



**Software Engineering and Testing. BSC Year 2, 2020/2021
(Assignment 3 - 20%)**

Assessment 3: Design and Draft Implementation

Submitted by: Habiba Nour,	B00151078
Steven Kelly,	B00150588
Rochelle Mullen,	B00156311
Piotr Momat,	B00156112

Submission date: 18/MARCH/2024

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of an Ordinary Degree in Computing in the Institute of Technology Blanchardstown, is entirely my own work except where otherwise stated.

Author: Habiba Nour

Dated: 18/MARCH/2024

Author: Steven Kelly

Dated: 18/MARCH/2024

Author: Rochelle Mullen

Dated: 18/MARCH/2024

Author: Piotr Momat

Dated: 18/MARCH/2024

1. Abstract:

The below document will outline the methodology, structure, and development plan of the pet service website we are building. Using PHP, CSS, HTML, and MySQL, we aimed to build a system that provides a user-friendly website that will allow them to book our services or allow businesses to offer the services. The website contains a booking system, a login/sign-up system, a review page, and a contact us page. This document will break down the website's structure, ERD design, UML design, user interfaces and use cases.

2. Project Definitions

2.1 The purpose of this document is to give an overall outlook and design structure of our website. Below you will see the design and implementation of the website including UML design, ERD design, database implementation and a general outline of the functionality of the website.

2.2 The project we are designing is a front-end website for a local pet walking, pet sitting and daycare business. Users will be able to log in, book the service they like or leave reviews. Business operators will be able to manage times and bookings. The administrator will have full privileges of the entire website.

2.3 The website has a few elements to get it to work, but our main priorities are ease of use and simplicity for the front end. We believe this is important as the user needs to have a clear understanding of the services provided and the layout of the site. Also, for ease of use, we have databases connected to the system to manage user and other site data. We are using MySQL to write the database, and PHP, CSS, and HTML are handling the scripting and front end of the website.

2.4 The software system has 2 main components, front-end and back-end. The front end is a simple design. It has three squares on the homepage which will bring the user to the area they would like (those being pet sitting, daycare and walking). When the user picks the service, they would like they will be brought to a booking page of that service. From the business end, they will be able to view their bookings and edit their available slots which site users with an account are able to see.

3. Document Revision

Rev. 2.0 Date: 18/MARCH/2024

4. Methodology

4.1 What is a system models – UML?

UML is a unified modelling language. It is a visual language that helps to visualise how the system will be designed. We used class diagrams and case diagrams to help us structure our system.

4.2 What is use of, and necessity of OOAD?

Object-Oriented Analysis and Design (OOAD) is crucial in software engineering for developing complex systems by breaking them down into small pieces. It enables the understanding of system requirements and defines objects, facilitating robust, scalable, and maintainable software.

4.3 What is a class diagram?

A class diagram is a visual representation of structure of the system you are creating showing the classes as well as the attributes and methods of these classes. It also shows the relationships between classes. Class diagrams can also be used to help develop ERD's needed for the database that the systems will need. Class diagrams can be used in many of the stages of developing software such as helping you find out what are the main components of your software and understand the requirements needed for the system.

4.4 Static Versus Dynamic Case Diagrams?

Static case diagrams are only show which entities interact with each. They don't show what occurs when this interaction happens. Dynamic diagrams do show what happens when they interact.

4.5 What is an ERD?

An Entity-Relationship Diagram (ERD) is a tool in database design, that shows the structure of data and how different entities relate to each other. Typically, we would use symbols such as rectangles to represent entities, diamonds to represent relationships, and lines to connect entities and relationships.

