

AÇIKLAMA

1) Giriş

Bu raporda BIL372 Veritabanı dersinin projesi için yapılan çalışmalar, ön hazırlık aşaması, geliştirme süreci, geliştirilen uygulamanın detayları, proje süresince karşılaşılan zorluklar, uygulamanın çalışma prensipleri ve sonuç olarak bahsi geçen projede rol alan bireylere ne gibi kazanımlar sağladığı açıklanmaktadır.

Öncelikle proje konusu olarak bir veteriner kliniğinin ihtiyaçlarını karşılayacak bir web uygulaması yapılmasına karar verildi. Uygulama kısaca özetlenecek olursa, bu uygulama bir veteriner kliniğinin uzun vadede klinik içinde ihtiyaçlarını karşılayacak, temel olarak hasta takibine dayalı web tabanlı bir veritabanı uygulamasıdır. Söz edilen uygulamayı geliştirmeye başlamadan önce bir veteriner kliniğine gidilerek konu ile ilgili uzmanların fikri alındı. Uzmanların görüşlerine ve tavsiyelerine dayanarak yazılım ile çözülebilecek gündelik sorunları, hayallerindeki ve “keşke şu an kullandığımız yazılımda şöyle bir özellik olsaydı.” şeklinde tabir ettiği nitelikler ve rutin olarak uzun ve kısa vadede neler yaptıkları not edildi. Uzman görüşleri ve alınan notlar sayesinde projenin miniworld assumption’ını tasarlama adımı tamamlandı. Zaman geçtikçe miniworld assumption üzerinde değişiklikler yapıldı. Daha sonra miniworld assumption’a uygun bir biçimde ER/EER şemaları tasarlandı. Sonraki adımda relational mapping yapılarak database schema diagram ortaya çıkarıldı. Ardından uygulama geliştirilmeye başlandı. Geliştirme aşamasında çeşitli zorluklarla karşılaşıldı. Bu zorluklardan bir sonraki bölümde bahsedilecektir.

2) Karşılaşılan Zorluklar

İlk olarak proje yapılırken proje konusuyla ilgili herhangi bir grup üyesinin fikri yoktu. Bu yüzden bir uzmana danışılması icab ediyordu. Bu nitelikte birini bulurken zorlanıldı. Fakat bu kişiyi arama süreci başarılı geçti ve en sonunda gerekli nitelikte bir uzman bulundu ve görüşleri alındı.

Karşılaşılan ve en fazla zaman kaybına sebep olan zorluk ise teknik yetersizlikti. Genel olarak grup üyeleri projede kullanılacak olan araçları kullanmaya yönelik yetkin bilgiye sahip değildi. Bizi en çok bu durum zorladı ve neredeyse

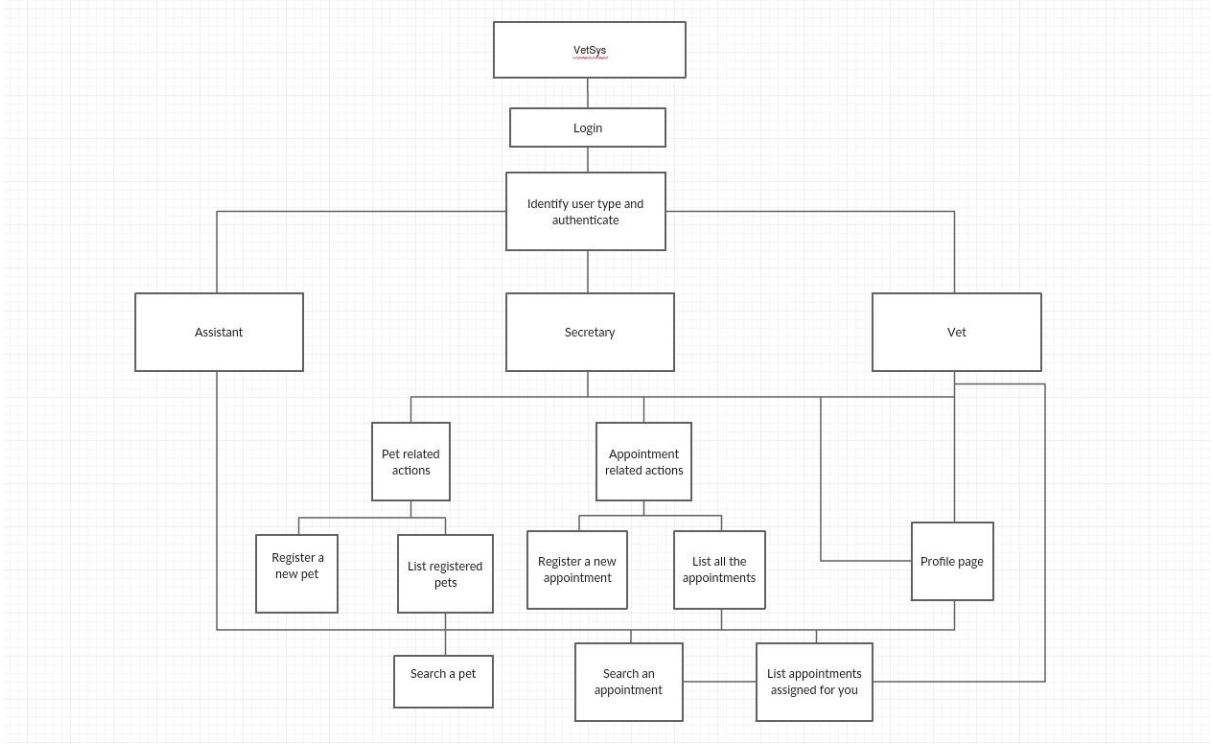
kullanılan araçların tümü sıfırdan öğrenildi. Bu durum dolayısıyla fazlasıyla zaman kaybına yol açtı.

