Database Management Systems

Phase-3

## GROUP-17

Berkay Çetin

Mechmet Chotzoglou Chalil

**VETERINARY MANAGEMENT SYSTEM**

**Detailed Progress Report Through Phase-1 to Phase-3**

PHASE-1

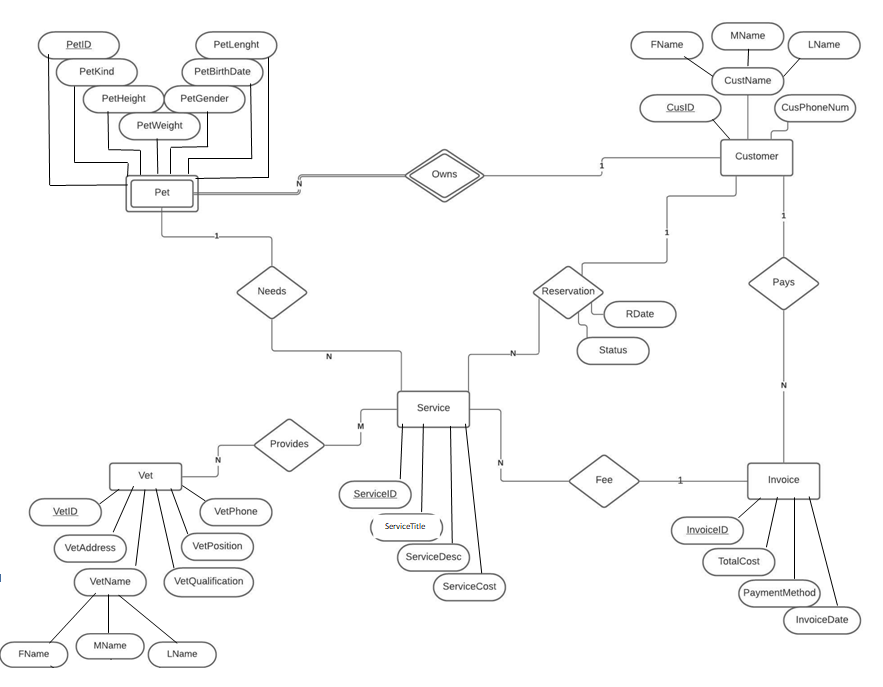
The topic given to our group during phase 1 was veterinary management system. And we were asked to create a database from start to finish using this topic.

In this phase we should have done Requirement analysis, E/R diagram and write at least 5 interesting queries.

We knew that the design we made at the beginning was important and that it would be difficult to make changes after creating the database so we first had a discussion as project group mates. As a result of our discussion, we deemed appropriate to design a database as follows.

-The Veterinary database we created keeps track of Pets, Pet characteristics, Customers (Owners of the pets), Service information, Vet (Veterinarians) information and the Invoice. In the process of creating this database we took reference from real life applications of a veterinary and based on that we created the relationships between entity sets.

Then we designed an E/R diagram to identify the tables and relationships we need to create. Our first E/R diagram was as follows. This diagram contained some mistakes. We fixed these mistakes in phase 2.



Our Interesting Queries in Phase-1

1. Provide a ranking of the customers based on the amount of money they spent in the clinic in wanted year
2. Which customer owns the biggest pet?
3. Which pet kind needed the most service?
4. Which customers made a reservation and didn’t come?
5. Who is the most working veterinarian? (for employee of the month)

PHASE-2

In this phase, we were asked to correct the mistakes we made in phase-1 (if any) and then pour the database that we specified in the E/R diagram into tables(Relational Schema), and then create the database by writing SQL codes.

It was said that we can use MySQL or Microsoft SQL Server as a DB Engine. We chose the MySQL option and used this DB Engine for our SQL codes.

First, we downloaded MySQL development tools (MySQL Workbench) from MySQL's web page. Since it is a platform that we have not used before, we had to do research first. Afterwards, we started to code our database using the research we did and what we learned in the lesson.

When we transferred the database to the physical environment, we saw our mistakes in phase-1. In addition, we saw that our interesting queries, which we wrote in phase-1, were insufficient and some were not appropriate. That's why we made some changes in our queries. (We changed fourth query and we updated first and fifth queries)

