**Database Systems Final Project**

**Project title:** Pizza Company

**Team members:** Arslan Gait

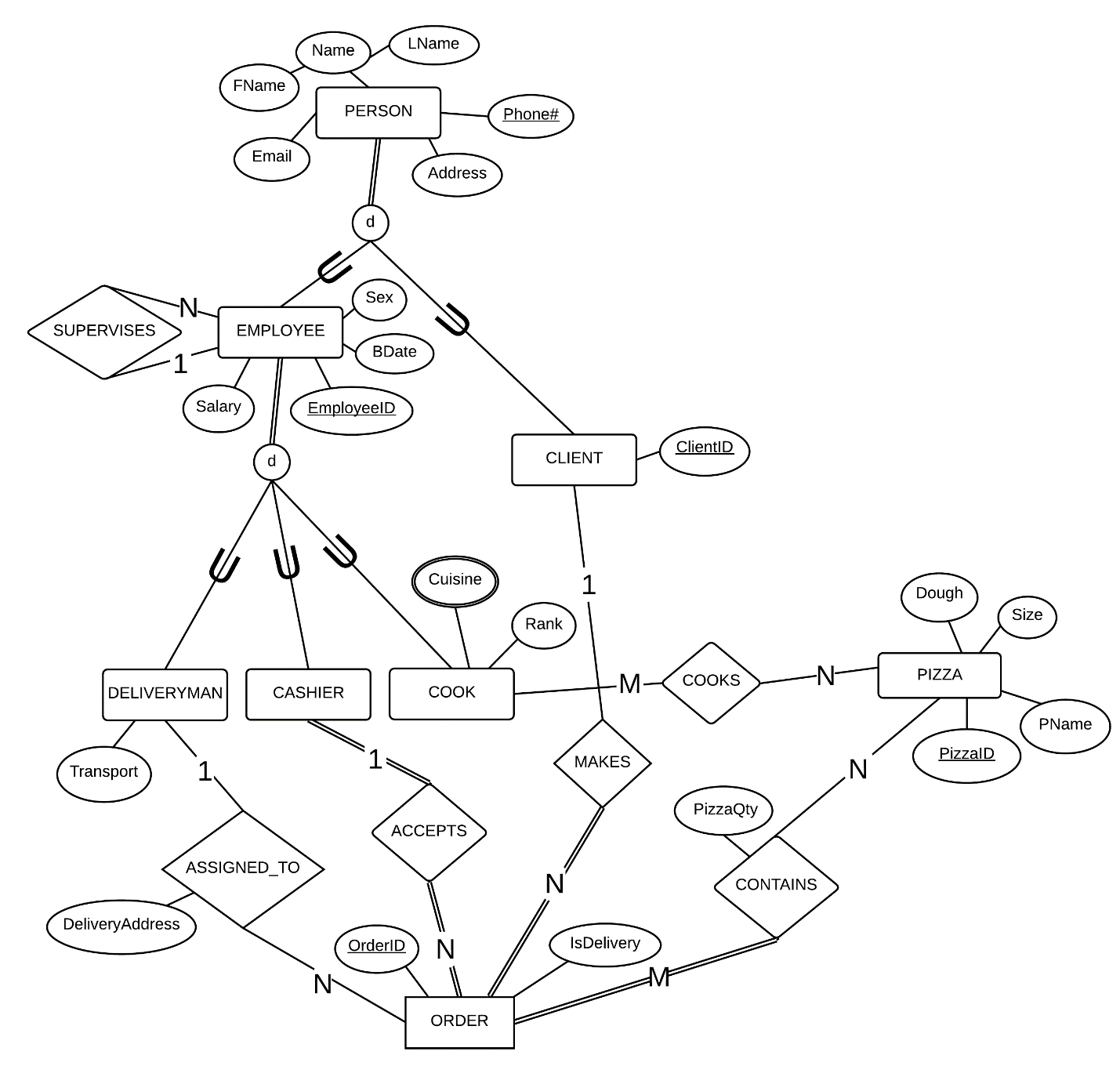
Vladimir Duisenbinov

**Project Overview**

To begin with, as in many modern different sale companies, pizza companies also have their own electronic sales services. Consequently, our project was connected to creating of a prototype of such services. Despite the fact that this project is not so complex, its parts, working functions, interface could be used in real-world situations. Mainly, the way our system works is applied in existing applications, which include work with databases (not only in pizza companies).