3)Kullanılan Araçlar

Uygulamayı geliştirirken birçok web geliştirme yazılımından yararlanılmıştır. Aşağıda kullanılan yazılımların listesi yer almaktadır:

- Python 3.x
- Flask
- SQLAlchemy
- PyCharm, Atom, VSCode
- git, GitHub
- HTML,CSS,JavaScript
- JQuery
- AJAX
- Jinja Templates
- Virtual Environments

HIPO



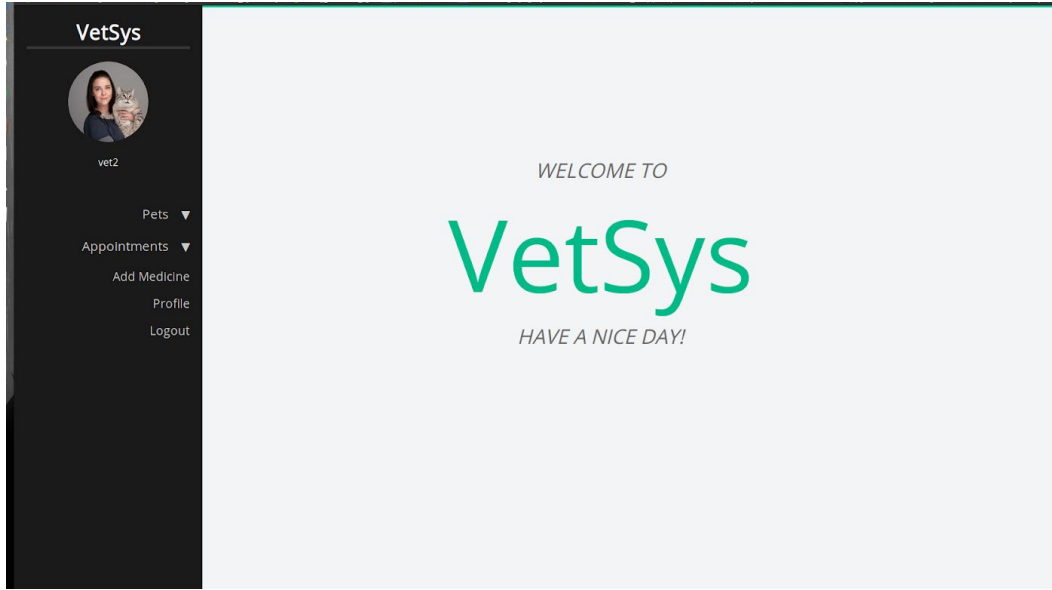
ARAYÜZ VE EKRAN GÖRÜNTÜLERİ

Tasarlanan yazılım server-client ilişkisiyle çalışmaktadır. İlk olarak client'lar web browser üzerinden sisteme '/login' yönlendirmesiyle giriş yaparlar.

The login screen for VetSys features a light gray background with a central white rounded rectangle containing the login form. The form includes the VetSys logo, a Username input field, a Password input field, and a green Login button.

Sistemde 4 farklı kullanıcı bulunmaktadır. Bunlar veterinerler, temizlik elemanları, sekreterler ve asistanlardır. Kullanıcı 'username' ve 'password' alanlarını doldurduktan sonra 'Login' butonuna tıklar.