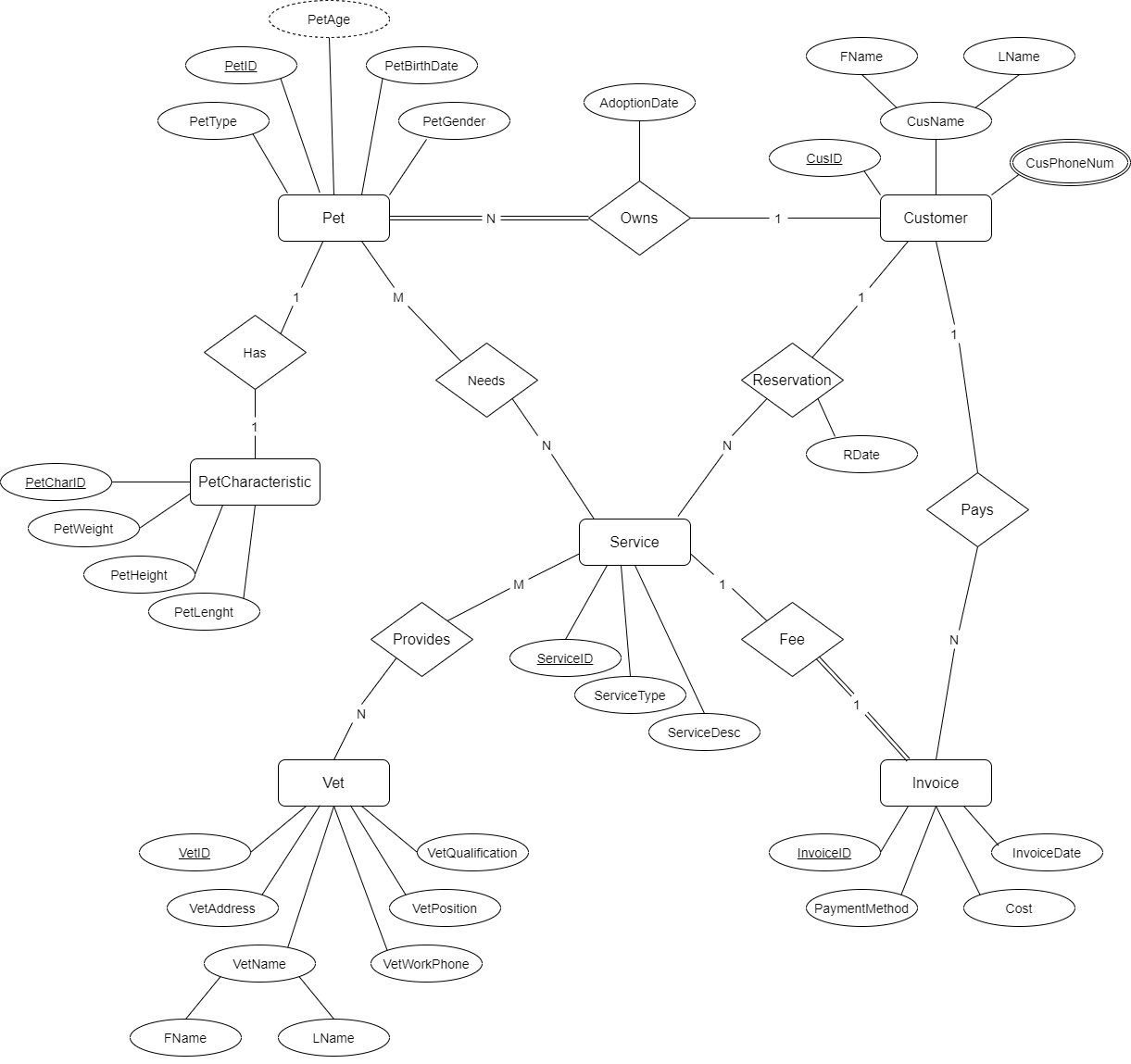
Our Interesting Queries in Phase-2

1. Top three most spending customer or customers at one time.
2. Which customer owns the biggest pet? (in terms of weight)
3. Which pet kind needed the most services (Sequentially)?
4. Total amounts of paid in both credit card and cash separately.
5. Which veterinarian has provided how many services in total (Sequentially)? --> For employee of the month

Also, when we looked at our E/R diagram, we came to the conclusion that it would be better to make some changes there.

In Phase-1, we determined the pet characteristics in the pet entity set, but we thought that these characteristics are variable and it would be more correct to have them in a different entity set. That's why we created a separate entity set for pet characteristics.

We also added adoption date as an extra attribute to the owns relation. Our second E/R diagram, in its corrected form, was as follows.



We've also fixed a few more mistakes we've seen here:

1-We changed customer telephone number attribute as multi-valued attribute. Because customer may have one or more telephone number.

2-We removed Middle Name attribute from either customer and vet entity sets. Because we thought this would cause null values ​​to increase.

3-We changed the relation a little between invoice and service entity sets. We changed the relation n-1 to 1-1 because we found it more accurate for a service to have an invoice. And we changed the participation as total participation. Because there will be no invoice if there is no service. Also we changed 1-N relation between pet and service with N-N. Because we thought it would be more accurate.

