AI 기술을 이용한
반려견 헬스케어 웹 서비스

A Dog Disease Diagnosis Web Service Using Machine Learning

201433931 황승환
201533970 전현우
201632230 이다은

CONTENTS

01 프로젝트 제안개요 아이디어 선정 이유

- 반려동물 1000만 시대, 반려동물 등록 현황은 매년 증가하는 추세
- 반려동물의 대부분은 개가 차지

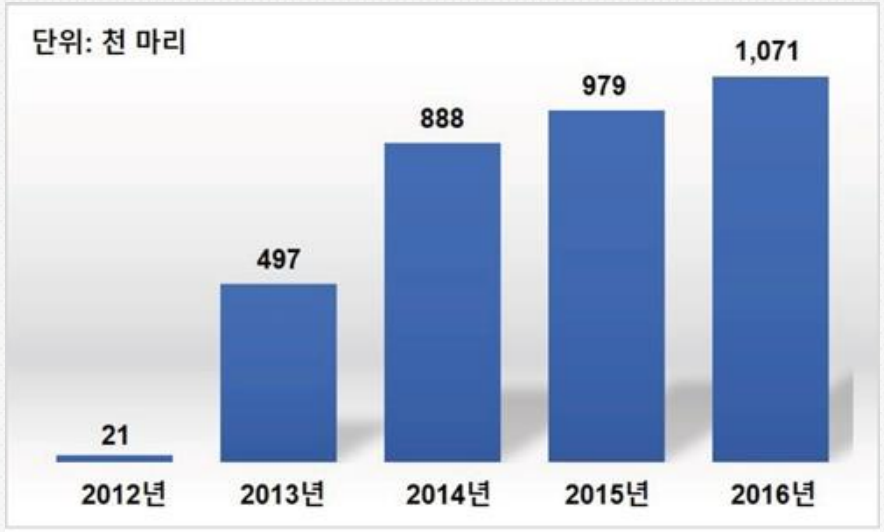
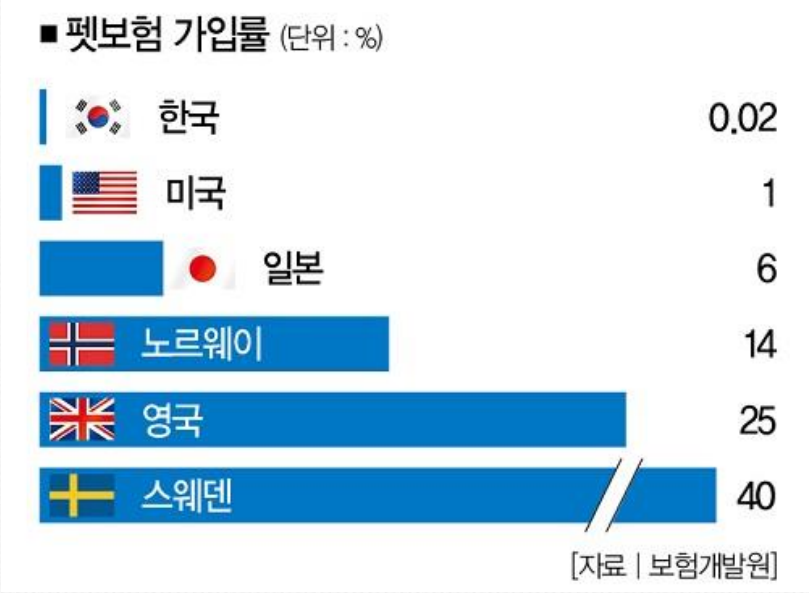


그림 8 반려동물 등록 현황(농림축산식품부 보도자료)

