B5814985 นายเฉลิมชัย เหลืองสกุลไทย
Sec01 Group 6 ระบบคลีนิคสัตว์เลี้ยง

1.Requirement

ระบบคลินิกรักษาสัตว์

ระบบย่อย ระบบการจองนัดหมาย

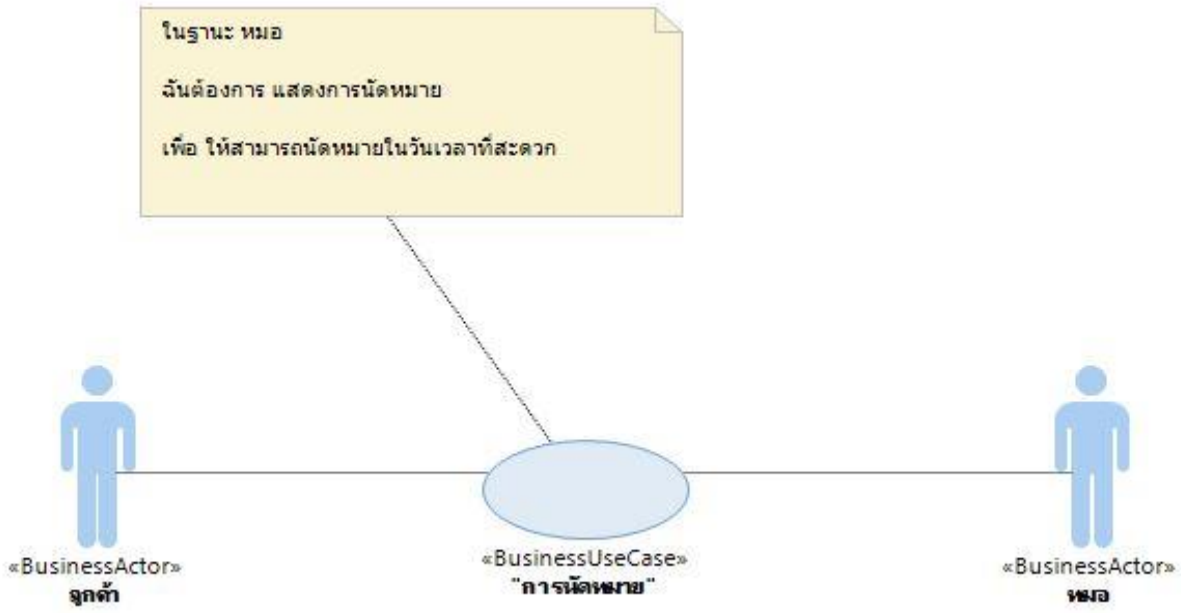
ระบบการจองนัดหมาย ระบบแจ้งเตือนเมื่อถึงวันนัดหมาย และ สามารถดูข้อมูลการนัดหมาย ล่วงหน้า และ ย้อนหลังได้ตลอดเวลา บันทึกข้อมูลการนัดหมายของสัตว์ป่วย หรือ รายละเอียดของสัตว์เลี้ยงและลูกค้า ของวันที่นัดว่ามาทำอะไรบาง ช่วงเวลาไหนของวันที่นัด เพื่อให้ง่ายในการติดตามสอบถาม

ในฐานะของ : หมอ

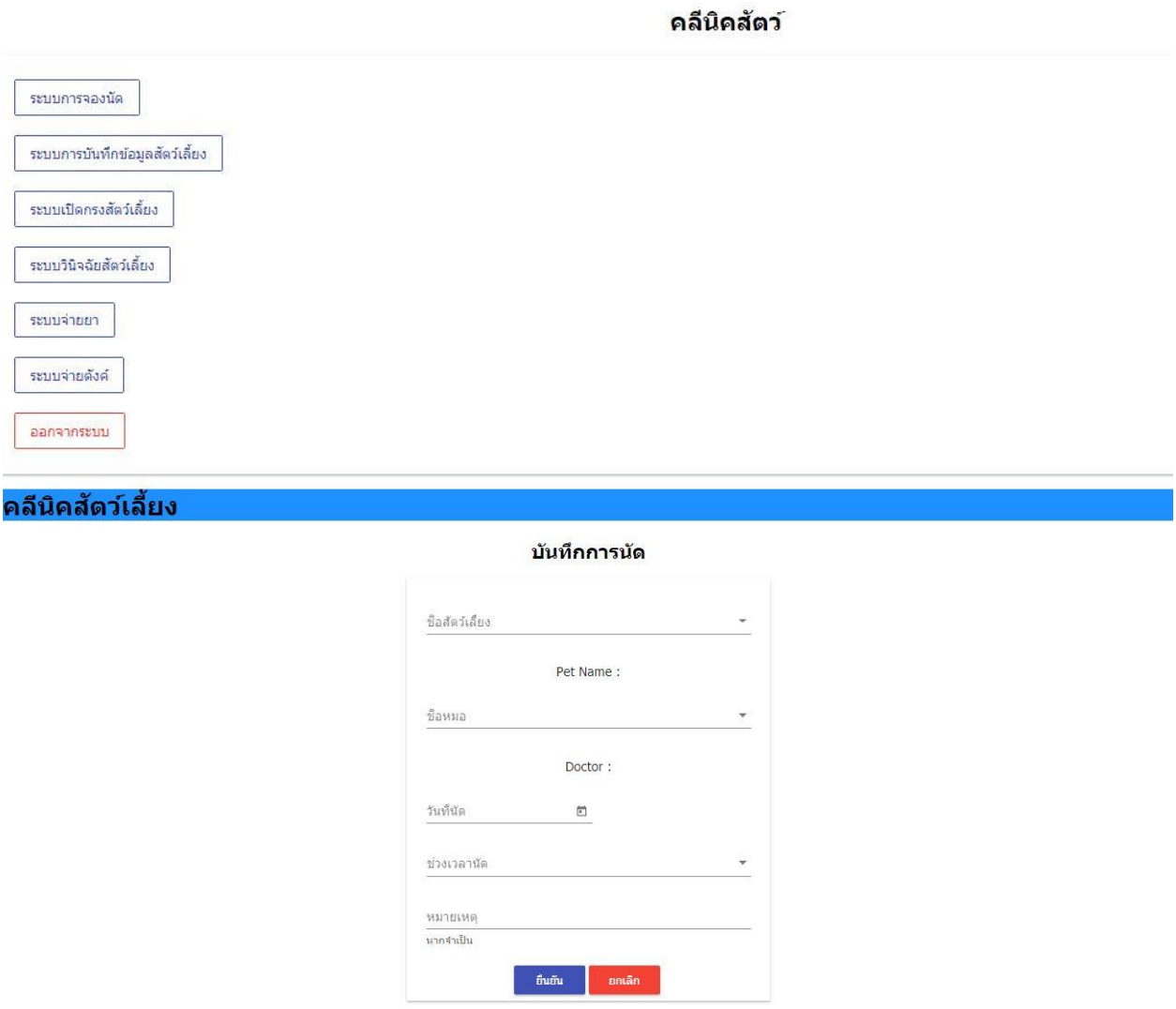
ฉันต้องการ : แสดงการนัดหมาย

เพื่อให้ : เพื่อให้สามารถนัดหมายในวัน ช่วงเวลาที่สะดวก และ จะได้ทราบข้อมูลของลูกค้า