4-We added age attribute as derived attribute. Because age can be calculated using the date of birth attribute.

5-We removed weak relation between pet and customer and we added total participation here.

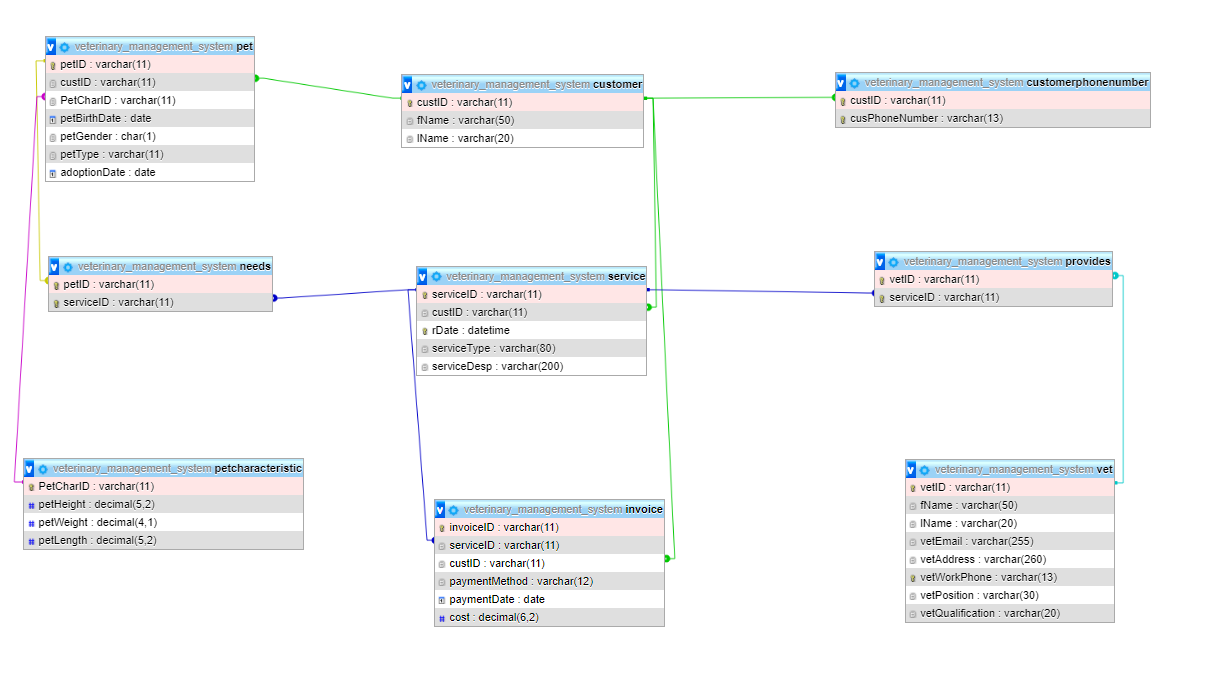
6-We removed cost attribute from service and we also removed total cost attribute from invoice. Instead of these attributes we added cost attribute to invoice entity set. We thought it would be more understandable and logical this way.

The Relational Schema We Created for Design Our Database in Phase-2

Our Relational Schema:

Diagram

Description automatically generated

Our Relational Schema from XAMPP 

Latest Improvements on the MySQL Side:

1-We added ON UPDATE CASCADE to every FK attribute. With this way Updates in linked tables can be updated simultaneously( ON DELETE CASCADE was already added).

2- In vet table vetEmail attribute changed to UNIQUE( for a unique veterinary login in our interface).

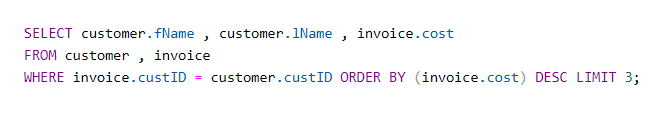
-Interesting Queries-

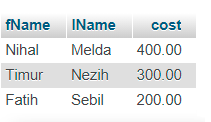
Our Interesting Queries and Their Results from Phase-2

(We didn't change our queries in phase 3)

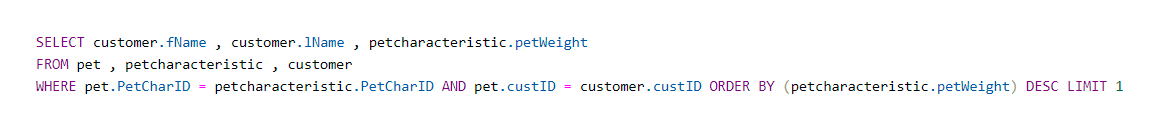
1. Top three most spending customer or customers at one time.
2. Which customer owns the biggest pet? (in terms of weight)
3. Which pet kind needed the most services (Sequentially)?
4. Total amounts of paid in both credit card and cash separately.
5. Which veterinarian has provided how many services in total (Sequentially)? --> For employee of the month