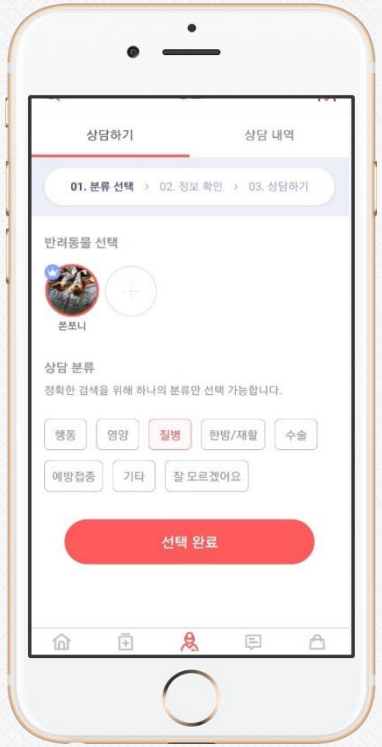
01 | 프로젝트 제안개요 아이디어 선정 이유

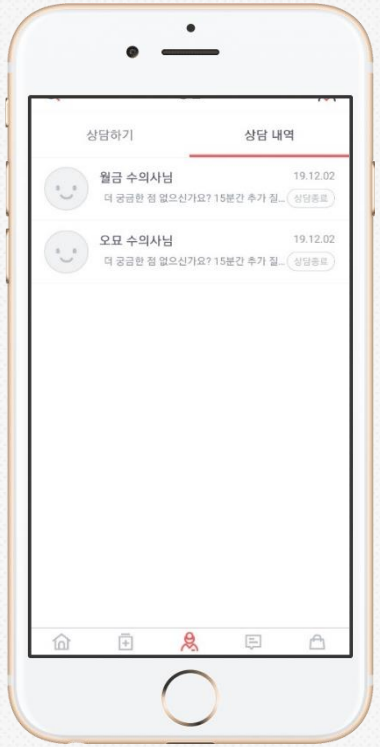
- 반려견을 위한 보험이 존재하나 나이 제한, 유전병 예외 조항, 비싼 보험료로 실효성 의문
- 보호자들의 동물병원비 지출 부담 증가



01 프로젝트 제안개요 벤치마킹 사례: 펫닥

- 실시간으로 수의사와 상담이 가능하도록 연결해주는 모바일 어플리케이션
- 수의사의 동물병원 전화번호도 공개되어 있어 필요시 해당 동물병원 방문 가능





01 프로젝트 제안개요 기존 서비스의 한계점

- 수의사 랜덤 지정
  - 수의사의 근무지가 자택에서 먼 경우 직접 방문 불가
- 동물병원 추천 기능 부재
  - 사용자가 상담한 수의사에 한하여 동물병원 정보 습득
  - 사용자 주변의 동물병원 확인 어려움
- 수의사와의 실시간 채팅
  - 긴급상황임에도 시간이 늦으면 상담하기 어려움

01 프로젝트 제안개요 기존 서비스와 차별점

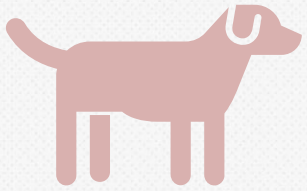
- ‘머신러닝’을 이용한 반려견 질병 진단 서비스
  - 공공데이터 포털에서 제공하는 오픈 데이터 전처리를 통한 신뢰성 확보
  - 랜덤 포레스트 알고리즘으로 정확도 높은 질병 진단 결과 제공
- 동물병원 추천 기능
  - 사용자 위치 기반 주변 동물병원 추천
  - 추천 동물병원 정보 조회

01 프로젝트 제안개요 기존 서비스와 차별점

- 한눈에 보는 데이터 시각화
  - 사용자는 예상되는 진단 결과에 대한 확률 시각화 정보를 조회
  - 관리자는 반려견의 성별에 따른 품종의 수 시각화 정보 조회
  - 관리자는 유행병 등 자주 걸리는 질병 빈도 시각화 정보 조회

01 프로젝트 제안개요 아이디어 기대효과

- 반려견에 초점을 맞춘 전문적인 서비스 제공
- 빠르고 간단한 방법으로 질병 진단 및 예방 가능
- 보호자들의 동물병원비 지출 부담 감소
- 데이터 시각화로 다양한 정보 제공 및 습득
- 동물병원 추천으로 사용자 편리성 증대