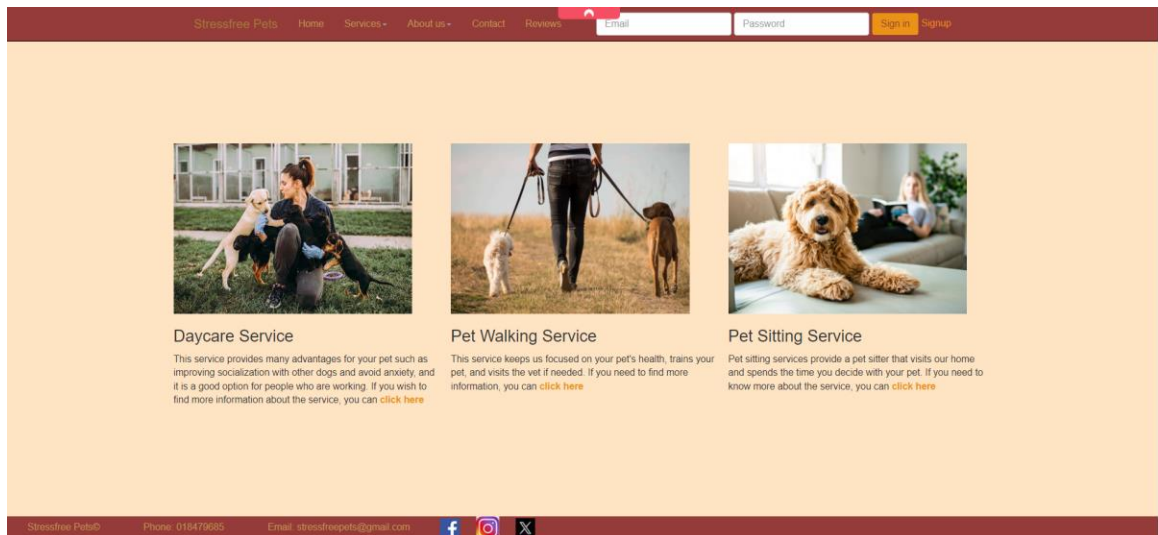
4.6 What is the purpose of using classes?

Classes are used to create objects. A class defines the properties, which are known as variables, and the methods, which are known as functions. Properties are variables that hold data related with an object. Methods are functions defined within the class. They are helping to perform actions related to the object.

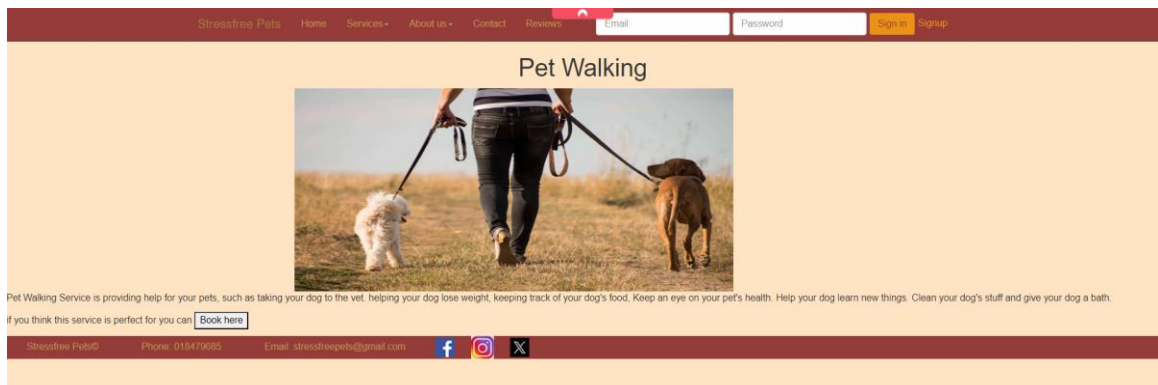
4.7 Volatile versus Persistent storage – Object Instances / Database?

Volatile storage refers to storage that retains data if the system is turned on. Persistent storage will retain data when the system is turned off. Most of the data on the site will be stored using persistent storage. We have a database built for the site which will contain information on users, administrators, the services provided etc. This is essential to the functionality of the system as it provides scalability and persistence to the website. A way to store data always is essential to a system like this as users need to be able to access and view their data whenever they like. If this data goes down if it was in volatile storage the functionality of the website would come into question. The data constraints and foreign key relationships ensure data will be updated correctly in the appropriate tables in our database.

As stated in the project definitions, the main goal of the system was to create a simple but functional layout and I believe this has been achieved.



Above is the home page of the website. This is very effective in functionality as it gives users a clear and defined idea of the sites purpose and provided services.



Above is what a user will see when they select one of the services (the other service pages are nearly identical). It is a very simple page just giving the user a small description of the service and a button to a page that allows them to book the service.

The image displays two registration forms side-by-side. The left form, titled 'Your Customer Profile', is for individual users and includes fields for First Name, Last Name, Address, Email Address, Phone Number, Dog Name, Dog Type, Dog Age, and an optional field for additional information about the dog. It also features an upload area for a dog image. The right form is for business providers and includes fields for Email address, Password, Business Name, Street address, City, County, Phone number, Service provided (a dropdown menu), Certification/Qualification, and an upload area for proof of certification/qualification. Both forms have a 'Submit' button at the bottom.