1-Top three most spending customer or customers at one time



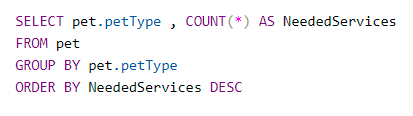


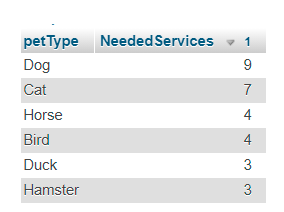
2-Which customer owns the biggest pet? (in terms of weight)



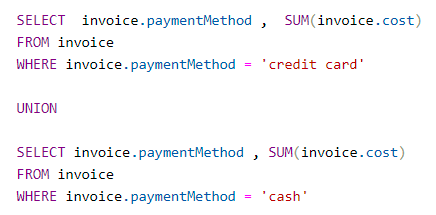


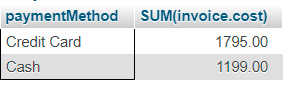
3- Which pet kind needed the most services (Sequentially)?



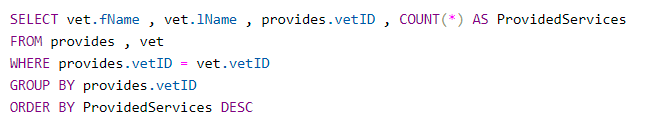


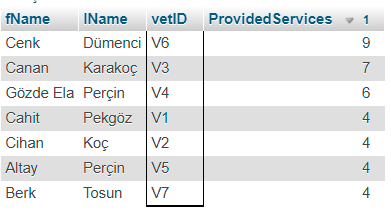
4- Total amounts of paid in both credit card and cash separately.





5-Which veterinarian has provided how many services in total (Sequentially)? -->For employee of the month





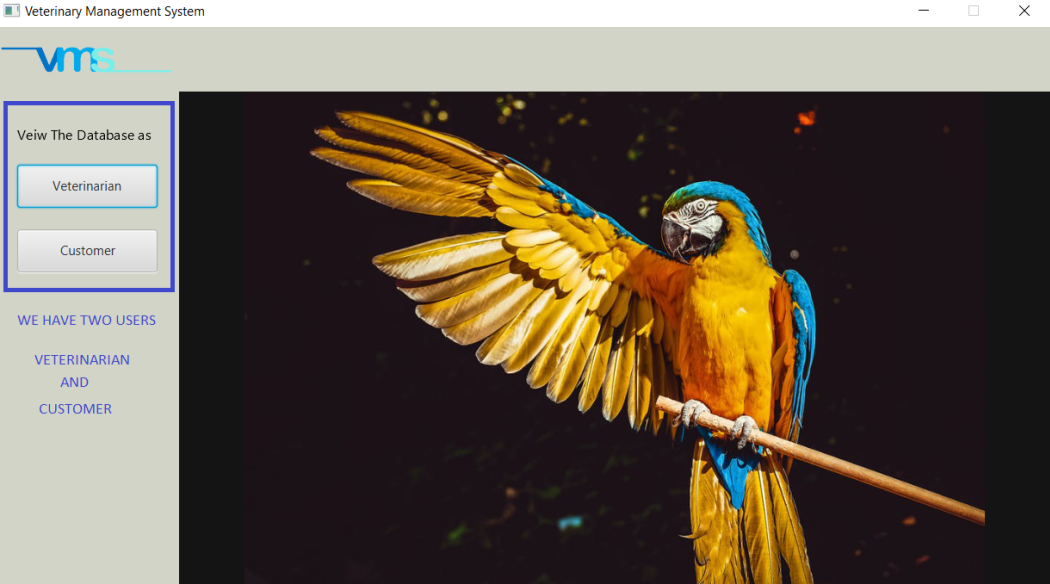
PHASE-3

For this phase, which is the last phase of our project, we were asked to create an app using the database we have created so far with an interface.