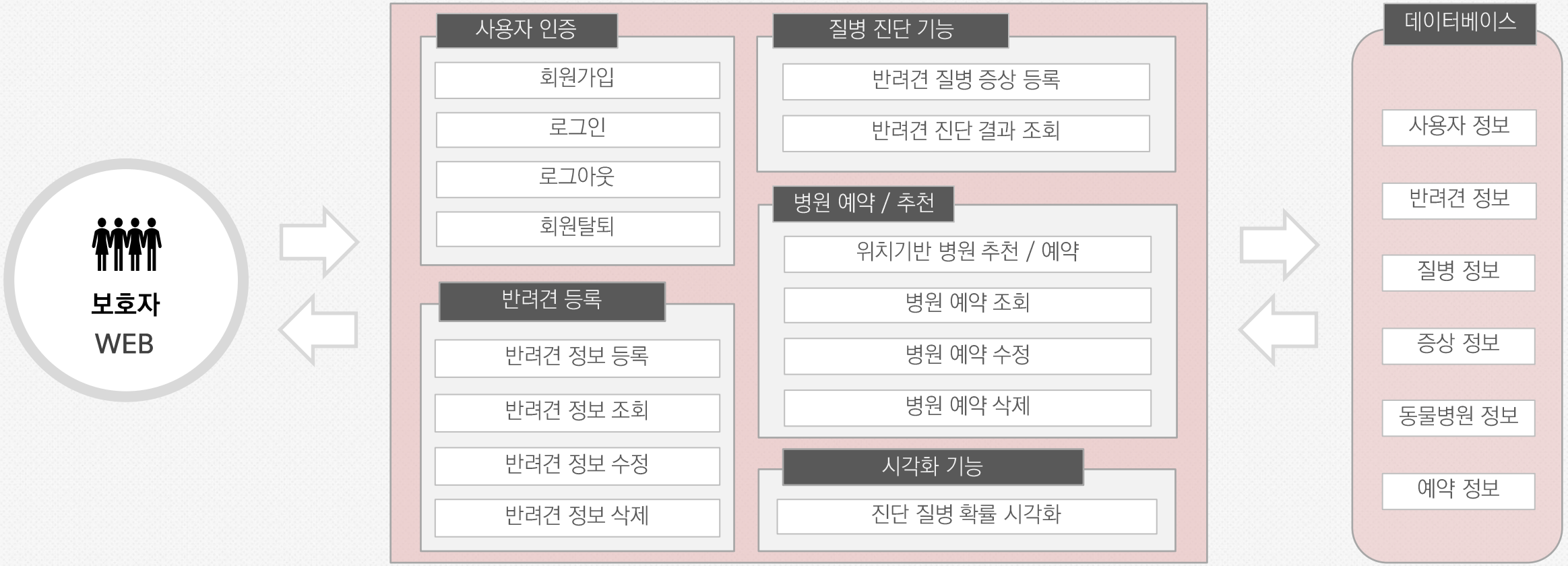
02 | 프로젝트 개발내용 서비스 시나리오

1. 보호자는 로그인/회원가입으로 사용자 정보를 인증한다.
2. 관리자는 보호자를 관리한다.
3. 서버는 보호자 정보를 출력한다.
4. 보호자는 강아지의 정보를 입력한다.
5. 서버는 강아지 정보를 출력한다.
6. 보호자는 강아지 질병 진단을 진행한다.
7. 보호자는 질병 진단 시각화 자료를 조회한다.
8. 서버는 질병 진단 시각화 정보를 출력한다.
9. 보호자는 질병 진단 정보를 조회한다.
10. 보호자는 주소 기반 동물병원 추천을 요청한다.
11. 보호자는 추천 받은 동물병원 정보를 조회한다.
12. 보호자는 동물병원 예약 정보를 조회한다.
13. 서버는 동물병원 예약 정보를 출력한다.
14. 관리자는 AI 모델을 관리한다.