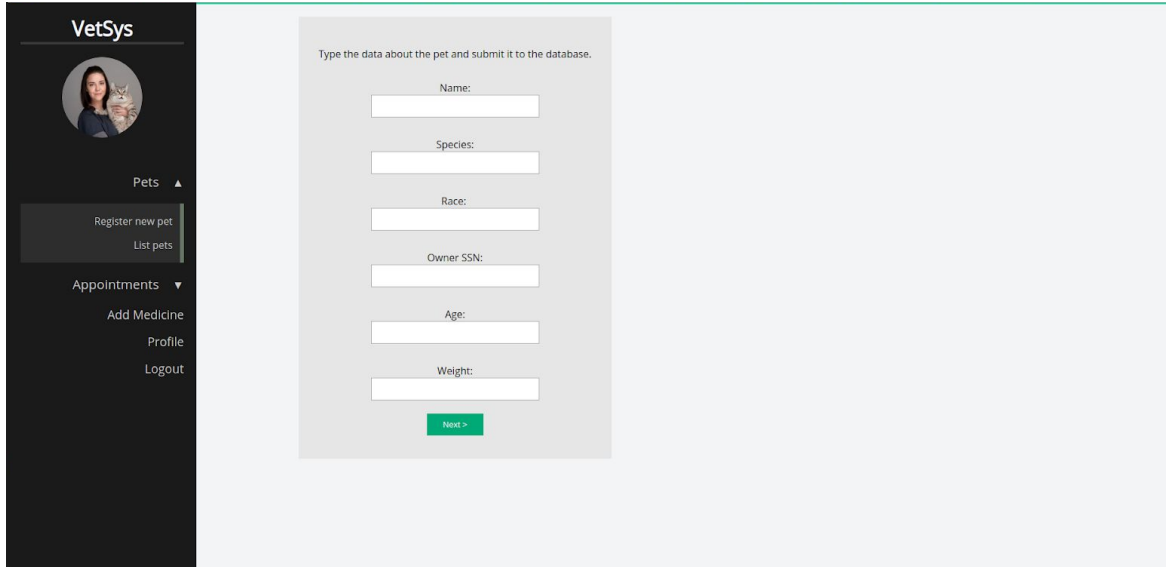
Ardından sistem ilgili kullanıcı tipine göre bir dashboard sayfasını kullanıcıya döner. Yani sekreter ile veterinerin dashboard'ları farklıdır. Bu aşamadan sonra kullanıcılar kullanıcı tiplerine göre aşağıdaki gibi bir dashboard göreceklerdir.



Burada solda kullanıcının ihtiyaçlarına göre bir panel tasarımı mevcuttur. Bu panelde randevular, pet'ler profil sayfası ve sistemden çıkış yapmak için gerekli yönlendirmeleri yapacak bölgeler bulunmaktadır.

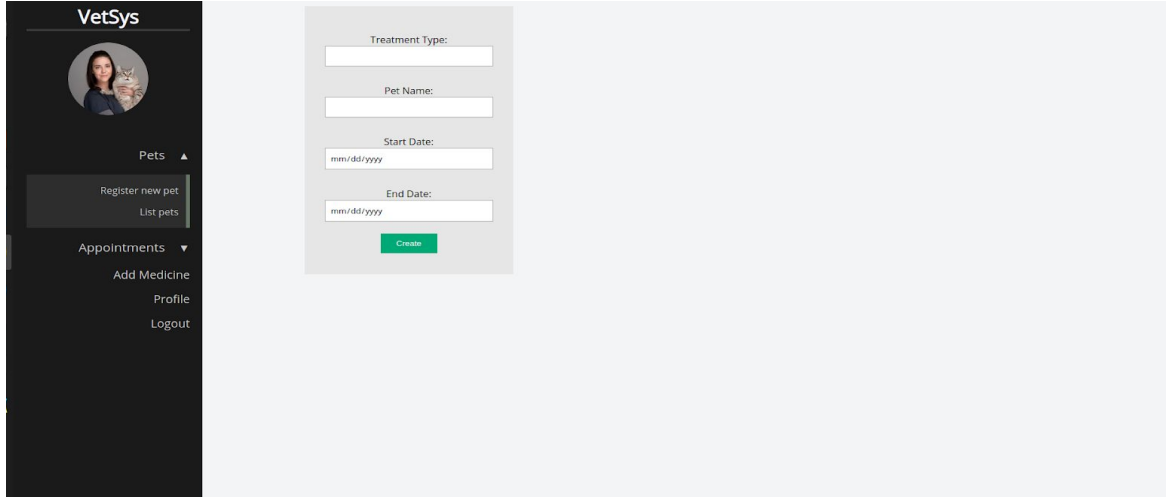
1) Pets

Pets alanında 2 adet alt alan mevcuttur. Bunlardan ilki yeni gelen hastayı sisteme kaydetmeye yarayan 'Register new pet' alt alanıdır. Buraya tıklanıldığında aşağıdaki gibi bir bölüm açılmaktadır. Burada gerekli bilgiler girildikten sonra 'Next' butonuna tıklanır ve sonraki forma geçilir.



The image shows a web application interface for 'VetSys'. On the left is a dark sidebar with a user profile picture and a list of navigation items: 'Pets' (expanded), 'Register new pet', 'List pets', 'Appointments', 'Add Medicine', 'Profile', and 'Logout'. The main content area is light gray and contains a form titled 'Type the data about the pet and submit it to the database.' The form has input fields for 'Name:', 'Species:', 'Race:', 'Owner SSN:', 'Age:', and 'Weight:'. At the bottom of the form is a green button labeled 'Next >'.


Bu formda ise hastaya yapılan tedavi bilgileri girilir ve 'Create' butonuna basılarak yeni tedavi, bu tedavinin ait olduđu hastaya kaydedilir.



The image shows the same 'VetSys' web application interface. The sidebar is identical. The main content area now displays a form for adding a new treatment. The form has input fields for 'Treatment Type:', 'Pet Name:', 'Start Date:' (with a date picker icon), and 'End Date:' (with a date picker icon). At the bottom of the form is a green button labeled 'Create'.