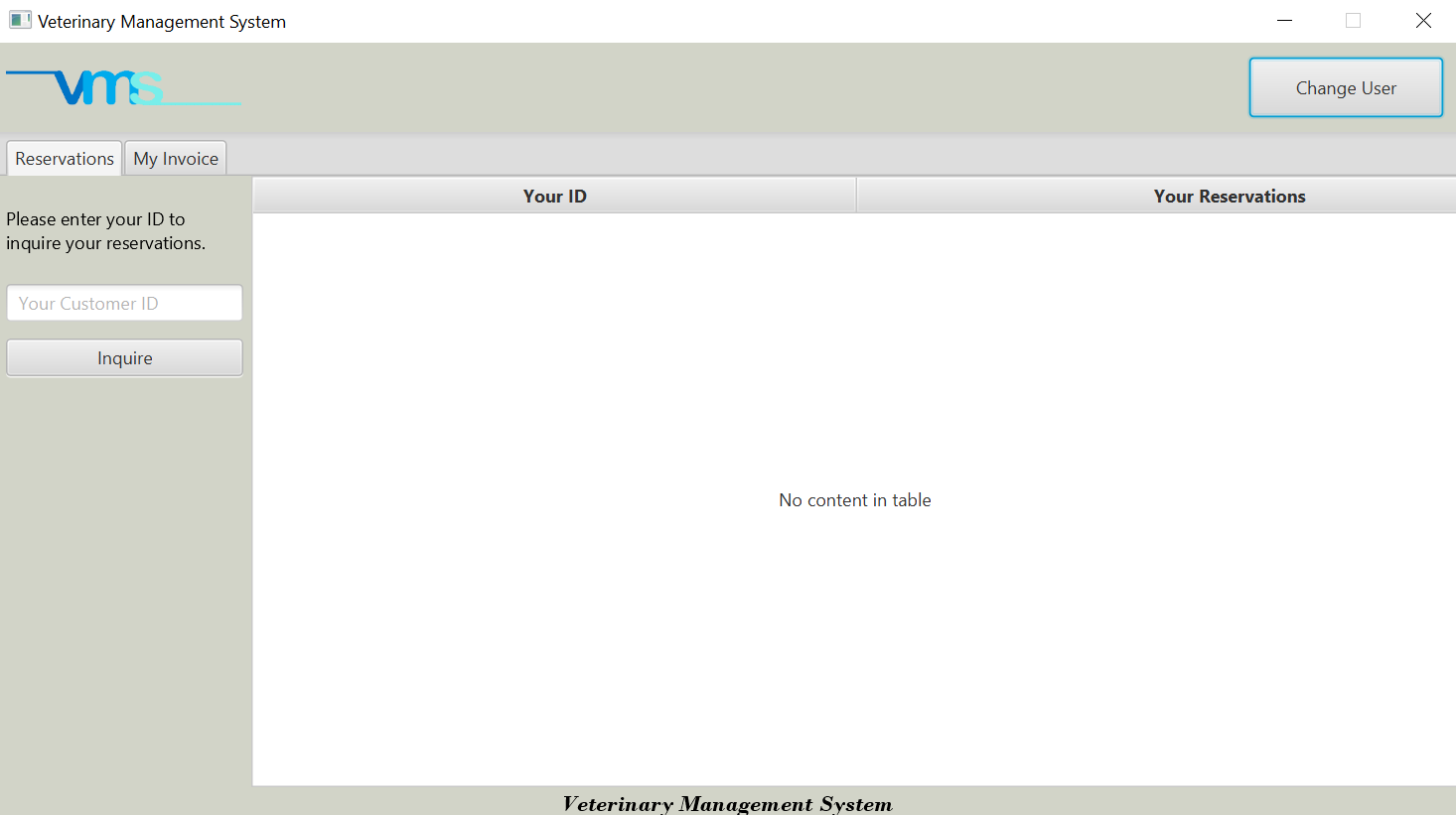
-We used JavaFX for our interface by using fxml documents we designed the whole system and with java back and the program logic and the connection with front-end has been made.

-We implemented a JavaFX and MySQL libraries and dependencies using maven. By doing that way a user which tries to run our program doesn’t need to implement VM options or any libraries.