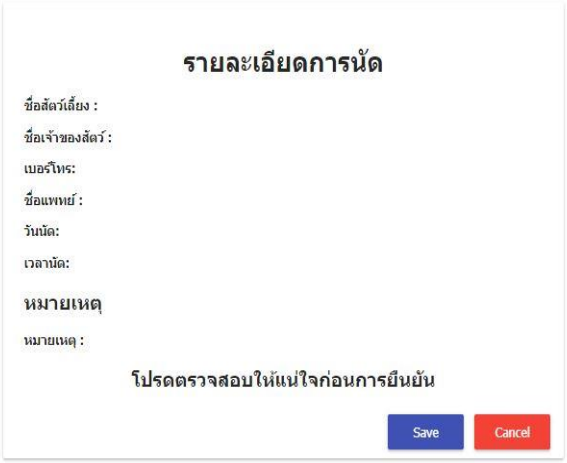
2. Business Use Case Diagram



3. UI Prototype



คลีนิคสัตว์เลี้ยง



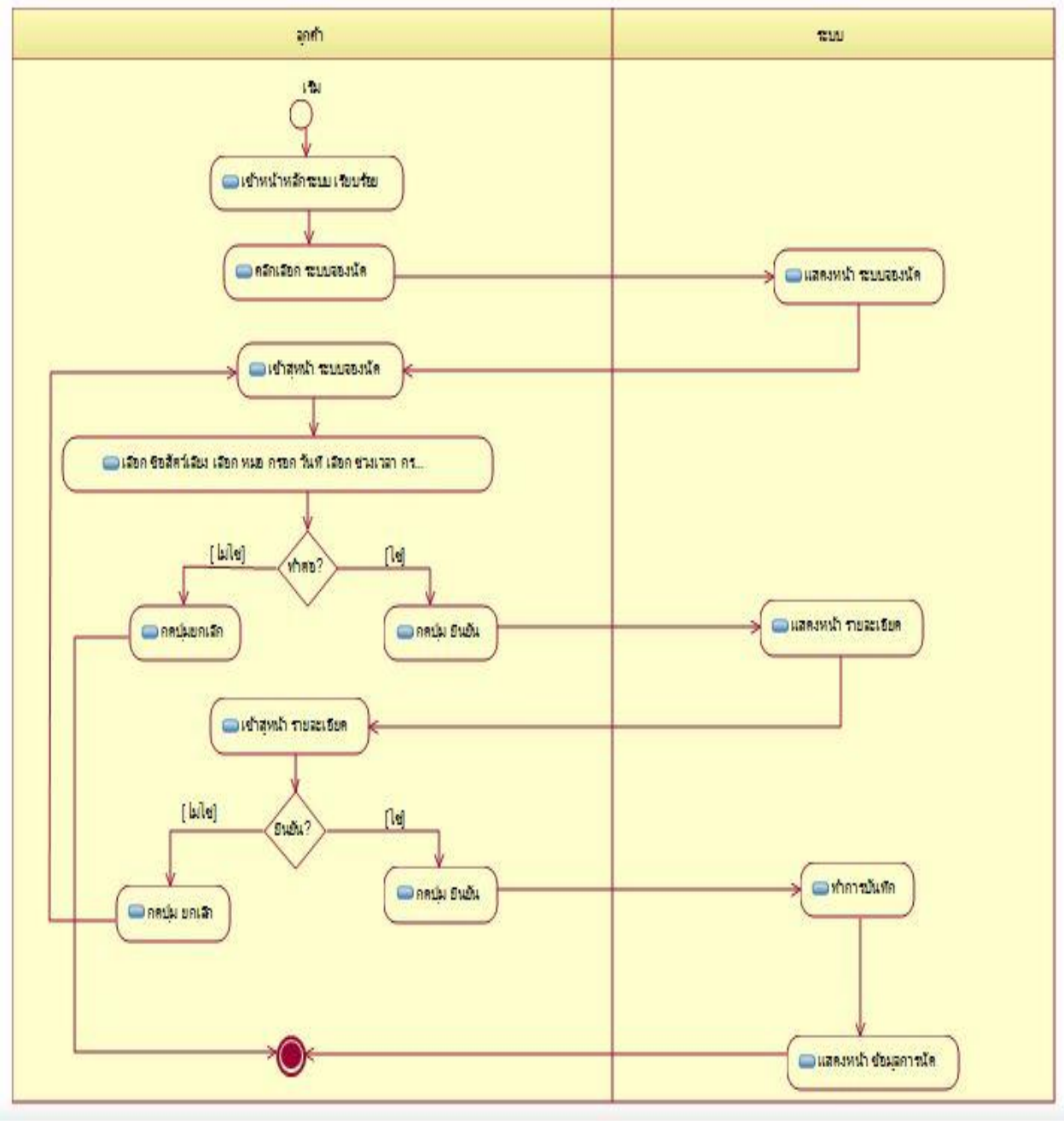
คลีนิคสัตว์เลี้ยง

ข้อมูลการนัด

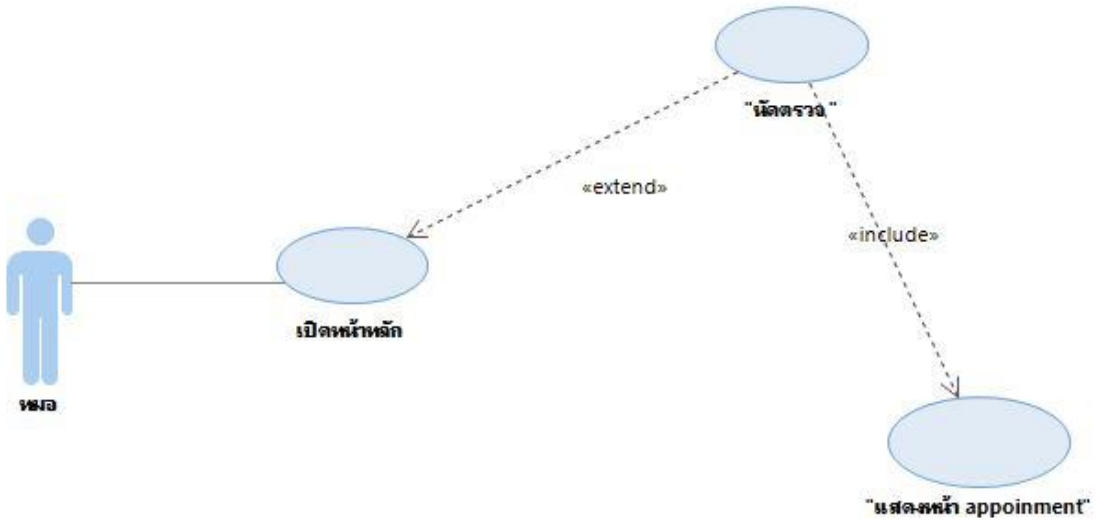
No.	รหัสสัตว์	ชื่อสัตว์เลี้ยง	อายุ	ชื่อเจ้าของสัตว์	นามสกุล	เบอร์โทร	รหัสแพทย์	ชื่อแพทย์	วันนัด	ช่วงเวลา	หมายเหตุ
1	2	Nobita	10	Zoro	Roronoa	0632418989	1	Tony Tony Chopper	2018-10-19	9.00-11.00	ฉีดวัคซีน
2	1	Doremon	90	Luffy	Monkey D.	0912832222	2	Black Jack	2018-10-19	13.00-16.00	ฉีดวัคซีน