Eğer hali hazırda mevcut olan bir hastaya yeni tedavi eklenecekse bir önceki açılan formda 'Name' kısmına hastanın ismi girilir. Name bölgesine her harf girildiğinde, server'a bir request atılmakta ve hastanın ismini veritabanında aramaktadır. Örnek olarak Oreo adlı hastaya bir tedavi yapılsın ve hastanın tedavi geçmişine son yapılan tedavi eklenilmek istensin. Bu durumda 'Name' kısmına 'Oreo' yazdığınızda aşağıdaki gibi bir sonuç elde edilecektir.

VetSys



Pets ▼

Appointments ▼

Add Medicine

Profile

Logout

Type the data about the pet and submit it to the database.

Name:

Oreo

Species:

Race:

Owner SSN:

Age:

Weight:

Next >

Search Results:

(you can select any result in order for the form complete itself)

Id: 6,

Name: Oreo,

Owner-SSN: 123456,

Age: 123,

Weight: 12,


Species: asdasdj,

Race: smdasdn

Görüldüğü üzere sistemde aranılan hastanın bilgileri bulundu. Sağ taraftaki arama sonucunun üstüne tıklanıldığında ise geri kalan bölgeleri sizin için otomatik olarak dolduracaktır ve sıradaki 'Tedavi kaydetme' formuna geçilecektir. Tedavi formunda gerekli bilgiler girildikten sonra 'Create' butonuna tıklanırsa sistemde var olan hastanın tedavi geçmişine yeni yapılan tedavi eklenir.

İkinci alan ise 'List pet' bölgesidir. Bu sayfada sistemde kayıtlı tüm hastaların bilgilerini topluca görebilmekteyiz. Kullanıcı 'List pet' tıkladığında aşağıdaki sayfa ile karşılaşacaktır.

VetSys



Pets ▼

Appointments ▼





Add Medicine

Profile

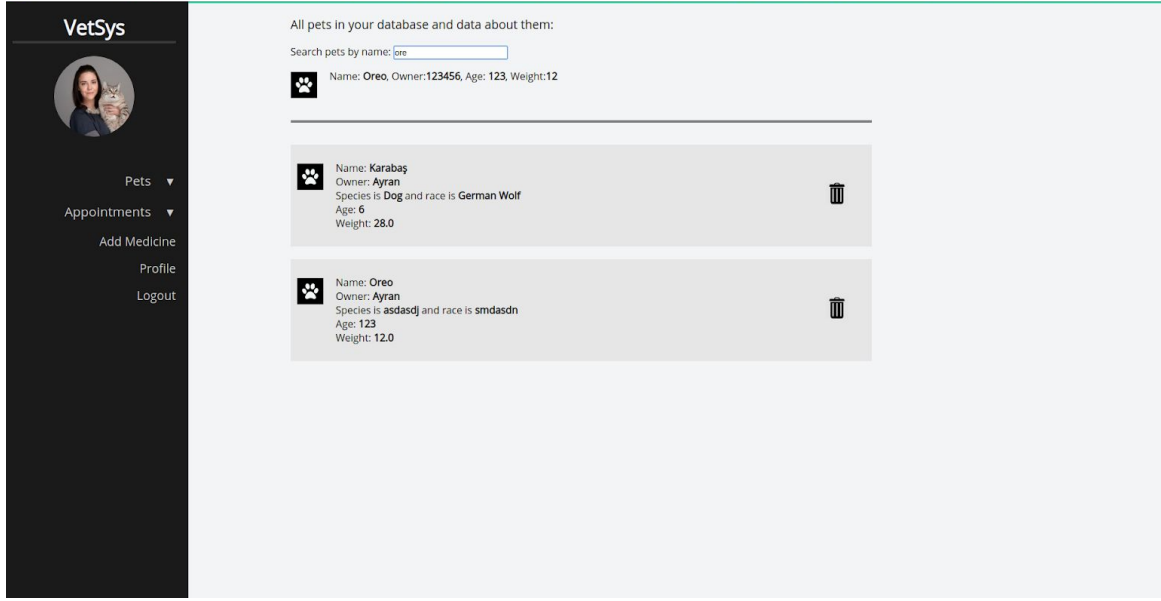
Logout

All pets in your database and data about them:

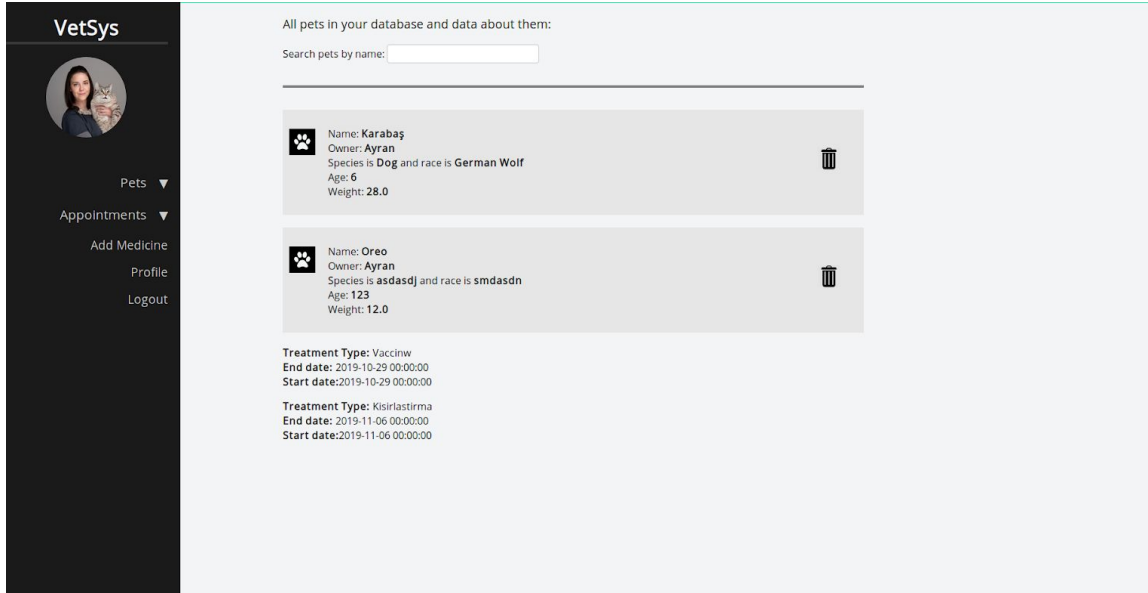
Search pets by name:

	<div><div>Name: Karabaş</div><div>Owner: Ayran</div><div>Species is Dog and race is German Wolf</div><div>Age: 6</div><div>Weight: 28.0</div></div> <div></div>
	<div><div>Name: Oreo</div><div>Owner: Ayran</div><div>Species is asdasdj and race is smdasdn</div><div>Age: 123</div><div>Weight: 12.0</div></div> <div></div>

Burada en üst kısımda bir search bar'ı mevcuttur. Eğer sistemde çok fazla hasta mevcutsa kullanıcı aradığı hastayı bulmakta zorlanabilir. Bu durumda hastanın ismini search bar'a yazılmasıyla aranılan hastanın bilgilerine ulaşılabilir. Search bar'ın kullanımı aşağıdaki gibidir.



Ayrıca burada koyu gri alanlara tıkladığınızda hayvanların tedavi geçmişini görebilmekteyiz. Aşağıda bu durumun bir örneği gösterilmektedir.



Yine aynı şekilde her hasta kaydının sağ tarafında çöp kutusu şeklinde bir sembol bulunmaktadır. Buraya tıklanıldığında hastanın tüm kayıtları sistemden silinmektedir.

2)Appointments

Bu bölümde veterinerlerin ihtiyaç duyduğu bir çeşit randevu sistemi gerçekleştirilmiştir. İlk olarak 'Appointments' kısmının altında 'Register new

appointment' alt bölgesinin mevcut olduğunu görüyoruz. Buraya tıkladığınızda karşınıza aşağıdaki gibi bir sayfa açılacaktır.

VetSys

Pets ▼

Appointments ▲

Register new appointment

List appointments

Add Medicine

Profile

Logout

Owner: 123456

Date: 11/29/2019

Time: 10:00 AM

example: 10:00 AM

For once ☒ Periodic ☐

Period:

Create

Appointment created successfully

Burada bir randevu oluşturmak için gerekli olan bilgiler girilmesi gerekmektedir. Eğer sistemde var olan bir 'Owner' a yeni bir randevu eklenecekse, randevu yazılacak kişinin ismini Owner bölgesine yazdığınızda hemen sağ tarafta sistemde kayıtlı kullanıcının bilgileri aranarak getirilmektedir. Search result'a tıklayıp gerekli bilgileri girip 'Create' butonuna tıkladığınızda yeni girilen randevu bu kişiye atanmaktadır.

VetSys

Pets ▼

Appointments ▼

Add Medicine

Profile

Logout

Owner: Ay

Date: mm/dd/yyyy

Time: 10:00 AM

example: 10:00 AM

For once ☒ Periodic ☐

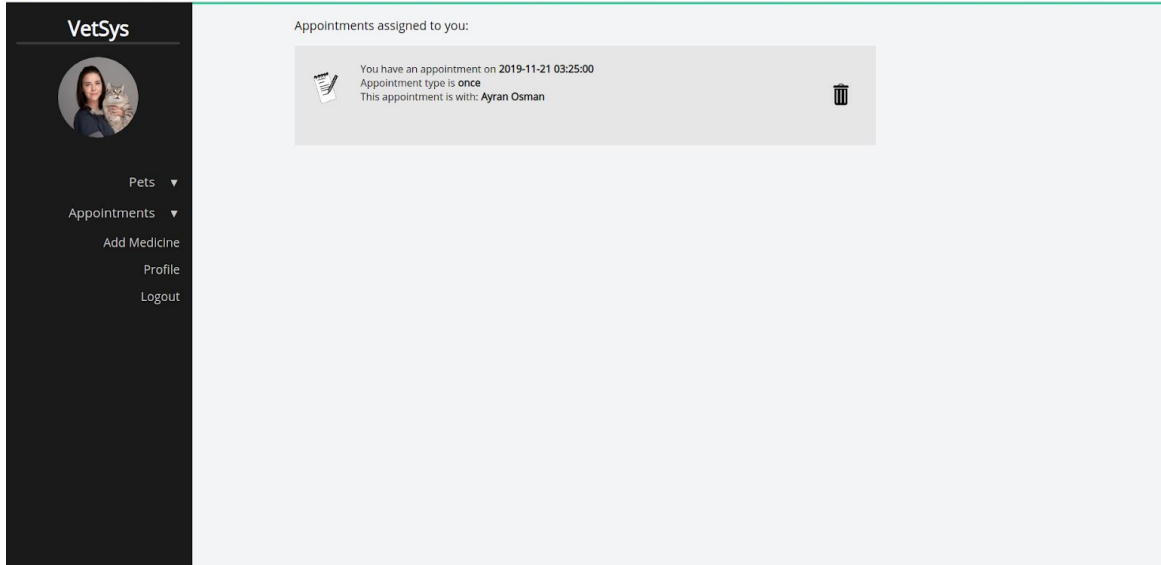
Period:

Create

Search Results:
(you can select any result in order for the form complete itself)

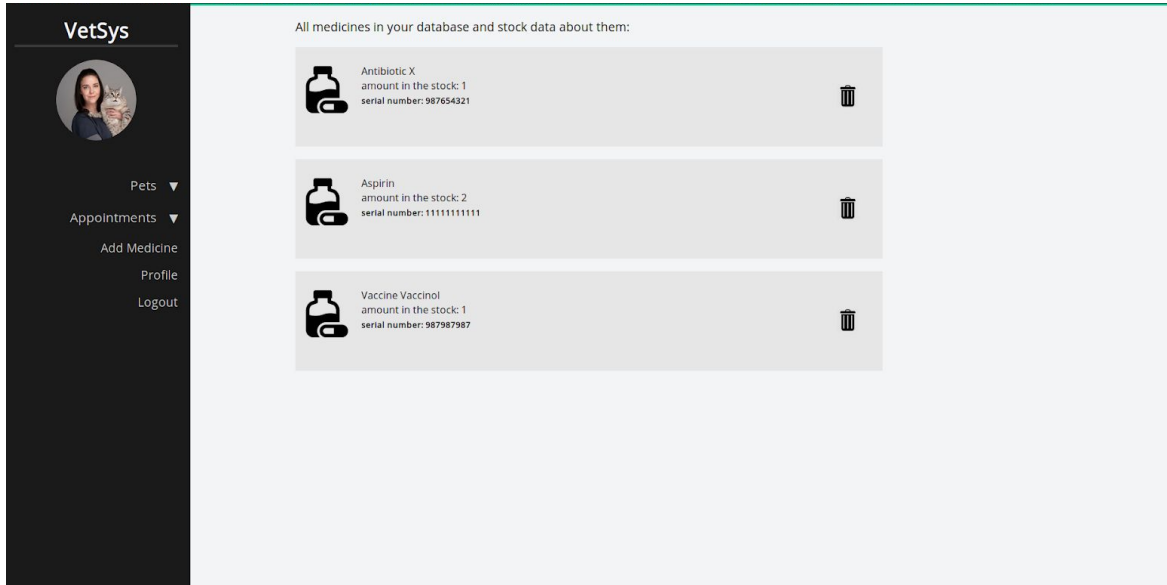
SSN: 123456,
Name: Ayran,
Last_name: Osman,
Phone Number: 57853

'Appointments' ın diğ er bir alt b lgesi olan 'List appointments' kısmında ise kullanıcının (veterinerin) kendi randevuları g z kmektedir. Buraya tıkladıđınızda ařađıdaki gibi bir sayfa a ılmaktadır.



Resimde g r ld đ   zere her randevu kaydının yanında silme butonu bulunmakta. Bu butona tıklanıldıđında kaydın t m bilgilerini sistemden silmektedir.

Ayrıca hen z sol panele eklememiđimiz fakat URL olarak sistemde mevcut olan sistemdeki ila  stođuna g steren bir sayfa mevcuttur. Ařađıda bu sayfayı g rebilirsiniz.



Profil kısmında ise kullanıcıların kiřisel bilgilerinin bulunduđu bir sayfa olarak tasarladık. Fakat zaman kısıtından ve  eřitli bazı sebeplerden dolayı yapım ařamasında kaldı.

Son olarak sol tarafta görülen 'Logout' butonuna tıklayarak sistemden çıkış yapılmaktadır.

SONUÇ

Projenin süresi boyunca grup üyeleri kullanılan araçlar hakkında en az temel seviyede bilgi edildi. Özellikle Python, Flask ve SQLAlchemy konusunda çok ciddi kazanımlar oluştu. Projenin başında kimsenin bu araçlar hakkında herhangi bir bilgisi yokken şuan geline nokta bu yazılımların tümünü iyi şekilde kullanılacak düzeye ulaşıldı. Derste öğrenilen teorik bilgiler üretime geçirilebilir bir proje olarak geri dönüt verdi. Projenin bağlı bulunduğu ders veritabanı dersi olmasına rağmen web uygulamaları hakkında bilgi edinildi ve pratiğe geçildi.