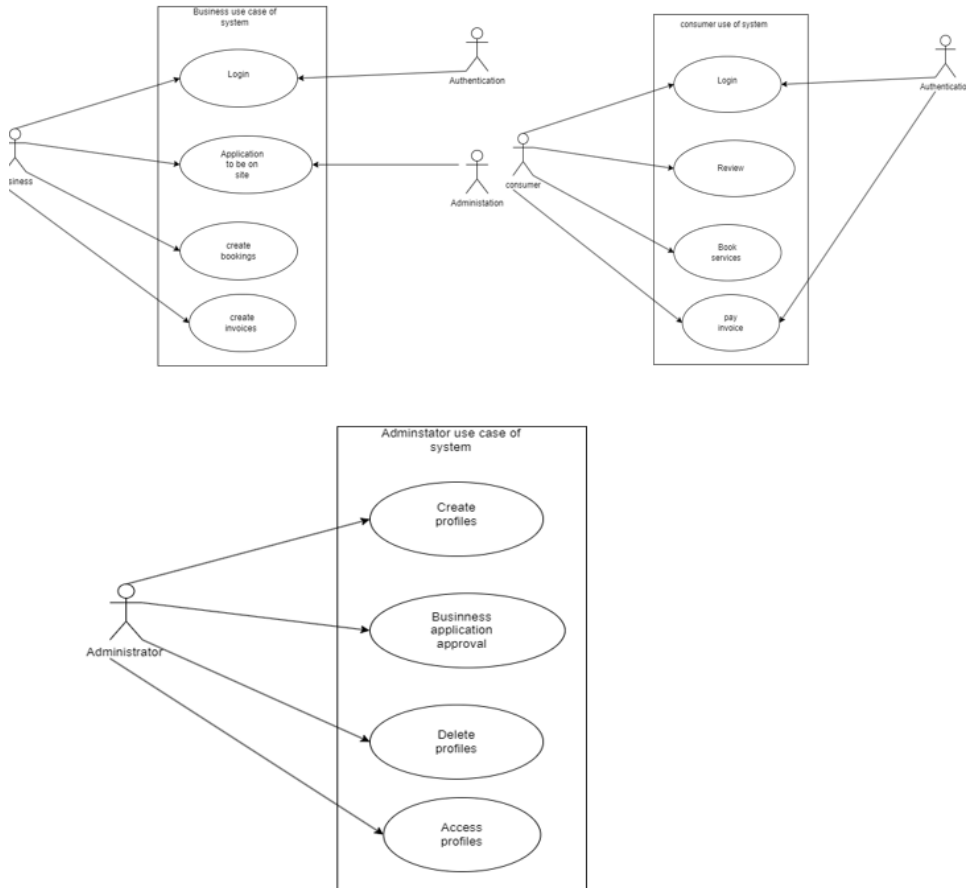
Above is two images, one of the user signup page and the other is the business signup page. The left image is the user signup form. This form authenticates the users data and allows them to create an account which will give them the access to book services, leave reviews and more.

The right image is the business signup form. While similar to the user signup the additional information required is the service they will be offering and proof of certifications/ qualifications. These will then be stored in the table for the business providers.

The simplicity of the UI is important to functionality and user use. This also assists us in data management as it allows our databases to have a relatively simple structure and make it easier for management.

5. Requirements:

5.1 Use Cases:



5.2 Use Case Specifications:

The classes that we created have methods in them, so we can use them for the connection between PHP and the database, and we created a class for cleaning, which allowed us to have sanitised and secure code and data. We used use case specification to help us know what table we should create and attributes.

table for customers

Attributes: custID, name, surname, petBreed, petImage, phoneNum, info, userID,

Table for user:

Attributes: userID, email, password

table for Freelancers/business:

Attributes: businessID, name, address, city, country, service, certs, img, userID

table for admin:

attributes: adminID, surname, name, userID

table for serviceType:

attributes: serviceName, description, img

table for daycare:

attributes: timeslot, datesAvailable, price, description, availableSlot, facilities, businessID

pet sitting:

attributes: timeslot, datesAvailable, price description, location, sottingInOut, additional, collectionOption

petWalking

attributes: walkingID, timeslot, datesAvailable, price, description, meetingpoint, business

invoice:

attributes: invoiceID, price, date, servicename, businessID, custID,

reviews:

attributes: reviewID, comment, stars, date, busiessI, custID

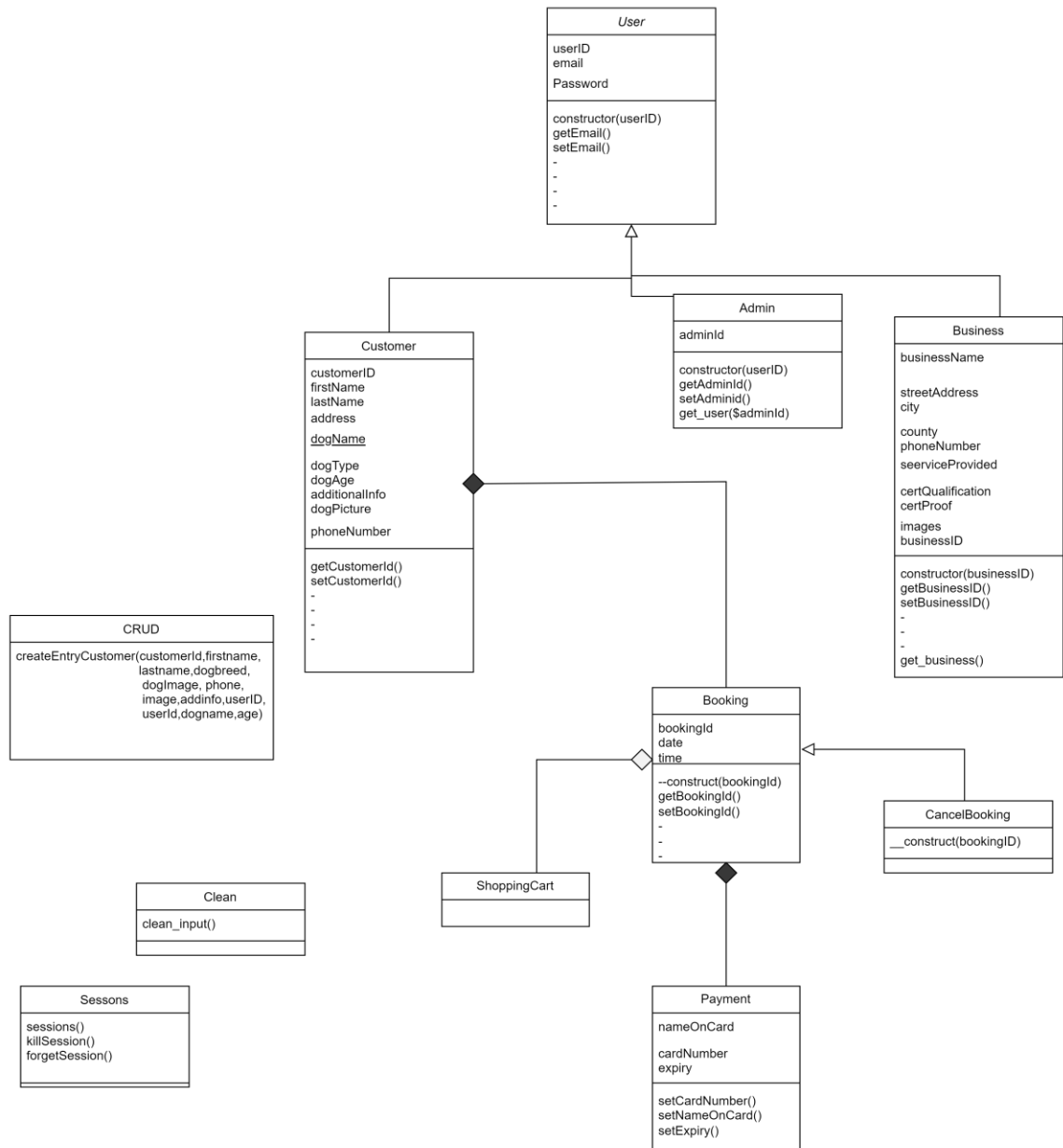
requestedProfile:

attributes: requestID, description, data, serviceID, custID, adminID

Use case Specifications was created when it wasn't clear what we wanted to put in the database therefore we added more tables and attributes