หน้าหลัก

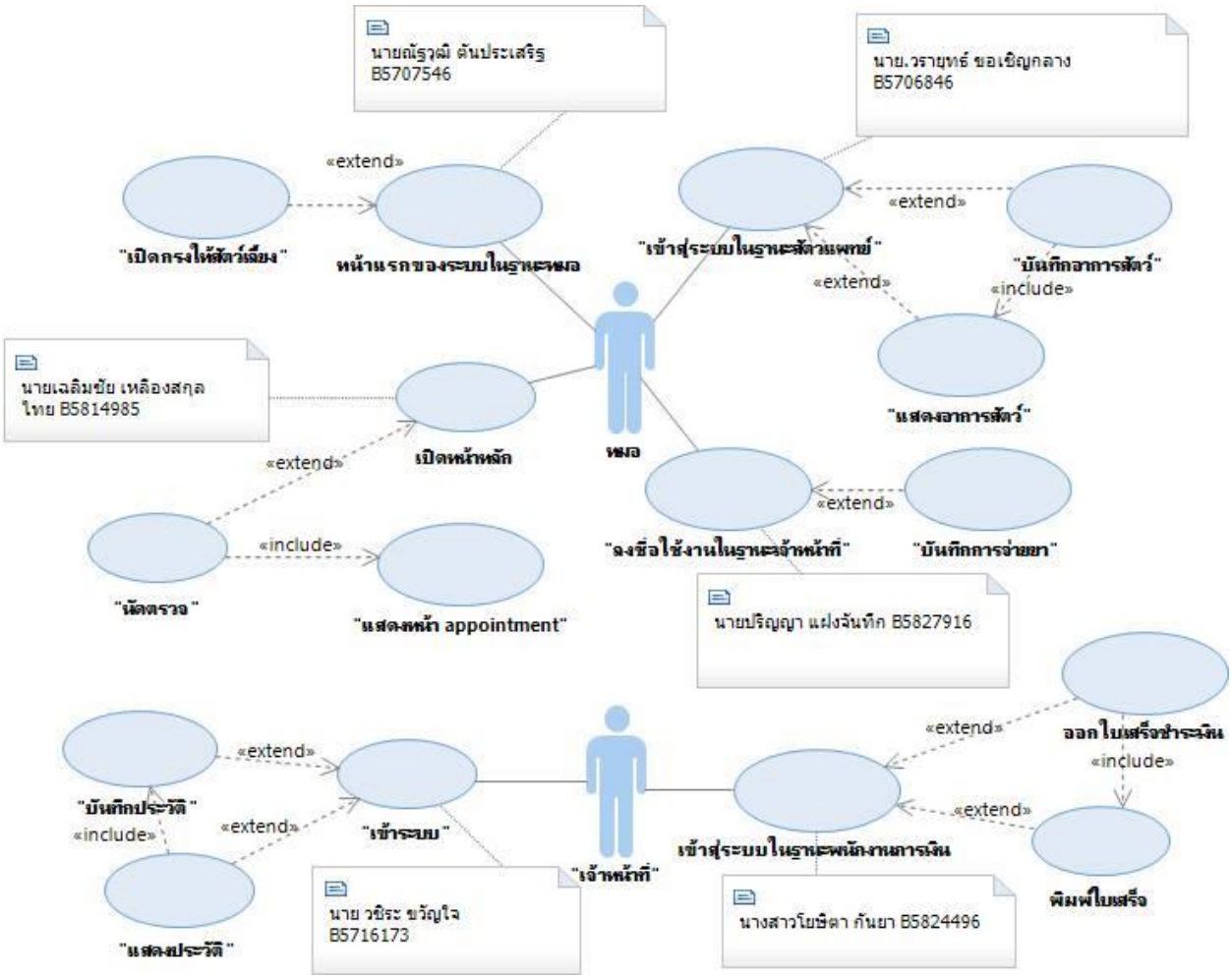
5. Activity Diagram



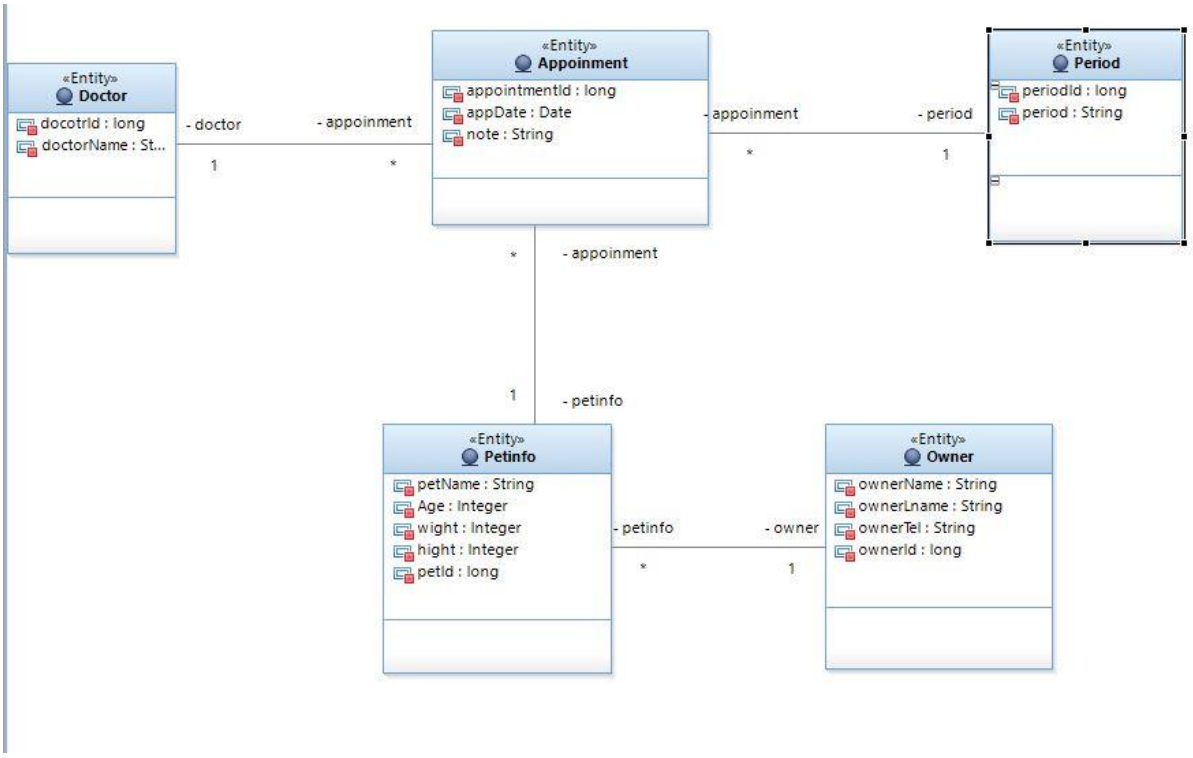
6. System Use Case Diagram



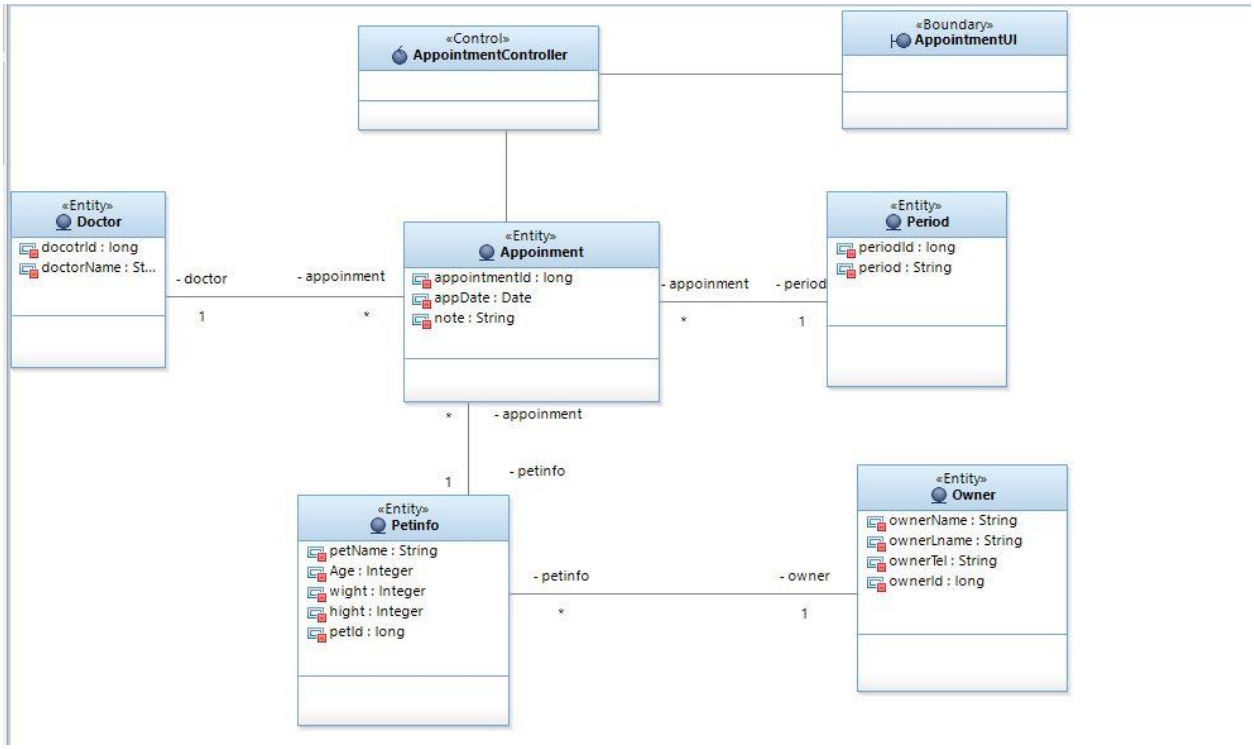
6.1 System Use Case Diagram รวม



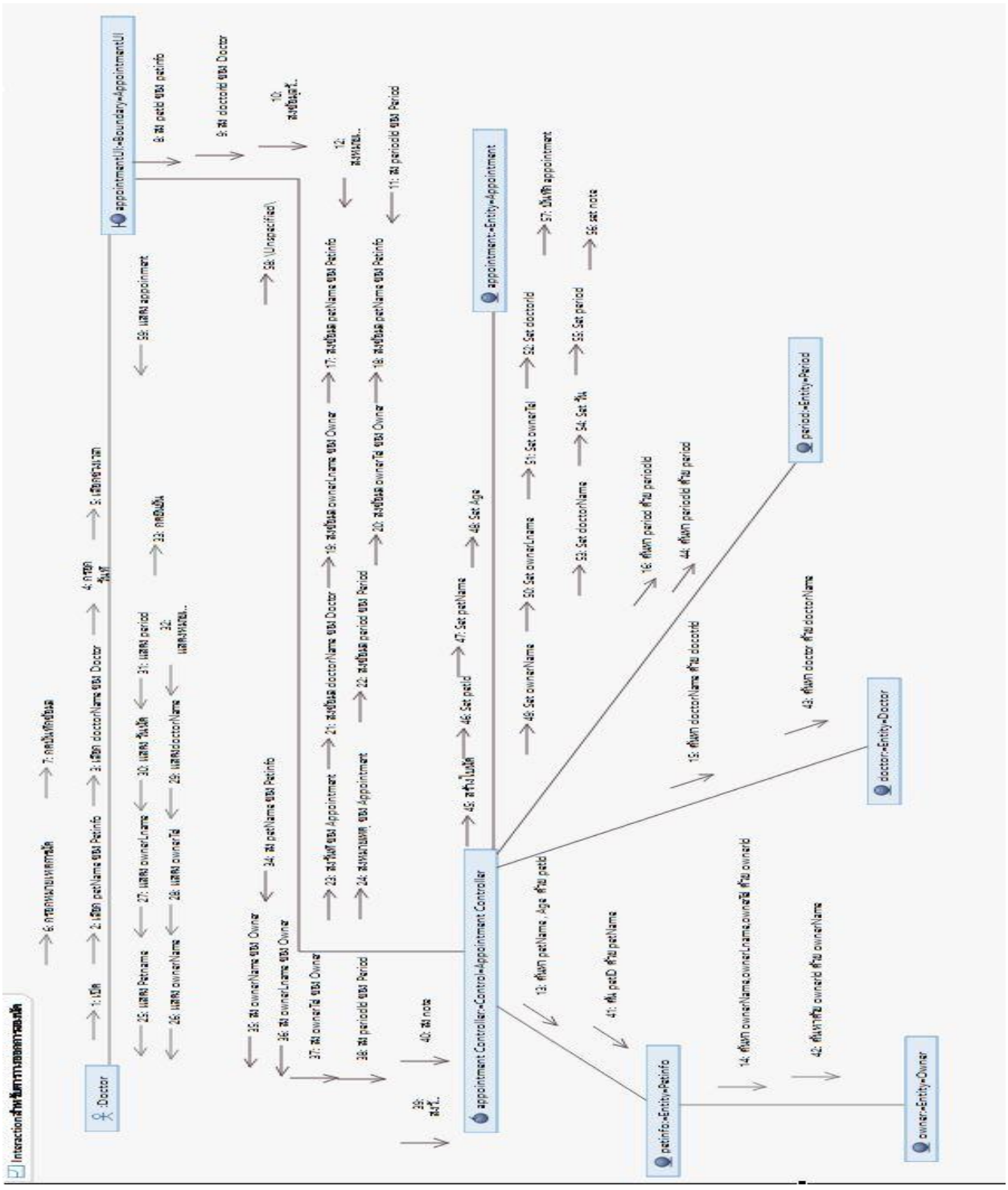
7. Class Diagram ระดับ Analysis



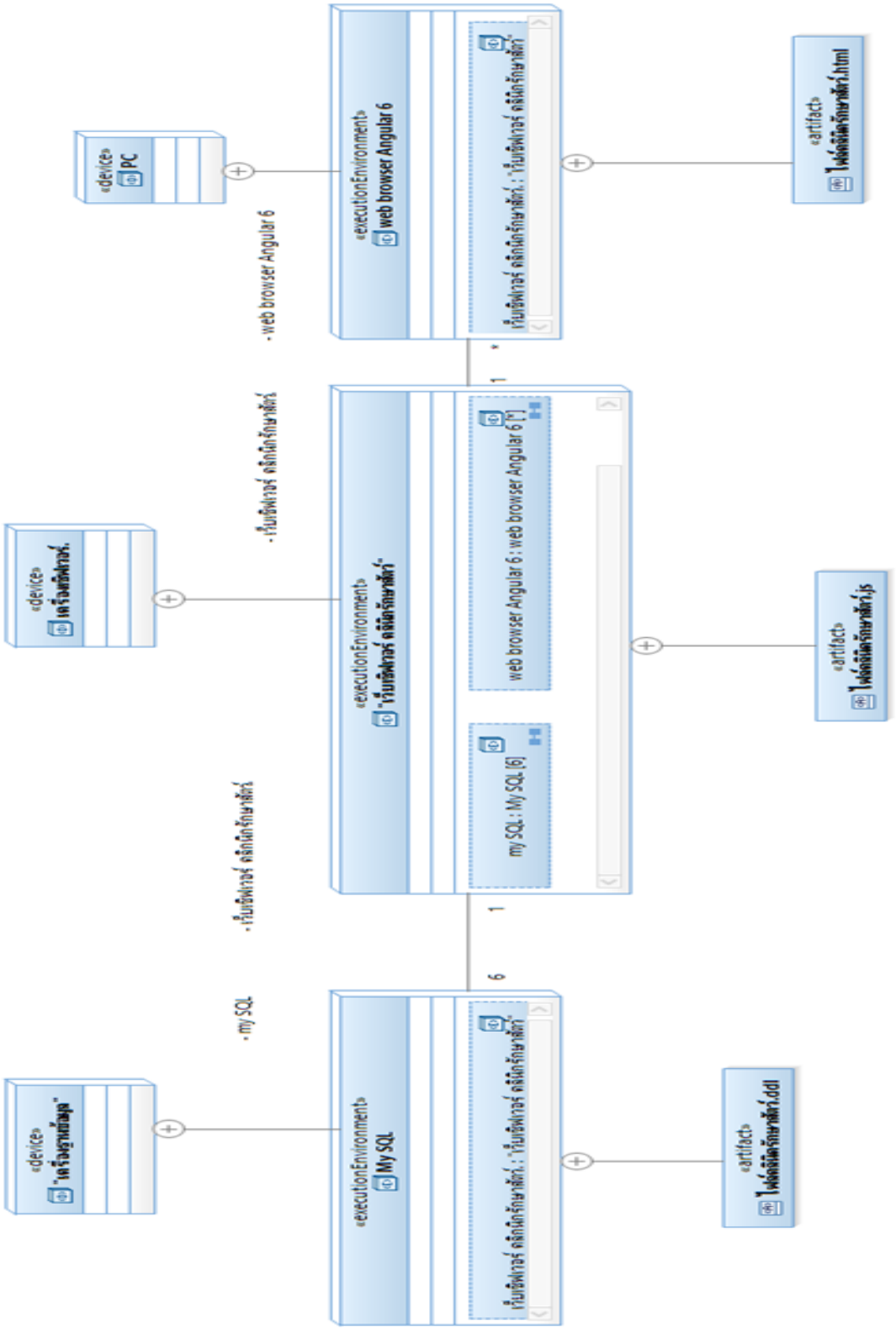
9. Class Diagram รวม



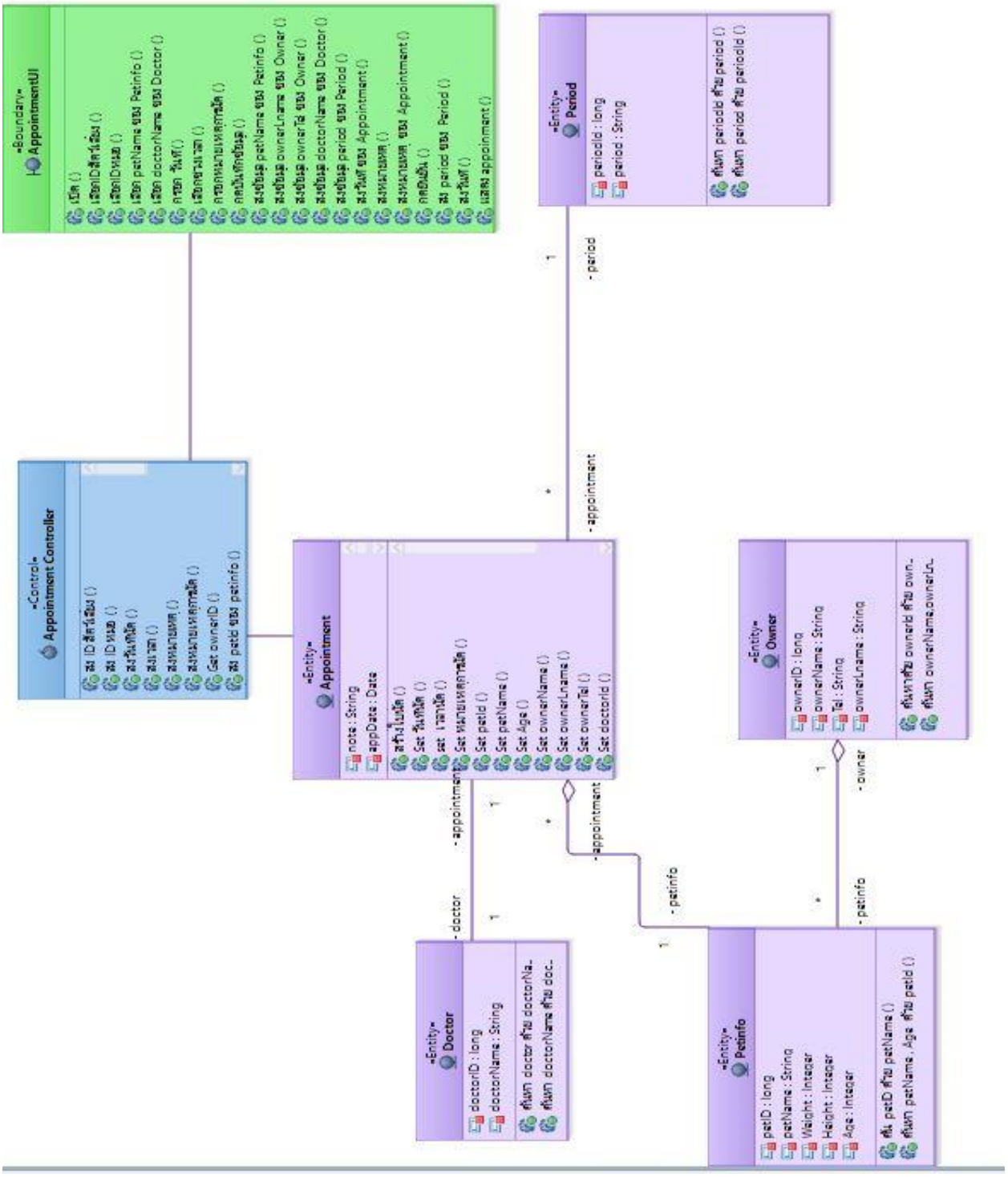
11. Communication Diagram



12. Deplyment Diagram

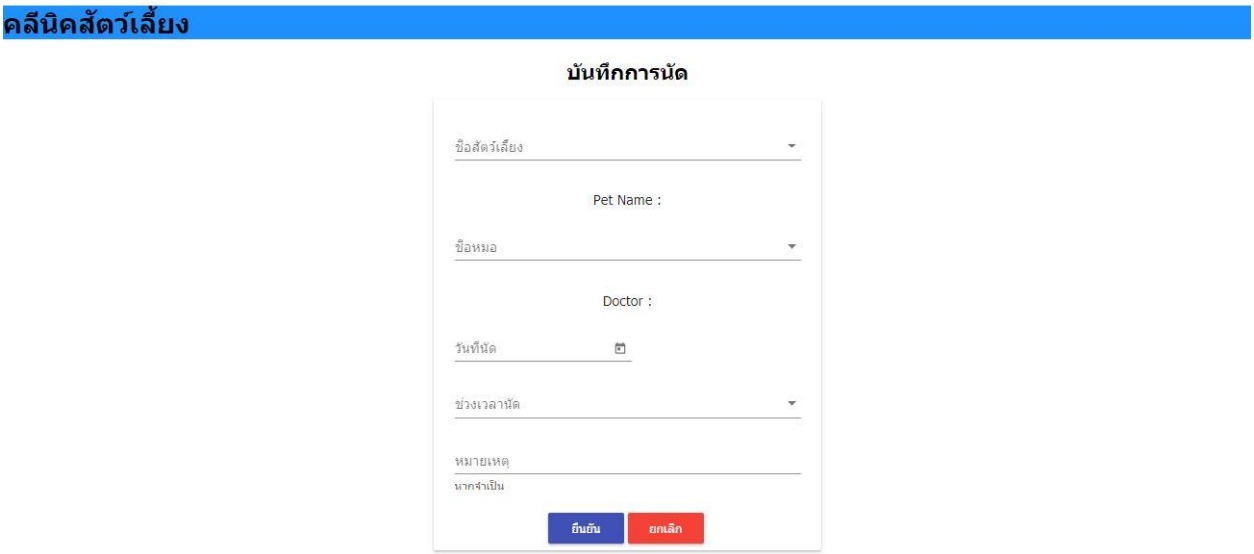


13.Class Diagram ระดับ Design\



14. UI สมบูรณ์


คลีนิคสัตว์
ระบบการจองนัด
ระบบการบันทึกข้อมูลสัตว์เลี้ยง
ระบบเปิดกรงสัตว์เลี้ยง
ระบบวินิจฉัยสัตว์เลี้ยง
ระบบจ่ายยา
ระบบจ่ายตังค์
ออกจากระบบ


คลีนิคสัตว์เลี้ยง
บันทึกการนัด
ชื่อสัตว์เลี้ยง
Pet Name :
ชื่อหมอ
Doctor :
วันที่นัด
ช่วงเวลานัด
หมายเหตุ
หากจำเป็น
ยืนยัน
ยกเลิก

คลีนิคสัตว์เลี้ยง

คลีนิคสัตว์เลี้ยง

ข้อมูลการนัด

No.	รหัสสัตว์	ชื่อสัตว์เลี้ยง	อายุ	ชื่อเจ้าของสัตว์	นามสกุล	เบอร์โทร	รหัสแพทย์	ชื่อแพทย์	วันนัด	ช่วงเวลา	หมายเหตุ
1	2	Nobita	10	Zoro	Roronoa	0632418989	1	Tony Tony Chopper	2018-10-19	9.00-11.00	ฉีดวัคซีน
2	1	Doremon	90	Luffy	Monkey D.	0912832222	2	Black Jack	2018-10-19	13.00-16.00	ฉีดวัคซีน

หน้าหลัก

15. Source Code

Client

Source Code : appointment.component.html

```
<div class="div-top" style="background-color:DodgerBlue;">

 <h1>คลีนิคสัตว์เลี้ยง</h1>

</div>

<div class="is-centered">

 <h2>บันทึกการนัด</h2>

 <mat-card class="main-card center">

        <mat-form-field class="example-full-width">

          <mat-select placeholder="ชื่อสัตว์เลี้ยง"

          [(ngModel)]="select.petnameSelect"

          [(ngModel)]="data.petN"

          [(ngModel)]="petName"

          (blur)="getPetinfoDetail()"

          name = "petN"#petN>

          <mat-option *ngFor="let petname of Petname" [value]="petname.petId">

           {{petname.petName}}

          </mat-option>
```

```
    </mat-select>
  </mat-form-field>
  <p>Pet Name : {{select.petnameSelect}}</p>
  <mat-form-field class="example-full-width">
    <mat-select placeholder="ชื่อหมอ"
    [(ngModel)]="select.doctornameSelect"
    [(ngModel)]="data.docN"
    [(ngModel)]="doctorName"
    (blur)="getDoctorDetail()"
    name = "docN"#docN>
    <mat-option *ngFor="let Doctor of Doctors" [value]="Doctor.doctorId">
      {{Doctor.doctorName}}
    </mat-option>
  </mat-select>
</mat-form-field>
<p>Doctor : {{select.doctornameSelect}}</p>
