to review both the restaurant and the driver. Customers can add more than one address and phone information to the system and choose a delivery address. Restaurant owners can register their restaurants to the system, add food and beverages for their restaurant and create menus from them for customers to order. Restaurant owners will be able to specify the category of the food they add. Restaurants can also view reviews for their restaurant and give responses to them. Restaurants will have average ratings for the food they serve to create an impression on the customers.

Additionally, each restaurant will have one address. The system will organize the delivery by randomly assigning the order to an available driver. Delivery personnel can deliver orders for various restaurants. Customers can take out or add ingredients from the items they choose.

3. Why Do We Use a Database?

The food and delivery system requires many entities such as customers, restaurants, foods, and beverages. We plan to store information related to these entities within a database in an organized and efficient way. There are many relationships between these entities. The information about these relations can be obtained using queries between tables. Storing data in a database makes it easy to access the data related to customers, restaurants, menus, foods, beverages reviews, and addresses. Also, we use a database to store all data and relationships among them permanently. The database is the best alternative for storing all necessary data for the long term.

4. How Do We Use a Database?

The entities and relations between them will be mapped to a database for users to perform several functionalities. The users will be specialized into three entities; customers, restaurant owners, delivery personnel. All users will have a username, password, name, email, and birthdate. The phone numbers of users will also be stored in a database. Additionally, customers will have credit and many addresses. We will store restaurant owners and their restaurants (addresses, phone information, menus, and items) in the database. Orders made by customers will be stored in the database to access a restaurant's past orders or a customer's past orders. We will store the reviews made by customers for restaurants, customers, and delivery guys to see. When customers add or take out ingredients from an item in a menu, the order entity's note attribute will be used to store the ingredients.

5. Requirements

5.1. Functional Requirements

5.1.1. Users

- Users should be able to register with a unique username, password, first-name, last-name, and birthdate.
- Users should be able to have multiple contact information (phone_number).
- A registered user can either be a customer, restaurant owner, or delivery person.

5.1.2. Customer

- Customers should be able to order multiple items from the restaurant at the same time.
- Customers can specify order details (take out or add ingredients) at the check-out page, such as when they want the order delivered and request plastic spoons, forks, and knives, with their order.
- Customers can add credit to their accounts through payment options.
- Customers should have enough credit on their accounts for the order to be completed.
- Customers can have multiple addresses with different names.
  - Addresses consist of street(street_number, street_name, apt_number), city, county and zip.
- Customers should be able to modify their existing addresses.
- Customers should be able to see their order histories.
- Customers should be able to have favorite restaurants.
- Customers should be able to leave a review after they have received their order.
  - A review consists of a driver comment, restaurant comment, driver rating, and restaurant rating.

5.1.3. Restaurant Owner

- Restaurant owners should be able to own multiple restaurants.
- Restaurant owners should be able to see the order details when a customer orders from the restaurant.
- Restaurant owners should be able to see the order histories for the restaurants they own
- Restaurant owners should be able to add new items to their menus.
- Restaurant owners should be able to edit existing items on their menus.
- Restaurant owners should be able to respond to customer reviews.

5.1.4. Delivery Personnel

- Delivery personnel should be able to see the address of the restaurant for the current order.
- Delivery personnel should be able to see the delivery address for the current order.
- Delivery personnel should be able to mark the current order as delivered.
- Delivery personnel should be able to see their average rating.

5.2. Non-functional Requirements

5.2.1. Efficiency in Operations/Response Time

The functional operations such as retrieving, deleting, adding should be completed under the given time constraints. For this purpose, the database system must be structure so that the common operations should be performed easily, satisfying the time constraints. In general, this criterion can be named as functional efficiency of the system. Whenever a tradeoff is encountered between functional efficiency and storage efficiency, functional efficiency will be our priority.

5.2.2. Robustness

The database system should respond to every kind of query, retrieval, and removal with an appropriate and stable response. In other words, the system should be able to deal with erroneous input, and thus, it should be ready to function correctly for each request.

5.2.3. Usability

The user interface of the food delivery database application should be simple yet handy enough to be operated even by an entry-level user. Therefore, the documentation and a manual for the usage of the database should be provided for the users of the database.

5.2.4. Modifiability

Whenever there is a demand for storing new information, discarding old information, or removing old tables and structures, the database should be easily modifiable to adapt to new demands. In this way, the cost of creating the database from scratch can be omitted, and the existing database can be revised for new demands. Moreover, this criterion is aligned with the backward compatibility concept.

5.2.5. Scalability

The database application that will be created for food ordering and delivery service should be scalable to meet the demand of big-scale data storage and manipulation. Especially with the impact of the “Big Data” concept, scalability criterion is essential for food delivery database applications.

5.2.6. Maintainability

When an error occurs in the database, the database system should handle the error without manipulating the non-erroneous parts of the database structure. Users of the database should be notified when such maintenance is required, and the usage of the database system should be adjusted or limited according to ongoing maintenance constraints.

5.2.7. Cost

The overall cost, which includes monetary cost, labor cost, and maintenance cost, of the database should satisfy the limitations for the initial stage of the project. The life-cycle cost of the database should not be huge for an initially small business plan.