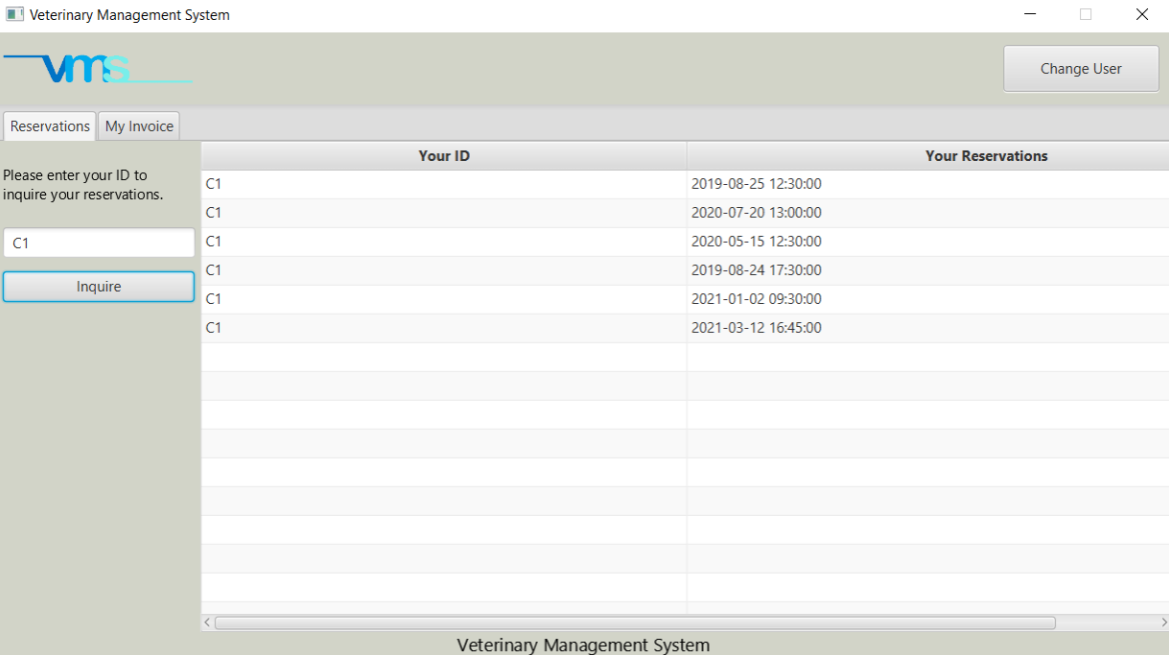
-When we start the program, the user selection screen welcomes us. In this screen we have two options. First one is viewing the database as Veterinarian and the second, option is to view the database as Customer.



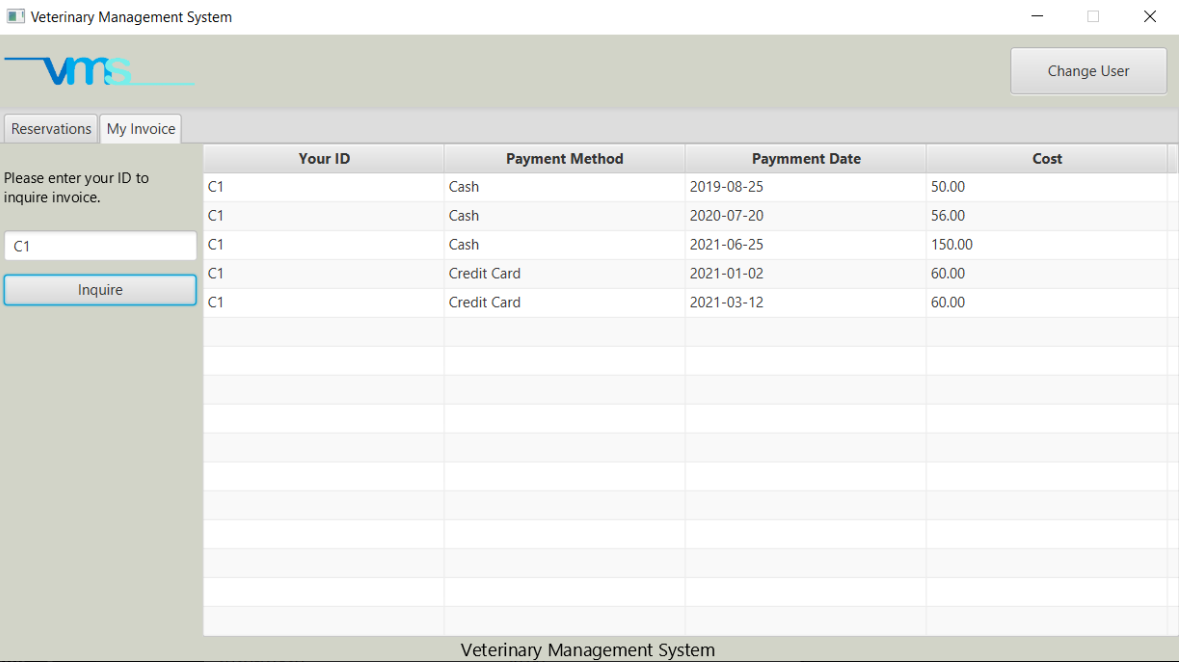
-If the user selects the Customer button a screen as follows opens.



-By writing your customer id and clicking the inquire button our customer can view his/her past reservations and up coming ones.