<div style="text-align: left">
  <mat-form-field color="accent">

    <mat-label>วันที่นัด</mat-label>
```

```
  <input matInput
  [matDatepicker]="picker2"
  placeholder="วันที่นัด"
  [(ngModel)]="appDate"
  [(ngModel)]="data.date">
  <mat-datepicker-toggle matSuffix [for]="picker2"></mat-datepicker-toggle>
  <mat-datepicker #picker2 color="primary"></mat-datepicker>
</mat-form-field>
<!-- <p>Date : {{appDate}}</p> -->
<mat-form-field class="example-full-width">
  <mat-select placeholder="ช่วงเวลานัด"
  [(ngModel)]="select.periodSelect"
  [(ngModel)]="data.period"
  [(ngModel)]="period"
  (blur)="getPeriodDetail()"
  name = "per" #per >
  <mat-option *ngFor="let Period of Periods" [value]="Period.periodId">
    {{Period.period}}
  </mat-option>
</mat-select>
```

```
    </mat-form-field>
    <!-- <p>Period : {{select.periodSelect}}</p> -->
  </div>
  <form class="example-form">
    <mat-form-field class="example-full-width">
      <input matInput
      placeholder="หมายเหตุ"
      [(ngModel)]="note"
      [(ngModel)]="data.note"
      name="note">
      <mat-hint align="start"><strong>หากจำเป็น</strong> </mat-hint>
    </mat-form-field>
    <p>{{note}}</p>
  </form>
 <mat-card-actions class="is-centered">
  <button mat-raised-button (click)="SubmittedData()"routerLink="/appointmentview"
color="primary" type="submit">ยืนยัน</button>
  <button mat-raised-button (click)="cancel()"routerLink="/menu" color="warn" >ยกเลิก</button>
 </mat-card-actions>
</mat-card>
```

```
</div>
```

Source Code : appointment.component.ts

```
import { Component, OnInit } from '@angular/core';

import { AppointmentService } from '../appointment.service';

import { HttpClient} from '@angular/common/http';

import { Router } from "@angular/router";

@Component({

  selector: 'app-appointment',

  templateUrl: './appointment.component.html',

  styleUrls: ['./appointment.component.css']

})

export class AppointmentComponent implements OnInit {



  select: any = {

    petnameSelect : '',

    doctornameSelect : '',

    periodSelect: '',
```

```
  };

  doc: any = {};

  pet: any = {};

  p: any = {};

Petname : Array<any>

Doctors : Array<any>

Periods : Array<any>

Owners : Array<any>

appDate : any;

note: any;



data: any  = {

  // appId:'',

  petN:'',

  ownerN:'',

  ownerLn:'',

  tl:'',

  docN:'',
```

```
  date:'',
  per:'',
  note:'',
}
