



2020-2021 Spring

CS-353

Database Systems

Project Proposal

Group 23

Alp Üneri 21802481

Ayberk Yılmaz 21702250

Metehan Gürbüz 21602687

Süleyman Semih Demir 21702949

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Problem Statement</b>	<b>2</b>
<b>Requirements</b>	<b>3</b>
Viewing Available Restaurants	3
Choosing an Available Restaurant	3
Browsing the Menu of the Restaurant	3
Selecting Items off a Menu to Order Them	3
Placing an Order	3
Adding Credit	3
Assigning the Order to a Delivery Guy	3
Order Reviews	3
Changing of Attributes	4
<b>Limitations</b>	<b>4</b>
<b>Conceptual Design (E/R Model)</b>	<b>5</b>
Entities	6
User	6
Owner	6
Customer	6
Delivery Guy	6
Restaurant	6
Menu	6
Menuitem	6
Food	6
Beverage	7
Order	7
Relations	7
Owns Relationship	7
Works for Relationship	7
Has Relationship	7
Review Relationship	7
Consists of Relationship	7
Contains Relationship	7
Place Relationship	8
Delivers Relationship	8

# Problem Statement

Our project is going to be the implementation of a food ordering and delivery system. There will be users, restaurants, menus, food and beverages, ratings and reviews on a particular order et cetera. In our system, a user will either be a restaurant owner, a customer, or a delivery guy.

Customers will be able to place orders from restaurants by picking food or beverages from the menu of that restaurant. The system will then check whether the customer has sufficient credit to make that order. If the user is able to issue the particular order, the system will then assign that order to any available delivery guy that works for that restaurant. After the delivery guy delivers the order, the customer will be able to write a review for that particular order, giving the order a star rating out of five and leaving any comment that they might have.

Our proposed system will use a database to keep track of information regarding the entities within our system, such as users, restaurants, menus, and ratings. A database is beneficial for our solution as it allows for neat storage of grouped information into a tight system. Information such as the food items on a menu are able to be neatly stored within a database as they all share common attributes which allow for easier grouping. For more detailed information regarding the requirements and limitations of our project, please see the following sections.

# Requirements

## Viewing Available Restaurants

A user will be able to browse a list of available restaurants to see the names of the restaurants, their review scores, and the type of cuisine that they offer.

## Choosing an Available Restaurant

Users will be able to choose a restaurant from the available restaurants list to place an order from.

## Browsing the Menu of the Restaurant

Once they have chosen a restaurant, users will be able to browse the menu of the selected restaurant to decide on the food or beverages they wish to order.

## Selecting Items off a Menu to Order Them

Users will be able to select particular items from a menu that they are browsing to order them from the restaurant.

## Placing an Order

After selecting the items that they wish to order from the menu of the restaurant, users will be able to place an order if they have sufficient credit to do so.

## Adding Credit

Customers will be able to add credit to their name by paying money to the system.

## Assigning the Order to a Delivery Guy

Once an order is placed, the system will assign it to any available delivery guy working for the restaurant.

## Order Reviews

After an order has been delivered, customers will be able to review their order; providing a star rating out of five for the order and as well any written comments that they might have.