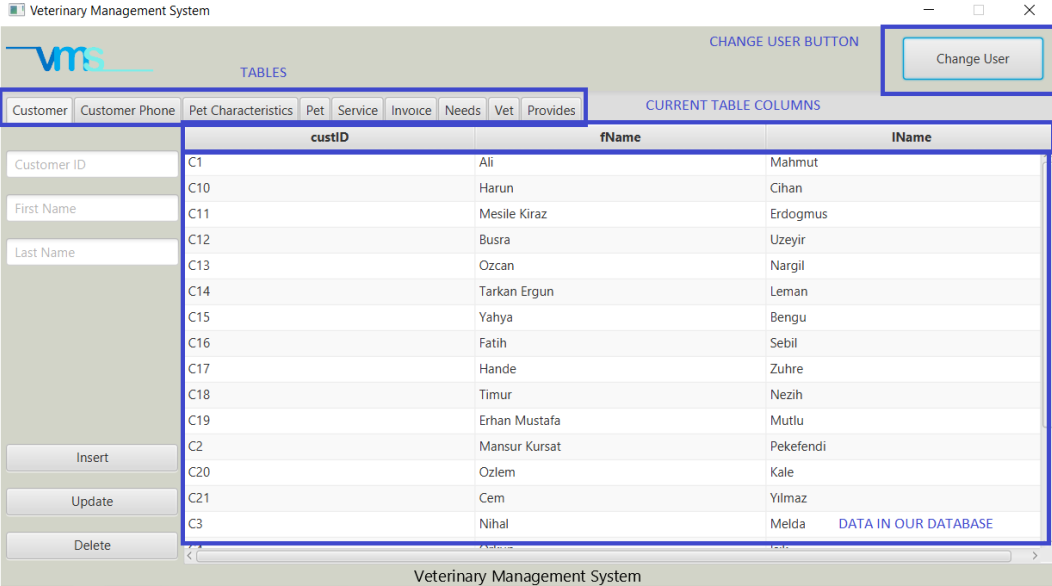
-In My invoices section the same logic and the same interface was implemented. If the user enters his/her customer id the system checks the database specially invoice table to find invoices for this specific customer.



* If the interface user selects view the database as Veterinarian user goes to a login screen. To log in this section, the user must be veterinarian. The email and the password (Vet Work Phone) of user must be in the database after checking the validness of the information that is entered then the user can view this section of the interface. Also, under the button we have a label that tells us the status of the database connection.



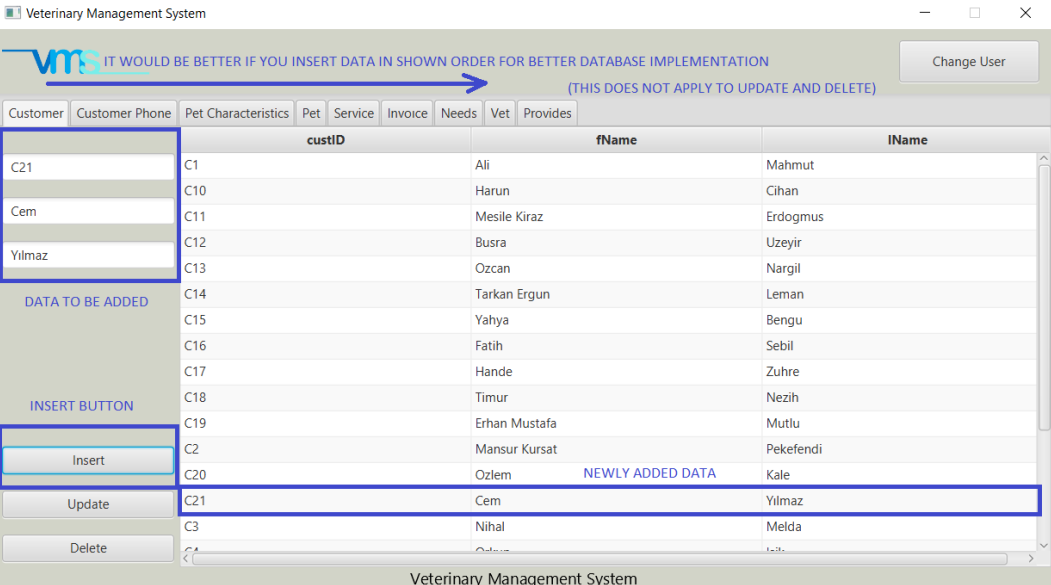
-After logging in to the veterinary interface we can view our database with all the tables. Tables are added into tabs named as same as table names in our database. The information in our database printed into the interface using Table View and Table Column attributes. Each tab contains the same table column attributes as we have in our database. Also, we a Change User button that when it is clicked user goes to Select User screen.