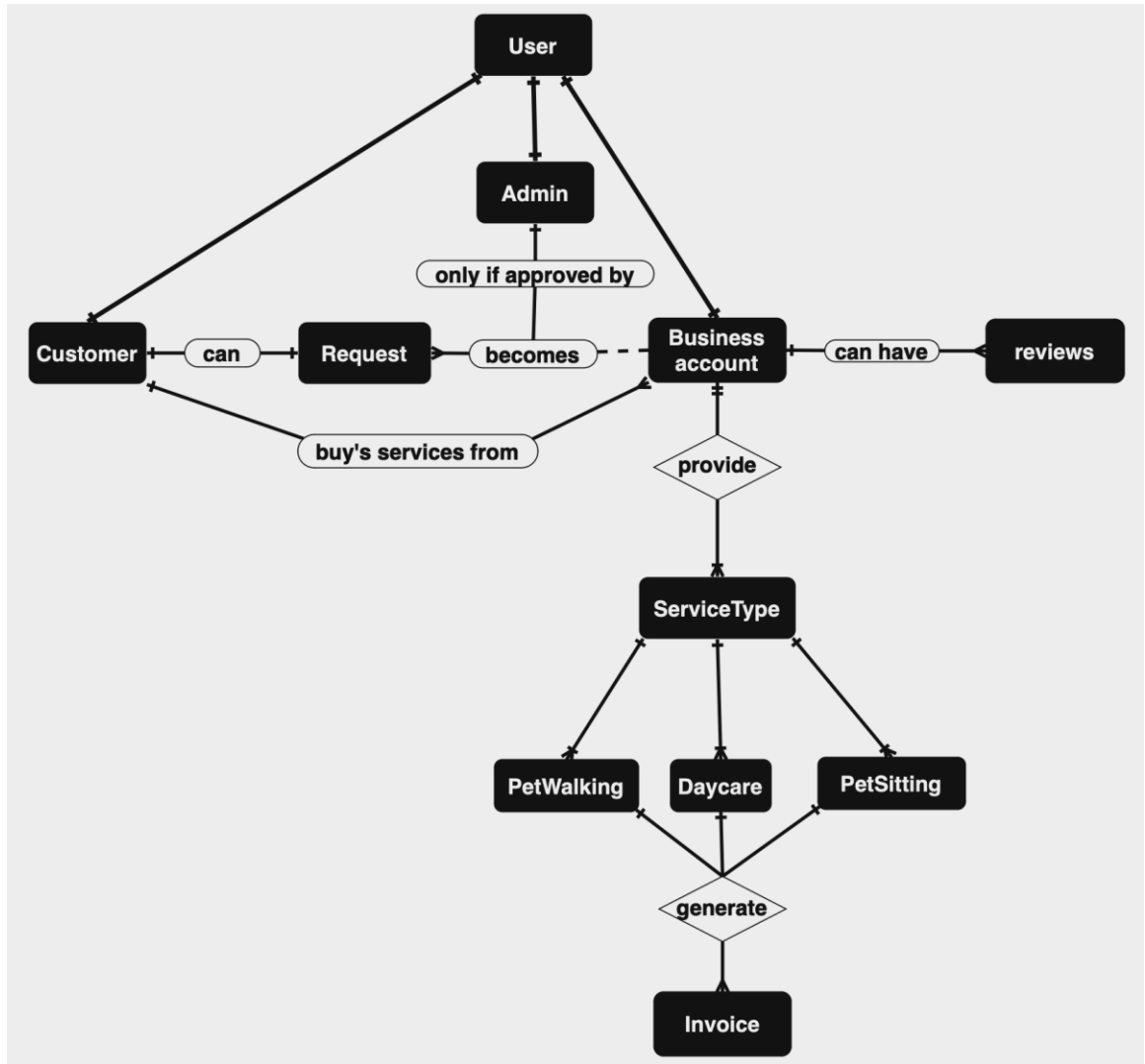
6. Case Diagrams:

Class Diagram – Show all relationships, multiplicities, associations, generalisations (inheritance), and aggregations (compositions) - See lecture 4.