5.2.8. Recovery

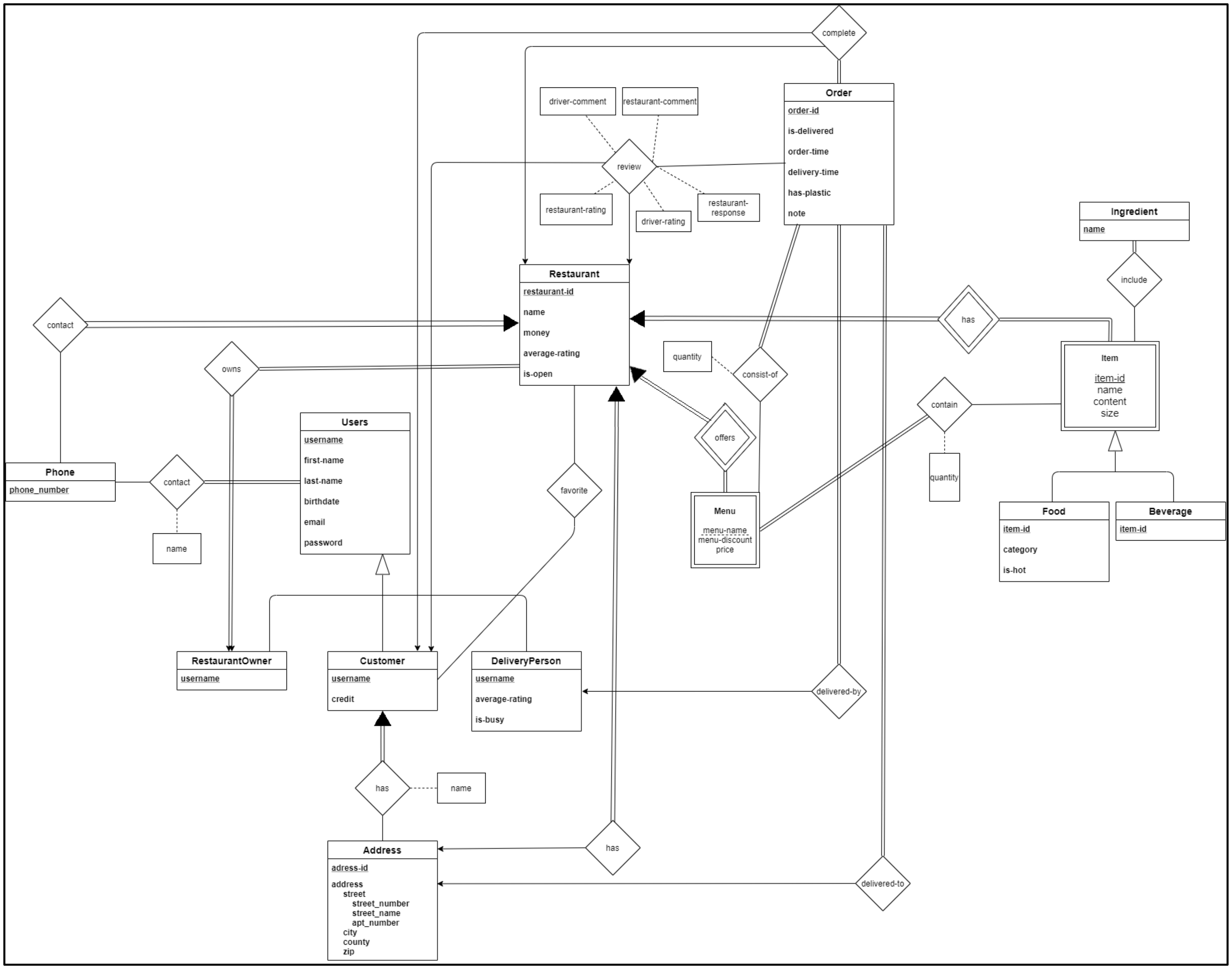
The database should be backed up to local storage regularly to prevent information loss whenever a system failure happens. For this purpose, the information stored in the database can be saved periodically to another location.

6. Constraints

1. Every restaurant must have at least one phone number.
2. Each customer must have at least one phone number.
3. Each restaurant must have exactly one address.
4. Every customer must have at least one address.
5. Each restaurant must have exactly one owner.
6. Each restaurant should have at least one menu and one item.
7. Each menu should have at least one item.
8. Each order is related to exactly one customer and one restaurant.
9. Different restaurants cannot have the same item.
10. Each order is delivered by exactly one delivery personnel.
11. Each order is delivered to exactly one address.
12. Each menu belongs to exactly one restaurant.
13. Restaurants have different phone numbers.

7. Conceptual Database Design Using the Entity-Relationship Data Model

Conceptual Database Design for our Food Ordering and Delivery System using the Entity-Relationship Data Model is given on the following page. Menu and item entitity sets are defined as weak entity sets because neither of them can exist in the database without an identifying (owner) restaurant. Orders can contain only menus and menus can contain one or more items.


complete
driver-comment
restaurant-comment
Order
order-id
is-delivered
order-time
delivery-time
has-plastic
note
review
restaurant-rating
driver-rating
restaurant-response
Ingredient
name
include
Restaurant
restaurant-id
name
money
average-rating
is-open
contact
owns
has
quantity
consist-of
contain
Item
item-id
name
content
size
offers
quantity
Users
username
first-name
last-name
birthdate
email
password
Phone
phone_number
contact
name
favorite
Menu
menu-name
menu-discount
price
Food
item-id
category
is-hot
Beverage
item-id
RestaurantOwner
username
Customer
username
credit
DeliveryPerson
username
average-rating
is-busy
delivered-by
has
name
Address
adress-id
address
street
street_number
street_name
apt_number
city
county
zip
has
delivered-to

8. The Website

Project Website Link: https://bozcomlekci.github.io/CS353-Project/

Reference

A. Silberschatz; H. Korth; S. Sudarshan, Database System Concepts, 7th edition, McGraw- Hill, 2019