  constructor(private appointmentService: AppointmentService,private httpClient:
HttpClient,private router:Router) { }

  ngOnInit() {
    this.appointmentService.getPetinfo().subscribe(data => {
      this.Petname = data;
      console.log(this.Petname);
    });
    this.appointmentService.getDoctor().subscribe(data => {
      this.Doctors = data;
      console.log(this.Doctors);
    });
    this.appointmentService.getPeriod().subscribe(data => {
      this.Periods = data;
```

```
    console.log(this.Periods);

   });

 }

 getDoctorDetail(){

  this.appointmentService.getDoctorDetail(this.select.doctornameSelect).subscribe(data
=> {

    this.doc = data;

    console.log(this.doc);

   });

 }

 getPetinfoDetail(){

   this.appointmentService.getPetinfoDetail(this.select.petnameSelect).subscribe(data
=> {

    this.pet = data;

    console.log(this.pet);

   });

 }

 getPeriodDetail(){

   this.appointmentService.getPeriodDetail(this.select.periodSelect).subscribe(data => {
```

```
    this.p = data;
    console.log(this.p);
  });
 }
 SubmittedData(){
  console.log(this.data)
  const data  = this.data
  if (this.select.petnameSelect === '' || this.select.doctornameSelect === ''  ||
this.appDate === ''
  || this.select.periodSelect === '' || this.note === '') {
    alert('กรุณากรอกข้อมูลให้ครบถ้วน');
  } else {
    this.router.navigate(['appointmentview',{
      // appId:  data.appId,
      petN:   data.petN  = this.pet.petName,
      ownerN:  data.ownerN = this.pet.petowner.ownerName,
      ownerLn:  data.ownerLn = this.pet.petowner.ownerLname,
      tl:       data.tl = this.pet.petowner.ownerTel,
```

```
      docN:     data.docN = this.doc.doctorName,

      date:    data.date ,

      per:      data.per = this.p.period,

      note:     data.note ,

      }])

   }

}

  save() {

   if (this.select.petnameSelect === '' || this.select.doctornameSelect === ''  ||
this.appDate === ''

   || this.select.periodSelect === '' || this.note=== '') {

   } else {

     this.httpClient.post('http://localhost:8080/appointment/' + this.select.petnameSelect +
'/'

     + this.select.doctornameSelect + '/' + this.appDate + '/' + this.select.periodSelect +
'/' + this.note ,this.select)

     .subscribe(

        data => {

           console.log('PUT Request is successful', data);
```

```
        },
        error => {
            console.log('Error', error);
        }
    );
  }
 }
}
```