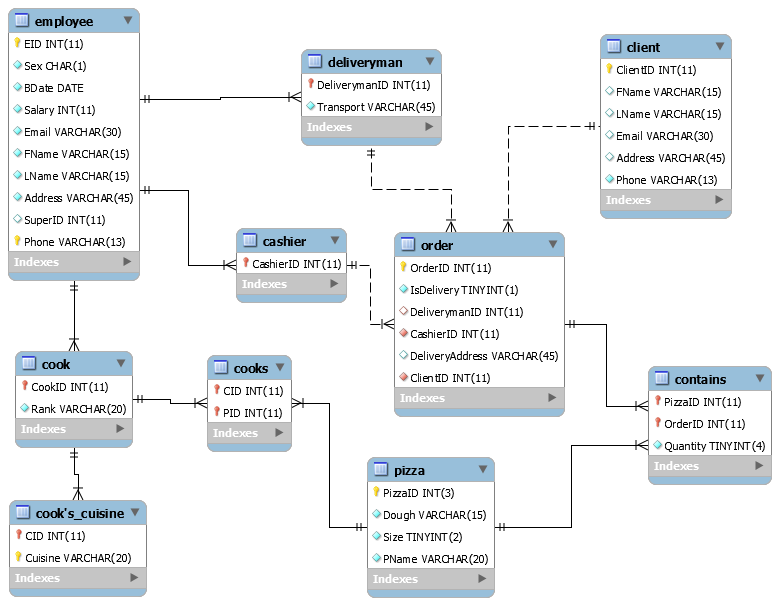
As it was pointed out above, our system is quite simple: user can sign in to his/her account, make some orders, view last order and change personal information.

**High-level Design**

****

**Figure 1. EER Diagram in Chen notation**

**Database design**

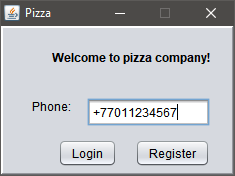


**Figure 2. Relational model in crow’s foot notation**

**Methodology**

Our database was created in MySQL Workbench. Everything in the application was written in Java, using NetBeans IDE.

**System’s GUI**



**Figure 3. Welcome window**

The first window, which user sees on a pizza company’s site or on a special application, asks him to sign in if he is already registered in the system. The way how user does logging into the system is through the entering of his phone number which was written during registration before (see Figure 3). If the user enters another phone number by mistake, the system will show that that there is no such phone number in the system (not registered).

When “Login” button is pressed, system gets user’s id and checks, whether the user exists in the database or not:

**SELECT** ClientID

**FROM** `client`

**WHERE** Phone **=** '+77011234567'**;**

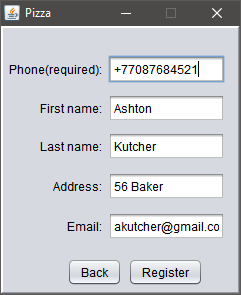
After that, the system loads data from pizza and deliveryman tables.

**SELECT** PName **AS** Pizza\_name**,** Dough**,** Size

**FROM** `pizza`

**ORDER** **BY** PName**;**

**SELECT** **\*** **FROM** deliveryman**;**



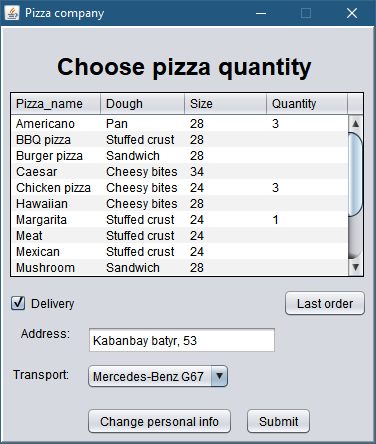
**Figure 4. Registration process**

If user makes an order for the first time and he is not registered in the system, he needs to press on “Register” button and fill fields such as “Phone” that is required, “First Name”, “Last Name”, “Address” and “Email” respectively (see Figure 4 above). If he enters the phone that is already registered, the system will inform user that the number is already used.

Registration of a new client:

**INSERT** **INTO** client **(**`FName`**,** `LName`**,** `Email`**,** `Address`**,** `Phone`**)**

**VALUES** **(**'Ashton'**,** 'Kutcher'**,** 'akutcher@gmail.com'**,** '56 Baker'**,** '+77087684521'**);**

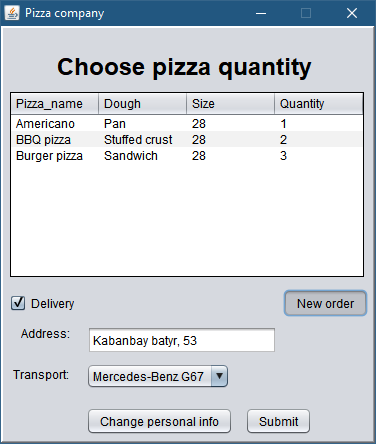


**Figure 5. Choosing pizza (-s) window**

Next, user sees the table of all available pizza types, its corresponding descriptive constituents such as “Dough” from which pizza was cooked, “Size” of the pizza, and “Quantity” field where user writes the respective quantity of pizza (-s) which he wills to order (see Figure 5). Every pizza has its “Quantity” field and user decides by himself whether to fill the “Quantity” field of particular pizza type or not. The only pizzas that will go into the order are those where “Quantity” field is filled with non-negative and not zero number.

Below the table of pizzas from Figure 5, there is a “Delivery” option for the client. If the user wills for pizza (-s) to be delivered, he ticks the “Delivery” option and writes a supposed address where pizza (-s) should be delivered. Below the “Address” field, there is a transport choose option where user can choose which transport should deliver pizza (-s) to him. If the user ticks the “Delivery” option but does not fill the “Address” field and presses on “Submit” button, then the system will inform him that “Address” field was not filled with appropriate information.