Bu kazanımlara rağmen seçilen proje için bazı kısıtlar bizi zorladı. Basitçe her şey sıfırdan öğrenildi. Bu durum dolayısıyla zaman kaybına yol açtı. Server tarafında oluşturulan veritabanı tam olarak istenildiği gibi çalışmasına rağmen web uygulamalar konusundaki yetersizliklerden ötürü database fonksiyonelliğinin sadece bir kısmı client tarafına aktarılabildi.

Sonuç olarak bilgi ufkumuzu genişleten ve takım koordinasyon niteliklerimizi geliştiren geniş kapsamlı bir proje yapmaya çalıştık. Projenin bir kısmı tamamlandı. Fakat platform kısıtlarından ötürü tam verimli bir şekilde çalışan bir sistem henüz oluşturulamadı.

EK

Project Miniworld Assumption

This is a model of veterinary clinic tracking system. In a veterinary, there are plenty of data needs to be tracked. This miniworld assumption can be revised while our group is working on it.

PET

A veterinary clinic examines pets. Each pet has a unique **ID**, **name**, **species**, **race**, **weight** and **age**. Vets may want to know if the pet for which the vet responsible has any **disabilities** so that he/she may decide the pet's treatment process according to its disability condition. During its treatment, pets are placed into a suitable **cage number** of the clinic if it is necessary. At the end of the treatment, vets may want to take some **notes** about the pet. Thus, vets can see some beneficial information about pet if a vet who is different from the previous one is assigned to cure for the same pet. At the end of the treatment, information of the **owner** of the pet is required because vets or staff may need to contact owner for any reason.

OWNER

Most of the pets that the vet examines have an owner. Each owner has the following informations: **name**, **e-mail**, **sex**, unique **ID**, **phone number**, and **address**. The veterinary take these information because they might need to contact the owner about the pet's needed vaccines, any other routine tasks or in emergencies.

TREATMENT RECORDS

A treatment record is a treatment history of a pet that has treated at the veterinary clinic. **Start date** and **end date** of each treatment is recorded. These time-stamps are both necessary, since a treatment process may take an unpredictable amount of time. Also, a pet may need to go to the veterinary clinic periodically. Vets decide which medicine or treatment will be applied to patient according to its treatment schedule. Tracking of these dates is crucial and beneficial both for the owners and vets.

In addition, what **kind of treatment** is a datum about the examination if it is a wound intervention, vaccine, periodic, etc. **Name of the procedure** is also important input (for instance “parasite vaccine”). Since vets want to know which treatment is applied to the pet in the past or now so this information should be held. A vet should be able to track **meds** used for treatment of a pet. Meds is medical name of drugs (for instance “antibiotic x”). Beside treatment name and type, meds should be recorded because drugs are part of the treatment.

MEDICINE

Drug stock is fundamental for a veterinary clinic. Most of the time a treatment in veterinary clinic requires some meds. Drugs may be needed anytime. Therefore, records of medicines should be kept in a veterinary clinic. As you can guess that these records contain information such that **medicine name**, unique **serial number**, unique **barcode number** and **expiration date**. If the medicine expires, then it must be removed from stock. In such cases, stock manager may order the same drug . In this case, stock manager has to contact with the **distributor** of medicine. That means contact information of distributor of the drug should be saved. Also, when a drug is out of stock, which means the remaining amount is zero, stock manager does the same.

CAGES

In some cases, pets coming to the vet have to stay there for their treatment. Therefore, most veterinary clinics have cages for inpatients. Such cages may be needed not only for treatment, but for an owner who is looking for a temporary home. A veterinary clinic must track condition and situation of their cages. For example a cages' **suitability** situation must be held. If a vet or a staff want to place a pet into a cage, he/she must first look at cage's suitability. Because a horses' cage is not the same as a dogs' cage. Also, the cage should not be full of **capacity**. There must be a **room** for a new pet in the cage before placement. In some cases, vets may take **notes** about a cage. For example, an epidemic stayed in the cage for a while. In such case, vets warn the staff so that they should sterilize the cage before any other placement.

STAFF

The veterinary clinic is a regular business. In other words, as in every enterprise, there are many employees in various **positions**. Veterinarians who perform treatment and cleaning staff who are responsible for hygiene in clinic are only some of the staff. Each position has specific **permissions**. As an example, vets and laboratory assistants are the only members who are allowed to enter the laboratory. Each **staff member** is assigned to one of the clinics. Every staff member has **salary** and monthly gets paid according to their **working hours** and **position**. If a staff member **temporarily leaves** from work, his/her salary is deducted.

CLINICS

A veterinary business may have several clinics in certain **locations**. Specifically, if a veterinary business is large scale enterprise, this is probably the case. Company members have to work at one of these clinics. Also, staff at clinic A may be assigned to another clinic B in time by a person in charge of the clinic A. In some cases, it is important to know **clinics contact information**. For example, if a clinic doesn't have a medical tool, a vet working at this clinic may request the tool from other clinics or

directs the patient to another clinic that has the medical tool. As another example, if a pet owner looks for temporary home and comes to one of the veterinary clinics. However, there is no suitable or empty cage for the new pet. A vet can contact other clinics to get information if there is an empty cage for the pet. After that, the vet directs customer to other clinic if there is any empty cage.