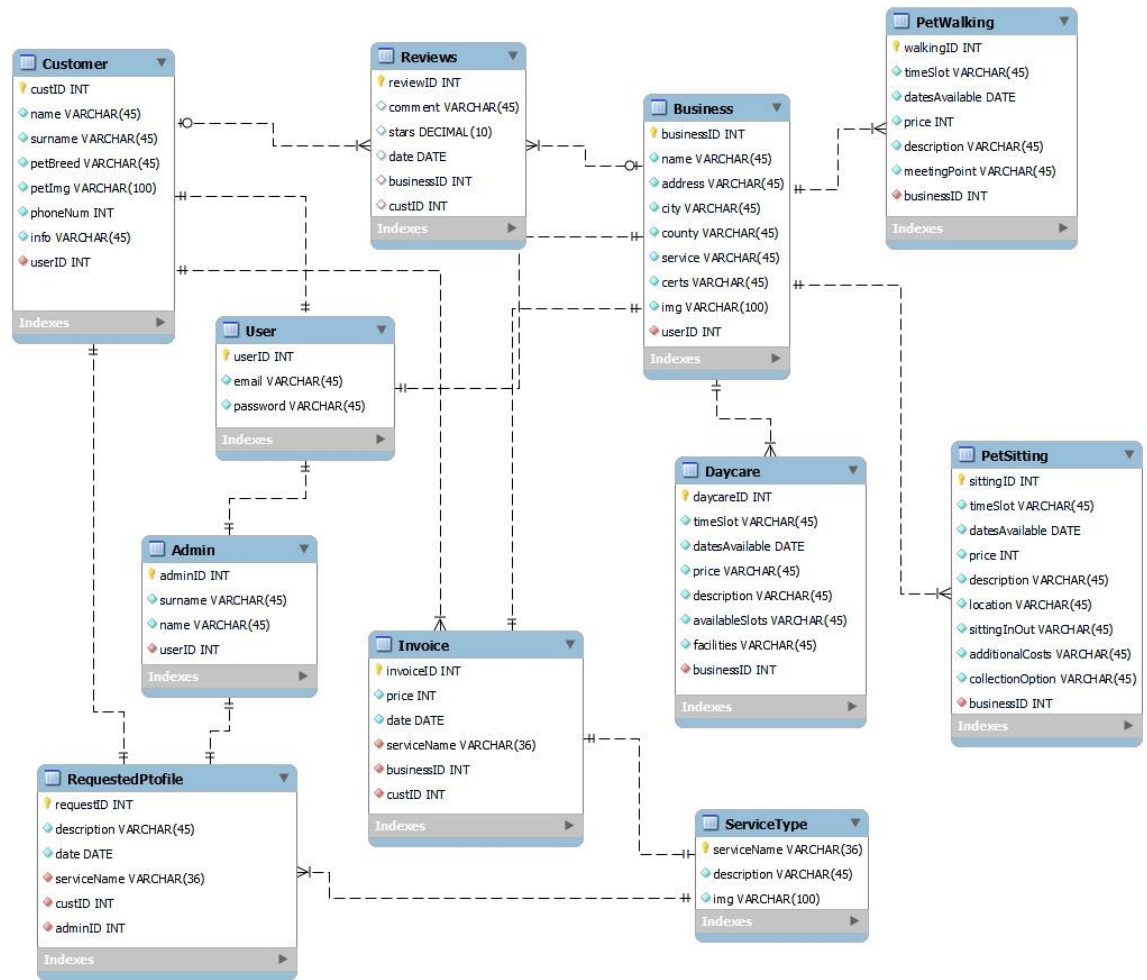


We chose to have a user class as it will be a parent class for all other users. These other users are business, admin, and customer. Each of these classes will inherit from user a userID. We did this to demonstrate inheritances. We also have a class called clean which will be used ensure any user input field cannot cause issues such as be used for SQL injections. Booking is used to make bookings and is a table in the database as well. Cancel booking inherits from booking. As the booking id will be need. We have a class called payment which will be used when booking. Sessions will be used to ensure that people cannot book without being logged in. the shopping cart will be used so that customer booking can be temporarily stored.

Conceptual Diagram of Database:



Physical ERD of DATABASE:



7. Conclusions

7.1 We have progressed a lot with our project. Successfully developed the database structure and class diagrams which gave us a clear system architecture. The front-end interface has most of the website functionalities, the same as the back end, where we created tables, defined relationships and designed ERDs.

7.2 Minor adjustments were made to the original proposal where we optimised data management like DB tables and new variables, front end has no changes.

7.3 Overall, the project has not changed a lot in architecture. However, we have improved a lot in the development of the website by sticking to our original goals. We're dedicated to adding the rest of the features and testing everything properly to make sure our pet service website works great.