02 | 프로젝트 개발내용 서비스 시나리오

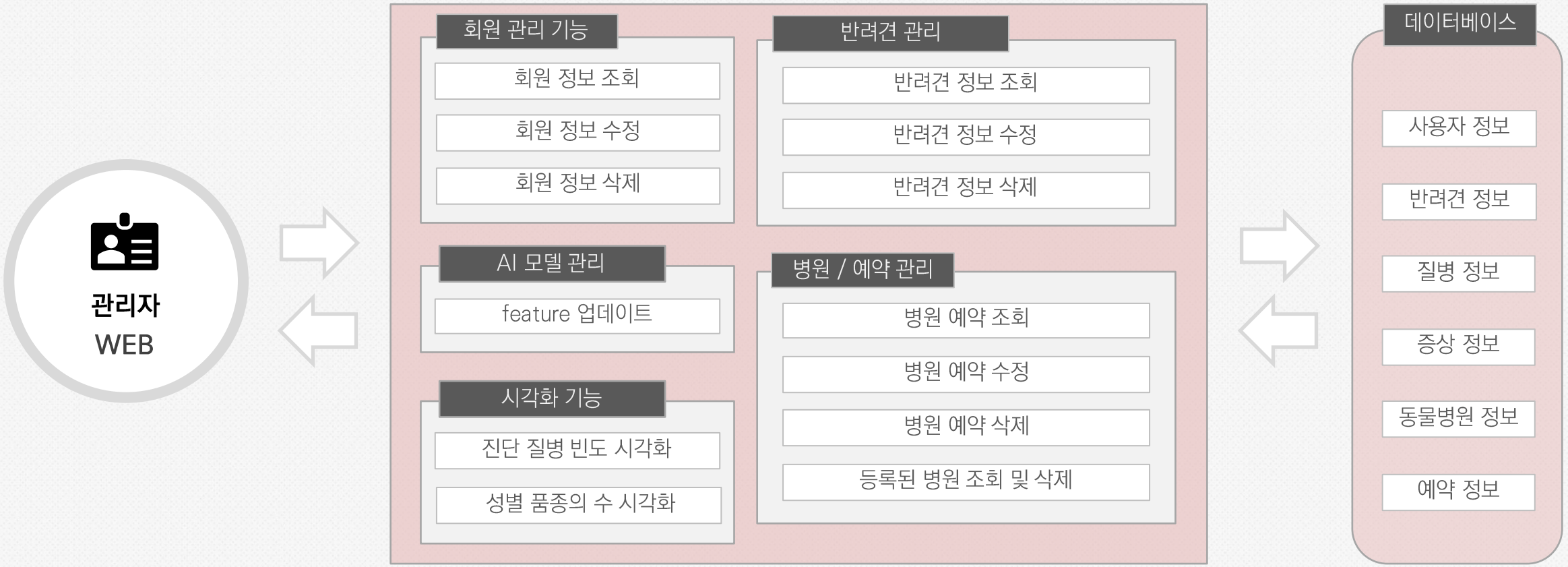
1. 수의사는 로그인/회원가입으로 사용자 정보를 인증한다.
2. 관리자는 수의사를 관리한다.
3. 서버는 수의사 정보를 출력한다.
4. 수의사는 동물병원 등록을 요청한다.
5. 수의사는 등록된 동물병원 정보를 조회한다.
6. 서버는 동물병원 정보를 출력한다.
7. 수의사는 예약 내역을 조회한다.
8. 서버는 예약 내역을 출력한다.

02 | 프로젝트 개발내용 보호자 기능 개발내용



02 | 프로젝트 개발내용 수의사 기능 개발내용

02 | 프로젝트 개발내용 관리자 기능 개발내용



02 | 프로젝트 개발내용 개발 기술 스택

클라이언트 단

템플릿


웹 서버


외부 API

어플리케이션 단

프레임워크


API

데이터 단

서버

데이터베이스


AI 단

서버

라이브러리


외부 API

03 | 프로젝트 개발전략 개발팀 구성 및 역할

이 다 은

- 팀원/일정/성과 관리
- 프로젝트 버전 관리
- 데이터베이스 설계
- AI 모델 관리 기능 구현
- 질병 데이터셋 전처리
- 질병 진단 데이터 시각화