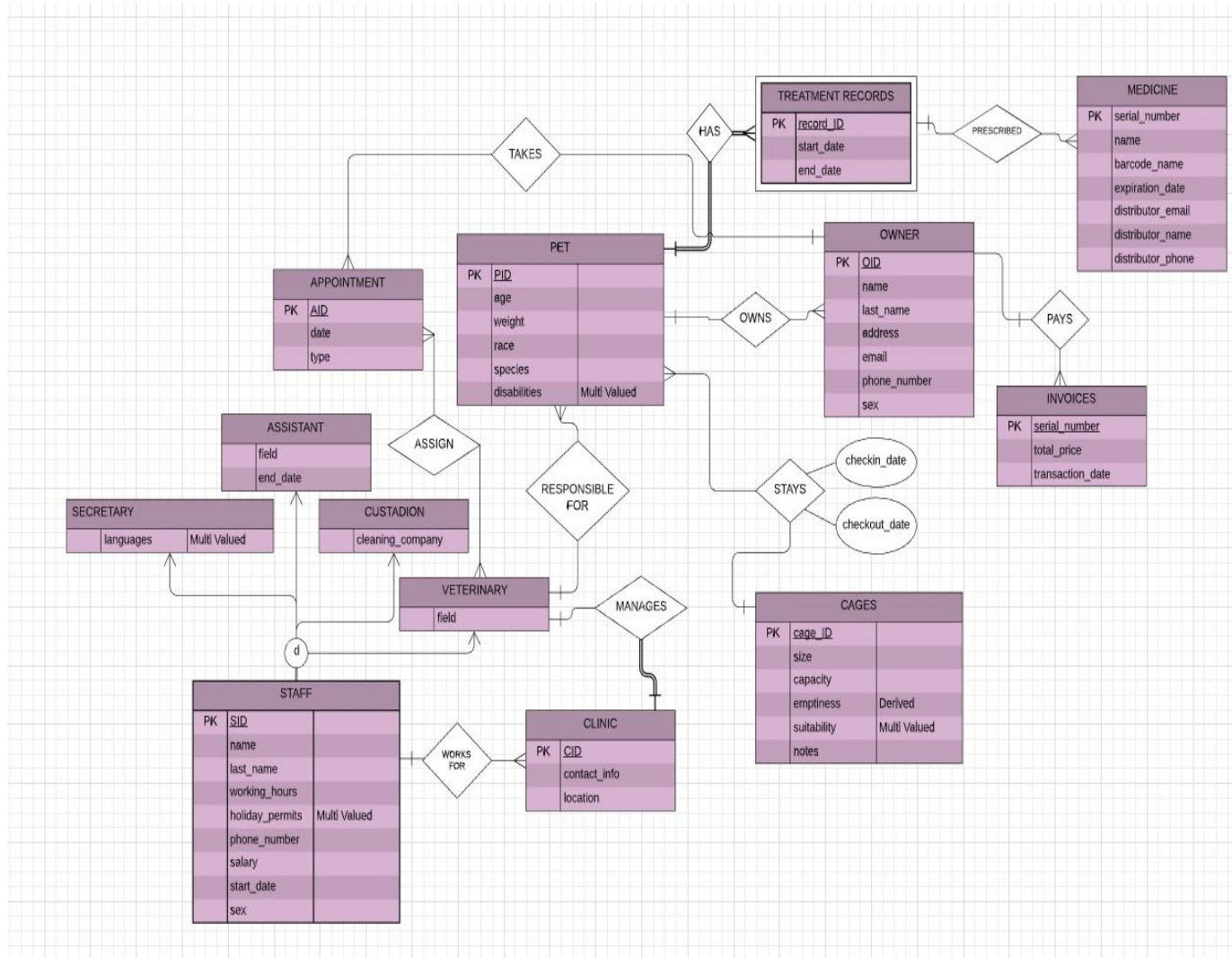
INVOICES

Customers of veterinary gets a bill. A bill has to be given to a customer lawfully and contains these kind of information: **customer information, quantity, service type, service cost, the total price, transaction date, serial number**. Bills have to be stored because they are required by economy related departments of veterinary business. For example, accountants of veterinary frequently works on invoices.

APPOINTMENT

In order to cure a pet, a pet owner has to make an **appointment** before going to a clinic if it is not an emergency situation. Each appointments are **assigned to a vet** according to the **treatment type** and **other parameters**. For example, if an appointment is made for a bird, it is assigned to a vet whose area of expertise is bird species. Also, the owners should be able to choose the vet and a **specific date and hour** for examination. At the end of the examination, the patient may need to come again for its ongoing treatment. Especially, most of vaccine type treatments are monthly or weekly. If a periodic treatment concerned, extra information such that **period** is needed to hold. When the appointment day comes, somehow pet owner has to be notified. Therefore, **owners' contact information** also has to be stored.

Project ER/EER



Project Schema Diagram