If the user presses on “Last Order” and the user did not have any orders before, it will be shown to him that he did not have it before, otherwise, the list of previous order which he can re-order will be shown to him (Figure 6). Then, the “New Order” button will appear, and if the user presses on it, window as in Figure 5 will be shown to him again.



**Figure 6. Last Order option**

**Last order** button is pressed:

1. The system gets client’s last order’s id:

**SELECT** **MAX(**`OrderID`**)** **FROM** `**order**` **WHERE** `ClientID` **=** '1'**;**

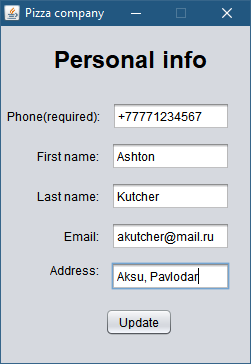
2. Previous order is loaded into the system:

**SELECT** `PName` **AS** Pizza\_name**,** `Dough`**,** `**Size**`**,** `Quantity`

**FROM** **(**`pizza` p **JOIN** `**contains**` c **ON** p**.**PizzaID **=** c**.**PizzaID**)**

**WHERE** `OrderID` **=** '4'

**ORDER** **BY** PName**;**



**Figure 7. Personal Info window**

Apart from this, there is “Change personal info” button. If the user presses on “Change Personal Info”, then the new window of personal info will appear (see Figure 7). It has the same fields as in the registration process and it will also show the user whether the new entered phone already exists in the system. Consequently, user cannot enter the phone that is already in the database.

**Update** button is pressed:

**UPDATE** `Client`

**SET** FName **=** 'Ashton'**,** LName **=** 'Kutcher'**,** Email **=** 'akutcher@mail.ru'**,**

Address **=** 'Aksu, Pavlodar'**,** Phone **=** '+77011234567'

**WHERE** ClientID **=** '1'**;**

If there are no problems and user presses on “Submit”, selected order is sent to the database and the next window of order confirmation will appear (see Figure 8).

**Submit** button is pressed:

1. Get deliveryman’s id:

**SELECT** DeliverymanID

**FROM** deliveryman

**WHERE** transport **=** 'Mercedes-Benz G67'**;**

1. Insert order’s data into the table:

**INSERT** **INTO** `**order**`

**(**`IsDelivery`**,** `DeliverymanID`**,** `ClientID`**,** `DeliveryAddress`**,** `CashierID`**)**

**VALUES** **(**'1'**,** '7'**,** '1'**,** 'Kabanbay batyr, 53'**,** '4'**);**

1. Get order’s id:

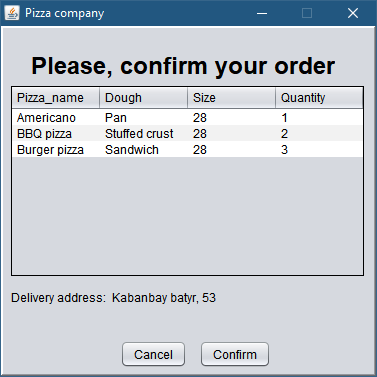
**SELECT** **MAX(**`OrderID`**)** **FROM** `**order**` **WHERE** ClientID **=** '1'**;**

1. Insert all selected pizzas into the table:

**INSERT** **INTO** `**contains**` **(**`PizzaID`**,** `OrderID`**,** `Quantity`**)** **VALUES** **(**'11'**,** '9'**,** '3'**);**

**INSERT** **INTO** `**contains**` **(**`PizzaID`**,** `OrderID`**,** `Quantity`**)** **VALUES** **(**'15'**,** '9'**,** '2'**);**

**INSERT** **INTO** `**contains**` **(**`PizzaID`**,** `OrderID`**,** `Quantity`**)** **VALUES** **(**'8'**,** '9'**,** '2'**);**



**Figure 8. Confirmation window**

The order confirmation window will show to user all pizza (-s), which he has chosen before. Table will show chosen pizzas with its descriptive constituents and its quantity number (-s). If the client chose the “Delivery” option before, then delivery address that is placed below the table, will be shown to user. If the user did not choose delivery, then “Delivery address” will not appear. This window allows user to look whether he made some mistakes during the pizza choose process. If the user did all correctly, he presses on “Confirm” button and window about success order will be shown to him. If he wills to cancel his order, he needs to push on “Cancel” and make another order from the beginning (all chosen pizza quantities are going to be as empty fields).

**Cancel** button is pressed:

1. The order is deleted from the database:

**DELETE** **FROM** `**contains**` **WHERE** `OrderID` **=** '9'**;**

**DELETE** **FROM** `**order**` **WHERE** `OrderID` **=** '9'**;**

1. The list of pizzas’ is loaded:

**SELECT** PName **AS** Pizza\_name**,** Dough**,** Size

**FROM** `pizza`

**ORDER** **BY** Pname**;**

**Development**

In our team, every member had his job. Vladimir was mostly responsible for writing part, since he was ill and had to prepare for missed and ongoing midterms. The rest part, including coding, was done by Arslan.