Source Code : appointment.component.css

```
.example-form {
    min-width: 150px;
    max-width: 500px;
    width: 100%;
 }
```

```
.example-full-width {
  width: 100%;
}
.button-row button,
.button-row a {
  margin-right: 8px;
}
.mat-radio-button ~ .mat-radio-button {
  padding-right: 16px;
}
.main-card {
  max-width: 400px;
  justify-content: center;
}
.mat-card-content {
  justify-content: center;
  text-align: center;
}
```

```
.center{
  width: 75%;
  margin: 10px auto;
}
.main-form {
  min-width: 150px;
  max-width: 500px;
  width: 100%;
}
.main-full-width {
  width: 100%;
}
.mat-select-panel {
  display: flex;
}
.is-centered{
  text-align: center;
}
```

Source Code : appointment-show.component.html

```
<router-outlet></router-outlet>

<div class="div-top" style="background-color:DodgerBlue;">

  <h1>คลีนิคสัตว์เลี้ยง</h1>

</div>



<!-- <div style="text-align: center"> -->

    <mat-card>

    <h2 class="title-name">ข้อมูลการนัด</h2>

    <table mat-table [dataSource]="appointments" class="mat-elevation-z8">

      <ng-container matColumnDef="no">

        <th mat-header-cell *matHeaderCellDef> No. </th>

        <td mat-cell *matCellDef ="let element"> {{element.appointmentId}} </td>

      </ng-container>



      <ng-container matColumnDef="PetId">

        <th mat-header-cell *matHeaderCellDef> รหัสสัตว์ </th>
```

```
      <td mat-cell *matCellDef ="let element"> {{element.petInfo.petId}} </td>

    </ng-container>



    <ng-container matColumnDef ="PetName">

      <th mat-header-cell *matHeaderCellDef> ชื่อสัตว์เลี้ยง </th>

      <td mat-cell *matCellDef ="let element"> {{element.petInfo.petName}} </td>

    </ng-container>



    <ng-container matColumnDef ="Age">

      <th mat-header-cell *matHeaderCellDef> อายุ </th>

      <td mat-cell *matCellDef ="let element"> {{element.petInfo.age}} </td>

    </ng-container>



    <ng-container matColumnDef ="OwnerName">

      <th mat-header-cell *matHeaderCellDef> ชื่อเจ้าของสัตว์ </th>

      <td mat-cell *matCellDef ="let element"> {{element.petInfo.petowner.ownerName}}
</td>

    </ng-container>
```

```
      <ng-container matColumnDef ="OwnerLname">
        <th mat-header-cell *matHeaderCellDef> นามสกุล </th>
        <td mat-cell *matCellDef ="let element">
{{element.petInfo.petowner.ownerLname}} </td>
      </ng-container>

      <ng-container matColumnDef ="Tel">
        <th mat-header-cell *matHeaderCellDef> เบอร์โทร </th>
        <td mat-cell *matCellDef ="let element"> {{element.petInfo.petowner.ownerTel}}
</td>
      </ng-container>

      <ng-container matColumnDef ="DoctorId">
        <th mat-header-cell *matHeaderCellDef> รหัสแพทย์</th>
        <td mat-cell *matCellDef ="let element"> {{element.doctor.doctorId}} </td>
      </ng-container>

      <ng-container matColumnDef ="DoctorName">
```

```
  <th mat-header-cell *matHeaderCellDef> ชื่อแพทย์ </th>
  <td mat-cell *matCellDef ="let element"> {{element.doctor.doctorName}} </td>
</ng-container>

<ng-container matColumnDef ="appDate">
  <th mat-header-cell *matHeaderCellDef> วันนัด </th>
  <td mat-cell *matCellDef ="let element"> {{element.appDate}} </td>
</ng-container>

<ng-container matColumnDef ="Period">
  <th mat-header-cell *matHeaderCellDef> ช่วงเวลา </th>
  <td mat-cell *matCellDef ="let element"> {{element.period.period}} </td>
</ng-container>

<ng-container matColumnDef ="Note">
  <th mat-header-cell *matHeaderCellDef> หมายเหตุ </th>
  <td mat-cell *matCellDef ="let element"> {{element.note}} </td>
</ng-container>
```

```
        <tr mat-header-row *matHeaderRowDef ="displayedColumns"></tr>
       <tr mat-row *matRowDef ="let row; columns: displayedColumns;"></tr>
     </table>
     <mat-card-actions style ="text-align:right">
        <a mat-raised-button routerLink ="/menu" color="primary" >หน้าหลัก</a>
      </mat-card-actions>
  </mat-card>
```

Source Code : appointment-show.component.ts

```
import { Component, OnInit } from '@angular/core';
import { AppointmentService } from '../appointment.service';

import { Location } from '@angular/common';
import { HttpClient} from '@angular/common/http';
import { Router } from "@angular/router";

@Component({
```

```
  selector: 'app-appointment-show',

  templateUrl: './appointment-show.component.html',

  styleUrls: ['./appointment-show.component.css']

})

export class AppointmentShowComponent implements OnInit {



petinfos : Array<any>;

owners : Array<any>;

doctors : Array<any>;

periods : Array<any>;

appointments : Array<any>



displayedColumns: string[] = ['no', 'PetId',

'PetName','Age','OwnerName','OwnerLname','Tel','DoctorId','DoctorName','appDate','Pe

riod','Note'];

constructor(private appointmentService: AppointmentService,private router:

Router,private httpClient: HttpClient,private location: Location) { }



  ngOnInit() {
```

```
this.appointmentService.getPetinfo().subscribe(data => {
  this.petinfos = data;
  console.log(this.petinfos);
});
this.appointmentService.getOwner().subscribe(data => {
  this.owners = data;
  console.log(this.owners);
});
this.appointmentService.getDoctor().subscribe(data => {
  this.doctors = data;
  console.log(this.doctors);
});
this.appointmentService.getPeriod().subscribe(data => {
  this.periods = data;
  console.log(this.periods);
});
this.appointmentService.getAppointment().subscribe(data => {
  this.appointments = data;
```

```
      console.log(this.appointments);

    });

  }

}
```

Source Code : appointment-show.component.css

```
table {

    width: 100%;

  }

.button-row button,

.button-row a {

  margin-right: 8px;

}
```

Source Code : appointment-view.component.html

```
<router-outlet></router-outlet>



<div class="div-top" style="background-color:DodgerBlue;">
```

```
 <h1>คลีนิคสัตว์เลี้ยง</h1>



</div>

<mat-card class="example-card">

   <mat-card-content style="text-align:left" >

     <p hidden></p>

     <h1 style="text-align:center">รายละเอียดการนัด</h1>

     <p><b>ชื่อสัตว์เลี้ยง :  {{ data.petN}} </b> </p>

     <p><b>ชื่อเจ้าของสัตว์ : {{data.ownerN}}   {{data.ownerLn}}</b></p>

     <p><b>เบอร์โทร:  {{data.tl}}</b> </p>

     <p><b>ชื่อแพทย์ :   {{data.docN}} </b> </p>

     <p><b>วันนัด:  {{data.date}}</b></p>

     <p><b>เวลานัด:  {{data.per}} </b></p>

     <h2>หมายเหตุ</h2>

     <p><b>หมายเหตุ :{{data.note}}</b></p>



     <h2 style="text-align:center">โปรดตรวจสอบให้แน่ใจก่อนการยืนยัน</h2>

   </mat-card-content>
```

```
    <mat-card-actions style="text-align:right">
      <div class="example-button-row">
     <button mat-raised-button (click)="save()" routerLink="/appointmentshow"
color="primary" type="submit">Save</button>
     <button mat-raised-button (click)="cancel()" color="warn"
routerLink="/appointment">Cancel</button>
     </div>
    </mat-card-actions>
  </mat-card>
```

Source Code : appointment-view.component.ts

```
import { Component, OnInit } from '@angular/core';
import { HttpClient} from '@angular/common/http';
import { ActivatedRoute } from "@angular/router";
import { Router } from "@angular/router";
import { AppointmentService } from '../appointment.service';
```

```
@Component({
  selector: 'app-appointment-view',
  templateUrl: './appointment-view.component.html',
  styleUrls: ['./appointment-view.component.css']
})
export class AppointmentViewComponent implements OnInit {
  data:any={}

  constructor(private route:ActivatedRoute,private appointmentService:
AppointmentService, private httpClient: HttpClient, private router:Router) { }

  ngOnInit() {
    this.route.params.subscribe(prams=>{
      this.data = prams
      console.log(prams)
    });
  }
```

```
  save() {
    this.httpClient.post('http://localhost:8080/appointment/' + this.data.petN + '/' +
this.data.ownerN + '/' + this. data.ownerLn + '/' + this.data.tl + '/' + this.data.docN + '/' +
this.data.date + '/' + this.data.per + '/' + this.data.note,this.data)
    .subscribe(
      data =>   {console.log('PUT Request is successful', data);},
      error =>  {console.log('Error', error);}
      );
    }
```