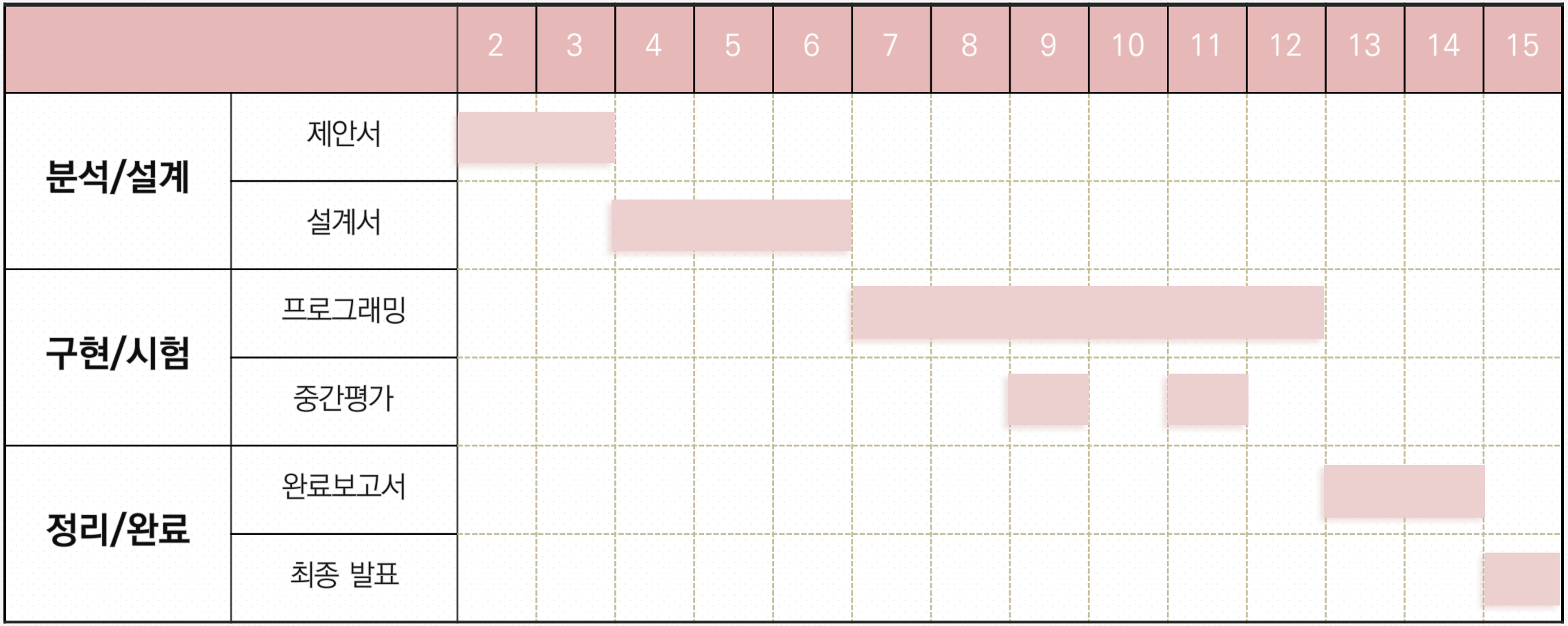
전 현 우

- 회원 관리 기능 구현
- REST API 설계
- MVC 모델 설계
- 예약 관리 기능 구현
- AI 모델 학습 및 테스트
- AI 질병 진단 기능 구현

황 승 환

- UX/UI 설계 및 디자인
- 사용자 인증 기능 구현
- 관리자 인증 기능 구현
- 병원 관리 기능 구현
- 동물병원 추천 기능 설계
- 동물병원 추천 기능 구현

03 | 프로젝트 개발전략 개발 일정 및 계획



04 | 개발 소스코드 AI 질병 진단 (데이터 학습)

```
# train dataset
df = pd.read_csv('./data/clean/dog_disease_train.csv', encoding='cp949')
df[request.form["add"]] = 0
cols = df.columns.to_list()
tmp = cols[-1]
del cols[-1]
cols.insert(-1, tmp)
df = df[cols]
df.to_csv("./data/clean/dog_disease_train.csv", mode ='w', index=None, encoding='cp949')
```