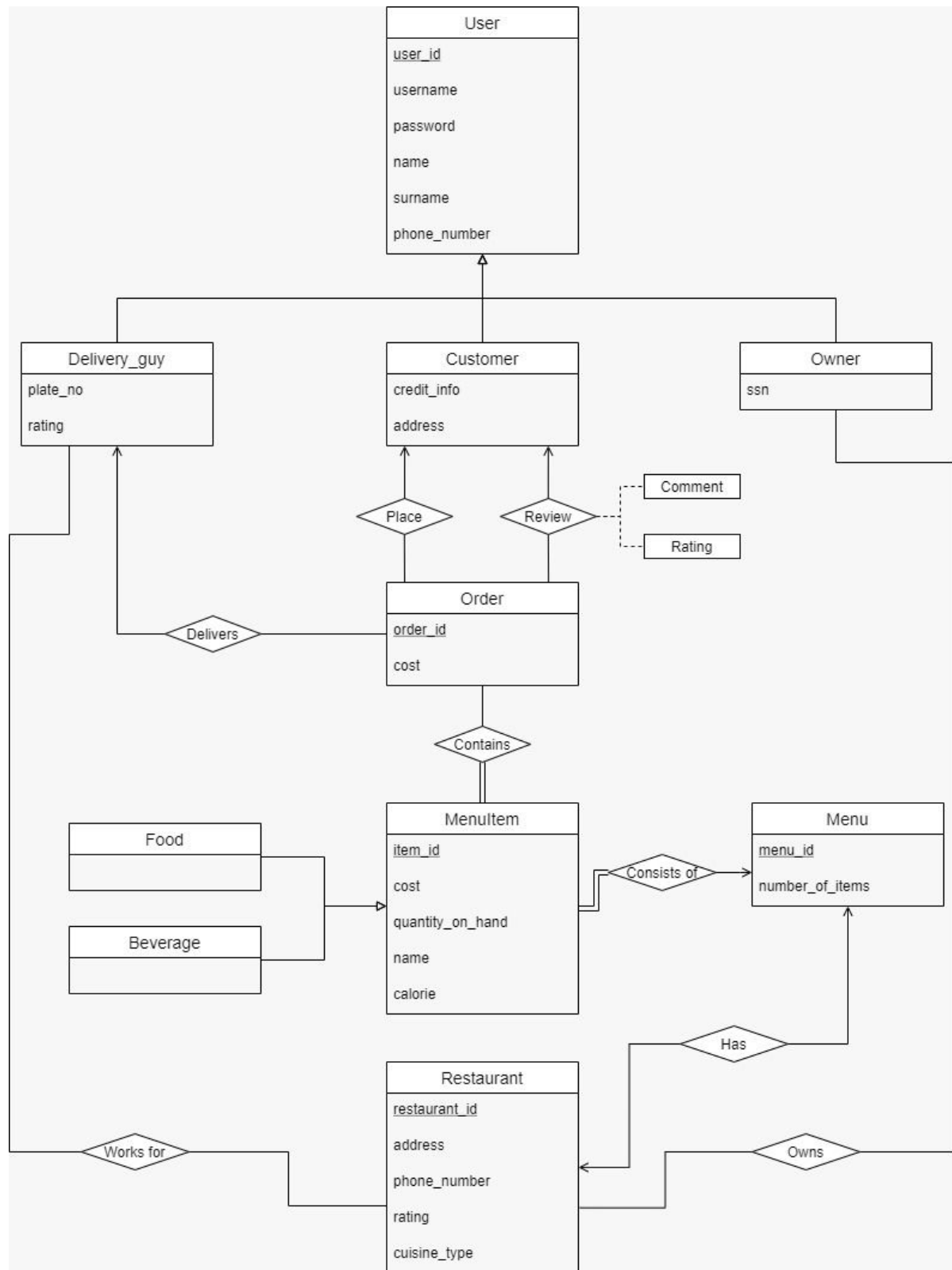
## Changing of Attributes

Users will be able to change their attributes to appropriate other values, such as a customer changing their address, a restaurant adding or removing certain items from their menu or an owner of a restaurant hiring new delivery guys to their restaurant.

## Limitations

- A customer will only be storing a single address as an attribute. Therefore, users will not be able to order to multiple different addresses without changing their address attribute manually from a menu.
- Restaurants will only have a single cuisine type that they will appear to serve. If a restaurant serves multiple types of cuisines, it will be registered as a mixed cuisine restaurant.
- Users will be rating orders as a whole, and will not be giving star reviews to individual qualities of the order such as its speed, taste, service et cetera. This is done in order to simplify the reviewing process.
- A restaurant will only be able to have a single menu, so restaurants will not be able to have different menus for food items, beverages, deserts et cetera. They will instead be forced to have a single menu containing all the items they offer, and will be able to separate them into categories depending on the ordering of their menu.
- An order will be delivered by only a single delivery guy in our system. In real life, for very large orders sometimes more than one delivery guy might be necessary for delivering that single order. However, for the purposes of our system, we will assume that every order will be small enough to not require more than one delivery guy.

# Conceptual Design (E/R Model)



# Entities

## User

The user entity will be the parent entity of the three types of users of our proposed system: owners, customers, and delivery guys. The user entity will have the primary key `user_id`. It will also have attributes `username`, `password`, `name`, `surname`, and `phone number`.

## Owner

Owner entities will be owners of restaurants. They will be a child entity of the user entity. They will have the additional attribute of `ssn` (social security number) for legal purposes.

## Customer

Customers will be the users who will be browsing restaurants and placing orders from their menus. They will have the additional attributes of `credit_info` to store how much credit they have as well as `address` to store where they wish their orders to be delivered to.

## Delivery Guy

Delivery guys will be the users who are delivering the orders of customers to their addresses. Each delivery guy will be able to work for any number of restaurants. They will have the additional attributes of `plate_no` for legal purposes and a `rating`.

## Restaurant

The restaurant entity will have a primary key called `restaurant_id`, and will as well store `address`, `phone_number`, `rating`, and `cuisine_type` attributes. If a restaurant serves more than a single type of food, its `cuisine type` will be set to `mixed`.

## Menu

A menu will have the primary key of `menu_id`, and will as well have the number of items it contains as another attribute.

## MenuItem

A menu item will have primary key `item_id`, and is the parent entity of both food and beverage entities. It will also store as attributes its `cost`, `quantity on hand`, `name`, and `caloric content`.

## Food

Food entities are a child entity of the Menu Item entity. They are on the diagram for the sake of clarity and classification, and will not have any additional attributes.

## Beverage

Beverage entities are a child entity of the Menu Item entity. They are on the diagram for the sake of clarity and classification, and will not have any additional attributes.

## Order

An order will have a primary key `order_id`, and will as well keep its cost as an attribute.

## Relations

### Owns Relationship

This relationship will be a many to many relationship between the Owner and Restaurant entities. A restaurant will be able to be owned by multiple owners, and likewise an owner will be able to own multiple restaurants.

### Works for Relationship

This relationship will be a many to many relationship between the Restaurant and Delivery Guy entities. A delivery guy will be able to work for multiple restaurants, and likewise a restaurant will be able to have many delivery guys.

### Has Relationship

This relationship will be a one to one relationship between the Restaurant and Menu entities. A restaurant will only have a single menu, and likewise a menu will only belong to a single restaurant.

### Review Relationship

This relationship is a one to many relationship between the Customer and Order entities with two stored attributes. A customer will be able to leave reviews for many orders, but an order will only have a single customer that can review it. This relationship will as well store two attributes being the five star rating and the comment associated with the customer review.

### Consists of Relationship

This relationship is a one to many strong relationship between the Menu and MenuItem entities. A menu will consist of at least one menu item, and a menu item will be in at most one menu.

### Contains Relationship

This relationship is a many to many strong relationship between the Order and MenuItem entities. An order will contain at least one menu item, and a menu item can be contained in many orders.



## Place Relationship

This relationship is a one to many relationship between the Customer and Order entities. A customer will be able to place multiple orders, but an order will be able to have only a single customer.

## Delivers Relationship

This relationship is a one to many relationship between the Delivery Guy and Order entities. A delivery guy will be able to deliver many orders, but an order will only have a single delivery guy.

Link to Website: <https://github.com/alpuneri/CS353>