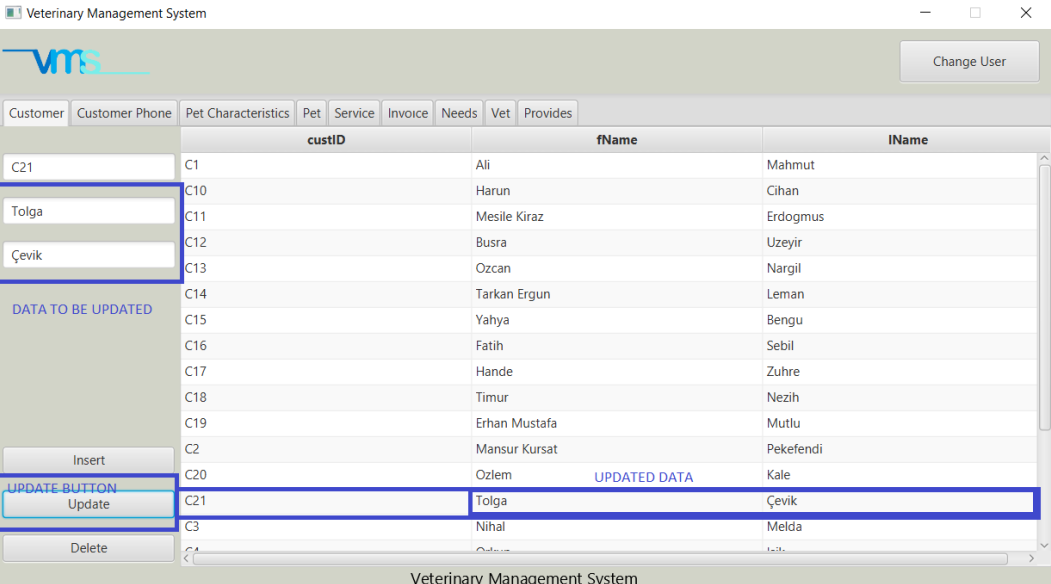
-Veterinarian can also manipulate the date in our database by using insert, update and delete functionality.

-To insert new data in the database our user just needs to fill the text fields and click the insert button.

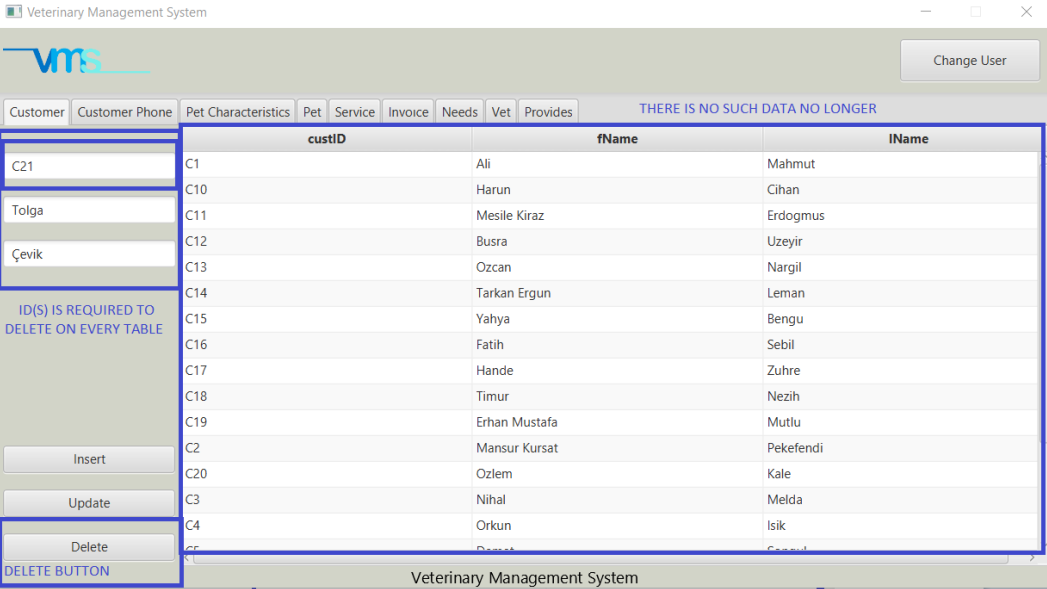
Also, if there is a scenario that a veterinarian wants to insert in all tables, we recommend to make the insertion in the order of our tabs from start to finish.



* To update the data in the database you required to fill all the text fields. The ID parts in all the tables indicated the row that in being updated and the rest of the fields are updatable data.



-To delete the data in the database you just need to enter the ID text field and press delete button to delete the row.



-To exit from the program is enough to press the x button placed at top right.

### -You can check our how to run folder to more detailed information about how to run our app and our operations for better understanding-