학습 데이터 불러오기

```
# test dataset
df_test = pd.read_csv('./data/clean/dog_disease_test.csv', encoding='cp949')
df_test[request.form["add"]] = 0
cols_test = df_test.columns.to_list()
tmp_test = cols_test[-1]
del cols_test[-1]
cols_test.insert(-1, tmp_test)
df_test = df_test[cols_test]
df_test.to_csv("./data/clean/dog_disease_test.csv", mode ='w', index=None, encoding='cp949')
```

테스트 데이터 불러오기

04 | 개발 소스코드 AI 질병 진단 (데이터 학습)

```
# train, validate dataset division
X = df.iloc[:, :-1]
y = df['질병명']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = float(request.form["test_size"]),
                                                    random_state = int(request.form["random_state"]))
rf_clf = RandomForestClassifier()

# train the model
rf_clf.fit(X_train, y_train)

with open("dog_disease_model.pkl", 'wb') as fid:
    pickle.dump(rf_clf, fid)
```

모델학습 및 저장

04 | 개발 소스코드 AI 질병 진단 (학습 결과 도출)

```
loaded_model = pickle.load(open("dog_disease_model.pkl", "rb"))

with open(json_path, "r", encoding='utf-8-sig') as json_file:
    symptoms_dict = json.load(json_file)
    symptoms_dict = symptoms_dict['symptoms'][0]

input_vector = np.zeros(len(symptoms_dict))
```

모델 불러오기

```
for i in range(len(request.form)):
    symp.append(request.form["증상" + str(i)]) # value 값 저장

if(len(request.form)!=0): # 세션이 비어 있을 경우 값을 추가하여 넣어준다
    for i, j in enumerate(symp):
        if(symp[i] != ''):
            _symp.append(symptoms_dict[j]) # 병명 저장

    input_vector[_symp] = 1 # 해당 인덱스 1로 표시
```

증상 불러오기

04 | 개발 소스코드 AI 질병 진단 (학습 결과 도출)

```
return jsonify({
    "data": loaded_model.predict([input_vector])[0],
    "기관지 확장증": loaded_model.predict_proba([input_vector])[0][0],
    "마카다미아너트 중독증": loaded_model.predict_proba([input_vector])[0][1],
    "코로나 바이러스": loaded_model.predict_proba([input_vector])[0][2],
})
else: # 세션이 비어 있지 않을 경우
input_vector[_symp] = 1 # 해당 인덱스 1로 표시
symp = []
_symp = []

return jsonify({
    "data": loaded_model.predict([input_vector])[0],
    "기관지 확장증": loaded_model.predict_proba([input_vector])[0][0],
    "마카다미아너트 중독증": loaded_model.predict_proba([input_vector])[0][1],
    "코로나 바이러스": loaded_model.predict_proba([input_vector])[0][2],
})
```

최종 결과 도출

04 | 개발 소스코드 보호자 – 질병 진단 결과 시각화 (구글 차트 API)

```
let dogName = document.getElementById("dogName");
dogName.innerHTML = dogName.textContent + "의 질병 진단 결과";

google.charts.load('current', {'packages':['bar']});
google.charts.setOnLoadCallback(drawColumnChart);

function drawColumnChart() {

    let tbody = document.getElementsByTagName("tbody");

    let diagVisData = [];
    diagVisData.push(['질병 이름', '비중']);

    let macak = tbody[0].children[0].children[0].textContent;
    let corna = tbody[0].children[0].children[1].textContent;
    let respir = tbody[0].children[0].children[2].textContent;

    diagVisData.push(['마카다미아너트 중독증', parseFloat(macak)*100]);
    diagVisData.push(['코로나 바이러스 감염증', parseFloat(corna)*100]);
    diagVisData.push(['기관지 확장증', parseFloat(respir)*100]);
```

```
var data = google.visualization.arrayToDataTable(diagVisData);

    var options = {
        width: 800,
        legend: { position: 'none' },
        axes: {
            x: {
                0: { side: 'top' } // Top x-axis.
            }
        },
        bar: { groupWidth: "90%" }
    };

    var chart = new
google.charts.Bar(document.getElementById('diagColumnChart'));

    chart.draw(data, google.charts.Bar.convertOptions(options));
};
```