Source Code : appointment-view.component.css

```
.example-card {
    max-width: 600px;
    margin-left: auto;
    margin-right: auto;
    font-family: 'Varela Round', sans-serif;
}
```

```
.example-button-row button,

.example-button-row a {

  margin-right: 8px;

}
```

Source Code : menu.component.html

```
<router-outlet></router-outlet>

<html>

  <head>

  </head>

  <br><br>

  <h2 style="text-align:center;">คลีนิคสัตว์</h2>

  <div class ="w3-display-middle">

    <mat-card  class ="card2 ">

      <button mat-stroked-button color="primary" class="w3-display-
middle"[routerLink]="['/checkup']" >ระบบการจองนัด</button><br><br>

      <button mat-stroked-button color="primary" class="w3-display-middle " >ระบบการ
บันทึกข้อมูลสัตว์เลี้ยง</button><br><br>
```

```
    <button mat-stroked-button color="primary" class="w3-display-middle">ระบบเปิดกรงสัตว์เลี้ยง</button><br><br>

    <button mat-stroked-button color="primary" class="w3-display-middle">ระบบวินิจฉัยสัตว์เลี้ยง</button><br><br>

    <button mat-stroked-button color="primary" class="w3-display-middle">ระบบจ่ายยา</button><br><br>

    <button mat-stroked-button color="primary" class="w3-display-middle">ระบบจ่ายตังค์</button><br><br>

    <button mat-stroked-button color="warn" class="w3-display-middle"[routerLink]="['/login']">ออกจากระบบ</button>

   </mat-card>

  </div>

</html>
```

Source Code : menu.component.ts

```
import { Component, OnInit } from '@angular/core';



@Component({
  selector: 'app-menu',
  templateUrl: './menu.component.html',
  styleUrls: ['./menu.component.css']
})
export class MenuComponent implements OnInit {


  constructor() { }


  ngOnInit() {
  }


}
```

Source Code : appointment.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs/Observable';


@Injectable({
  providedIn: 'root'
})
export class AppointmentService {


  public API = '//localhost:8080';


  constructor(private http: HttpClient) { }


getAppointment(): Observable<any> {
    return this.http.get(this.API + '/appointment');
  }
```

```
getDoctor(): Observable<any> {
    return this.http.get(this.API + '/doctor');
}
getPeriod(): Observable<any> {
  return this.http.get(this.API + '/period');
}
getPetinfo(): Observable<any> {
  return this.http.get(this.API + '/petinfo');
}
getOwner(): Observable<any> {
  return this.http.get(this.API + '/owner');
}

getDoctorDetail(docN: number) : Observable<any> {
  return this.http.get(this.API + '/Doctors/getById/' + docN);
}
getPetinfoDetail(petN: number) : Observable<any>{
```

```
  return this.http.get(this.API + '/Petnames/getById/' + petN);

}

getOwnerDetail(ownerN , ownerLn , tl : number) : Observable<any>{

  return this.http.get(this.API + '/Owners/getById/' + ownerN + ownerLn + tl );

}

getPeriodDetail(period : number) : Observable<any>{

  return this.http.get(this.API + '/Periods/getById/' + period );

}



}
```

Server

Entity

Source Code : Appointment.java

```
package com.sut.sa.group.entity;

import java.util.Date;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.FetchType;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.ManyToOne;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;

import javax.persistence.Temporal;

import javax.persistence.TemporalType;



import lombok.Data;
```

```
import lombok.EqualsAndHashCode;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

import lombok.ToString;



@Data

@Entity

@Setter @Getter

@NoArgsConstructor

@ToString

@EqualsAndHashCode

@Table(name = "appointment")

public class Appointment{

    @Id

    @SequenceGenerator(name="appointment_seq",sequenceName="appointment_seq")

    @GeneratedValue(strategy=GenerationType.SEQUENCE,

generator="appointment_seq")
```

```
    @Column(name="APPOINTMENT_ID")

    private Long appointmentId;

    @Temporal(TemporalType.DATE)

    private Date appDate;

    private String note;



    @ManyToOne(fetch = FetchType.EAGER, targetEntity = PetInfo.class)

    @JoinColumn(name = "PETINFO_ID", insertable = true)

    private PetInfo petInfo;



    @ManyToOne(fetch = FetchType.EAGER, targetEntity = Doctor.class)

    @JoinColumn(name = "DOCTOR_ID", insertable = true)

    private Doctor doctor;



    @ManyToOne(fetch = FetchType.EAGER, targetEntity = Period.class)

    @JoinColumn(name = "PERIOD_ID", insertable = true)

    private Period period;

}
```

Source Code : Doctor.java

```
package com.sut.sa.group.entity;


import javax.persistence.*;


import lombok.*;


@Data
@Entity
@Getter @Setter
@NoArgsConstructor
@ToString
@EqualsAndHashCode
@Table(name = "doctor")


public class Doctor{
    @Id
```

```
    @SequenceGenerator(name="doctor_seq",sequenceName="doctor_seq")
    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="doctor_seq")
    @Column(name="DOCTOR_ID")
    private Long doctorId;
    private String doctorName;


}
```

Source Code : Owner.java

```
package com.sut.sa.group.entity;
import javax.persistence.*;
import lombok.*;
@Data
@Entity
@Getter @Setter
@NoArgsConstructor
@ToString
@EqualsAndHashCode
```

```
@Table(name = "owner")



public class Owner{

    @Id

    @SequenceGenerator(name="owner_seq",sequenceName="owner_seq")

    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="owner_seq")

    @Column(name="OWNER_ID")

    private Long ownerId;

    private String ownerName;

    private String ownerLname;

    private String ownerTel;



}
```

Source Code : Period.java

```
package com.sut.sa.group.entity;
```

```
import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.SequenceGenerator;

import javax.persistence.Table;


import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.Getter;

import lombok.NoArgsConstructor;

import lombok.Setter;

import lombok.ToString;


@Data

 @Entity
```

```
@Getter @Setter

@NoArgsConstructor

@ToString

@EqualsAndHashCode

@Table(name = "period")

public class Period {

  @Id

  @SequenceGenerator(name="period_seq",sequenceName="period_seq")

  @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="period_seq")

  @Column(name="PERIOD_ID")

  private Long periodId;

  private String period;

}
```

Source Code : PetInfo.java

```
package com.sut.sa.group.entity;
```

```
import javax.persistence.*;

import lombok.*;

@Data

@Entity

@Getter @Setter

@NoArgsConstructor

@ToString

@EqualsAndHashCode

@Table(name = "petinfo")



public class PetInfo{

    @Id

    @SequenceGenerator(name="petinfo_seq",sequenceName="petinfo_seq")

    @GeneratedValue(strategy=GenerationType.SEQUENCE, generator="petinfo_seq")

    @Column(name="PETINFO_ID")

    private Long petId;

    private String petName;

    private Double wiegth;
```

```
    private Double hight;

    private int age;



    @ManyToOne(fetch = FetchType.EAGER, targetEntity = Owner.class)

    @JoinColumn(name = "OWNER_ID", insertable = true)

    private Owner petowner;

}
```

Repository

Source Code : AppointmentRepository.java

```
package com.sut.sa.group.repository;

import com.sut.sa.group.entity.Appointment;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.rest.core.annotation.RepositoryRestResource;

@RepositoryRestResource

public interface AppointmentRepository extends JpaRepository<Appointment,Long>{

}
```

Source Code : DoctorRepository.java

```
package com.sut.sa.group.repository;

import com.sut.sa.group.entity.Doctor;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.rest.core.annotation.RepositoryRestResource;

@RepositoryRestResource

public interface DoctorRepository extends JpaRepository<Doctor,Long>{

    Doctor findByDoctorId(long doctorId);

}
```

Source Code : PeriodRepository.java

```
package com.sut.sa.group.repository;
```

```
import com.sut.sa.group.entity.Period;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.rest.core.annotation.RepositoryRestResource;

@RepositoryRestResource

public interface PeriodRepository extends JpaRepository<Period, Long> {

    Period findByPeriodId(long periodId);

}
```

Source Code : PetInfoRepository.java

```
package com.sut.sa.group.repository;

import com.sut.sa.group.entity.PetInfo;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.rest.core.annotation.RepositoryRestResource;

@RepositoryRestResource

public interface PetInfoRepository extends JpaRepository<PetInfo,Long>{

    PetInfo findByPetId(long petId);

}
```

Source Code : OwnerRepository.java

```
package com.sut.sa.group.repository;
```

```
import com.sut.sa.group.entity.Owner;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.data.rest.core.annotation.RepositoryRestResource;

@RepositoryRestResource

public interface OwnerRepository extends JpaRepository<Owner,Long>{

    Owner findById(long ownerId);

}
```

Controller

Source Code : AppointmentController.java

```
package com.sut.sa.group.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.*;

import java.util.Date;

import java.util.Collection;

import java.util.stream.Collectors;

import com.sut.sa.group.entity.Appointment;

import com.sut.sa.group.entity.Doctor;

import com.sut.sa.group.entity.Period;

import com.sut.sa.group.entity.PetInfo;

import com.sut.sa.group.repository.*;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

public class AppointmentController {

    @Autowired private AppointmentRepository appointmentRepository;

    @Autowired private DoctorRepository doctorRepository;

    @Autowired private PetInfoRepository petInfoRepository;
```

```
@Autowired private PeriodRepository periodRepository;



public AppointmentController(AppointmentRepository appointmentRepository){

    this.appointmentRepository = appointmentRepository;

}



@GetMapping("/appointment")

public Collection<Appointment> appointment() {

    return appointmentRepository.findAll().stream().collect(Collectors.toList());

}



@PostMapping("/appointment/{petId}/{doctorId}/{appDate}/{periodId}/{note}")

public Appointment newAppointment(@RequestBody Appointment newAppointment,

                    @PathVariable long petId,

                    @PathVariable long doctorId,

                    @PathVariable long periodId,

                    @PathVariable Date appDate,

                    @PathVariable String note
```

```
    ) {


        Doctor doctorapp = doctorRepository.findByDoctorId(doctorId);

        PetInfo petapp = petInfoRepository.findByPetId(petId);

        Period periodapp = periodRepository.findByPeriodId(periodId);




        newAppointment.setAppDate(appDate);

        newAppointment.setNote(note);

        newAppointment.setDoctor(doctorapp);

        newAppointment.setPetInfo(petapp);

        newAppointment.setPeriod(periodapp);

        return appointmentRepository.save(newAppointment);

    }

}
```

Source Code : DoctorController.java

```
package com.sut.sa.group.controller;
```

```
import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.*;

import java.util.Collection;

import java.util.stream.Collectors;

import com.sut.sa.group.entity.Doctor;

import com.sut.sa.group.repository.DoctorRepository;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

public class DoctorController {

    @Autowired private DoctorRepository doctorRepository;

    public DoctorController(DoctorRepository doctorRepository){

        this.doctorRepository = doctorRepository;

    }

    @GetMapping("/doctor")

    public Collection<Doctor> doctror() {

        return doctorRepository.findAll().stream().collect(Collectors.toList());

    }

    @GetMapping("/Doctors/getById/{doctorId}")
```

```
    public Doctor Doctor2(@PathVariable Long doctorId) {

        Doctor  AreaData = doctorRepository.findByDoctorId(doctorId);

        return AreaData;

    }

}
```

Source Code : OwnerController.java

```
package com.sut.sa.group.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.*;

import java.util.Collection;

import java.util.stream.Collectors;

import com.sut.sa.group.entity.Owner;

import com.sut.sa.group.repository.OwnerRepository;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

public class OwnerController {

    @Autowired private OwnerRepository ownerRepository;

    public OwnerController(OwnerRepository ownerRepository){
```

```
        this.ownerRepository = ownerRepository;
    }
    @GetMapping("/owner")
    public Collection<Owner> owner() {
        return ownerRepository.findAll().stream().collect(Collectors.toList());
    }
}
```

Source Code : PeriodController.java

```
package com.sut.sa.group.controller;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import java.util.Collection;
import java.util.stream.Collectors;
import com.sut.sa.group.entity.Period;
import com.sut.sa.group.repository.PeriodRepository;
@CrossOrigin(origins = "http://localhost:4200")
@RestController
```

```
public class PeriodController {

    @Autowired private PeriodRepository periodRepository;

    public PeriodController(PeriodRepository periodRepository){

        this.periodRepository = periodRepository;

    }

    @GetMapping("/period")

    public Collection<Period> period() {

        return periodRepository.findAll().stream().collect(Collectors.toList());

    }

    @GetMapping("/Periods/getById/{periodId}")

    public Period Period2(@PathVariable Long periodId) {

        Period  AreaData = periodRepository.findByPeriodId(periodId);

        return AreaData;

    }

}
```

Source Code : PetInfoController.java

```
package com.sut.sa.group.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.*;

import java.util.Collection;

import java.util.stream.Collectors;

import com.sut.sa.group.entity.Owner;

import com.sut.sa.group.entity.PetInfo;

import com.sut.sa.group.repository.OwnerRepository;

import com.sut.sa.group.repository.PetInfoRepository;

@CrossOrigin(origins = "http://localhost:4200")

@RestController

//@RequestMapping("/api")

public class PetInfoController {

    @Autowired private PetInfoRepository petInfoRepository;

    @Autowired private OwnerRepository ownerRepository;



    public PetInfoController(PetInfoRepository petInfoRepository){

        this.petInfoRepository = petInfoRepository;
```

```
}

@GetMapping("/petinfo")

public Collection<PetInfo> PetInfo() {

    return petInfoRepository.findAll().stream().collect(Collectors.toList());

}

@PostMapping("/petinfo/{petname}/{weight}/{hight}/{age}/{ownerid}")

public PetInfo newPetInfo(@RequestBody PetInfo newPetInfo,

                    @PathVariable String petname,

                    @PathVariable Double weight,

                    @PathVariable Double hight,

                    @PathVariable Integer age,

                    @PathVariable long ownerid

) {

    Owner petOwner = ownerRepository.findById(ownerid);

    newPetInfo.setPetName(petname);

    newPetInfo.setWiegth(weight);

    newPetInfo.setHight(hight);

    newPetInfo.setAge(age);
```

```
        newPetInfo.setPetowner(petOwner);

        return petInfoRepository.save(newPetInfo);

    }

    @GetMapping("/Petnames/getById/{petinfoId}")

    public PetInfo PetInfo2(@PathVariable Long petinfoId) {

        PetInfo  AreaData = petInfoRepository.findByPetId(petinfoId);

        return AreaData;

    }

}
```

DataBase

```
package com.sut.sa.group;

import java.util.Date;

import com.sut.sa.group.entity.Appointment;

import com.sut.sa.group.entity.Doctor;

import com.sut.sa.group.entity.Owner;

import com.sut.sa.group.entity.Period;

import com.sut.sa.group.entity.PetInfo;

import com.sut.sa.group.repository.AppointmentRepository;

import com.sut.sa.group.repository.DoctorRepository;

import com.sut.sa.group.repository.OwnerRepository;

import com.sut.sa.group.repository.PeriodRepository;

import com.sut.sa.group.repository.PetInfoRepository;

import org.springframework.boot.ApplicationRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.annotation.Bean;



@SpringBootApplication
```

```
public class GroupApplication {

	public static void main(String[] args) {

		SpringApplication.run(GroupApplication.class, args);

	}

	@Bean

	ApplicationRunner init(AppointmentRepository appointmentRepository,

                                        PetInfoRepository petInfoRepository,

                                        DoctorRepository doctorRepository,

                                        PeriodRepository periodRepository,

                                        OwnerRepository ownerRepository) {

		return args -> {

			Owner owner1 = new Owner();

			owner1.setOwnerName("Luffy");

			owner1.setOwnerLname("Monkey D.");

			owner1.setOwnerTel("0912832222");

			ownerRepository.save(owner1);



			Owner owner2 = new Owner();
```

```
owner2.setOwnerName("Zoro");

owner2.setOwnerLname("Roronoa");

owner2.setOwnerTel("0632418989");

ownerRepository.save(owner2);

Doctor doctor1 = new Doctor();

doctor1.setDoctorName("Tony Tony Chopper");

doctorRepository.save(doctor1);

Doctor doctor2 = new Doctor();

doctor2.setDoctorName("Black Jack");

doctorRepository.save(doctor2);

PetInfo petInfo1 = new PetInfo();

petInfo1.setPetName("Doremon");

petInfo1.setWiegth(40.00);

petInfo1.setHight(100.00);

petInfo1.setAge(90);

petInfo1.setPetowner(owner1);

petInfoRepository.save(petInfo1);

PetInfo petInfo2 = new PetInfo();
```

```
petInfo2.setPetName("Nobita");

petInfo2.setWiegth(57.50);

petInfo2.setHight(150.00);

petInfo2.setAge(10);

petInfo2.setPetowner(owner2);

petInfoRepository.save(petInfo2);

Period period1 = new Period();

period1.setPeriod("9.00-11.00");

periodRepository.save(period1);

Period period2 = new Period();

period2.setPeriod("13.00-16.00");

periodRepository.save(period2);

Appointment appointment1 = new Appointment();

appointment1.setPetInfo(petInfo2);

appointment1.setDoctor(doctor1);

appointment1.setAppDate(new Date());

appointment1.setPeriod(period1);

appointment1.setNote("ฉีดวัคซีน");
```

```
                    appointmentRepository.save(appointment1);

                    Appointment appointment2 = new Appointment();

                    appointment2.setPetInfo(petInfo1);

                    appointment2.setDoctor(doctor2);

                    appointment2.setAppDate(new Date());

                    appointment2.setPeriod(period2);

                    appointment2.setNote("ฉีดวัคซีน");

                    appointmentRepository.save(appointment2);

                    ownerRepository.findAll().forEach(System.out::println);

                    petInfoRepository.findAll().forEach(System.out::println);

                    doctorRepository.findAll().forEach(System.out::println);

                    periodRepository.findAll().forEach(System.out::println);

                    appointmentRepository.findAll().forEach(System.out::println);

            };

        }

}
```