화면에 차트 출력

04 | 개발 소스코드 관리자 – 질병 진단 결과 시각화 (구글 차트 API)

```
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawPieChart);

function drawPieChart() {

    let diseaseBody = document.getElementsByTagName("tbody");
    let digNameTR = document.getElementsByClassName("digNameTR");

    let diagName = [];
    let diagCount = [];
    for(let i = 0; i < digNameTR.length; i++) {
        diagName.push(diseaseBody[0].children[i].children[0].textContent);
        diagCount.push(parseInt(diseaseBody[0]
        .children[i].children[1].textContent));
    }

    let diagData = [];
    diagData.push(['진단 질병 이름', '빈도']);

    for(let i = 0; i < digNameTR.length; i++) {
        let tempData = [];
        tempData.push(diagName[i]);
        tempData.push(diagCount[i]);

        diagData.push(tempData);
    }

    var data = google.visualization.arrayToDataTable(diagData);

    var options = {
        title: '질병 이름에 따른 질병 진단 정보',
        slices: {
            0: { color: '#3399FF' },
            1: { color: 'orange' },
            2: { color: '#9933CC' }
        }
    };

    var chart = new
google.visualization.PieChart(document.getElementById('digPieChart'));

    chart.draw(data, options);
}
```

화면에 차트 출력

04 | 개발 소스코드 병원 추천 (카카오 지도 API)

```
// 검색결과 항목을 Element로 반환하는 함수입니다
function getListItem(index, places) {
    let hospitalBody = document.getElementsByTagName("tbody");
    let hospitalTR = document.getElementsByClassName("hospitalTR");

    let hosName = [];
    let hosAddr = [];
    for(let i = 0; i < hospitalTR.length; i++) {
        hosName.push(hospitalBody[1].children[i].children[0].textContent);
        hosAddr.push(hospitalBody[1].children[i].children[1].textContent);
    }

    var el = document.createElement('li'),

        itemStr = '<span class="markerbg marker_' + (index+1) + '"></span>' +
            '<div class="info" id="hosInfo' + (index+1) + '">' +
            '   <h5 class="hospital_name' + (index+1) + '" id="' + (index+1) + '" ' +
            ' onclick="openModal(this.id)">' + places.place_name + '</h5>';

    if (places.road_address_name) {
        itemStr += '    <span id="road' + (index+1) + '">' + places.road_address_name + '</span>' +
            '   <span class="jibun gray" id="jibun' + (index+1) + '"> ' + places.address_name + '</span>';
    } else {
        itemStr += '    <span>' +  places.address_name  + '</span>';
    }
```

```
// 병원이름 클릭시 모달창을 띄우는 이벤트
function openModal(id){

    hospital_name = $(".hospital_name" + id).html();
    road = $("#road" + id).html();
    tel = $("#tel" + id).html();

    $("#modal_div").modal();
}
// 이벤트 끝

// 모달창 예약하기 눌렀을 때 이벤트
function completeReserve(){
    let date = $("#visit_date").val();
    let description = $("#description").val();
    let dog = $("#choice").val();

    let data = {
        address : road,
        date : date,
        description : description,
        name : hospital_name,
        tel : tel,
        dog : dog
    };
```

모달창 오픈

카카오 API를 통해 데이터를 가져옴

04 | 개발 소스코드 병원 추천 (카카오 지도 API)

```
List<DiseaseResponseDto> diseaseAll = diseaseService.findAllDesc();
List<HospitalResponseDto> hospitalList = hospitalService.findAllDesc();

if(member != null) {

    model.addAttribute("member", member);
    model.addAttribute("diagnosis", jsonObj.get("data"));
    model.addAttribute("macak", jsonObj.get("마카다미아너트 중독증"));
    model.addAttribute("corna", jsonObj.get("코로나 바이러스"));
    model.addAttribute("bronchus", jsonObj.get("기관지 확장증"));
    model.addAttribute("diseases", diseaseAll);
    model.addAttribute("forms", form);
    model.addAttribute("hosList", hospitalList);
}

    return "members/recommends/recommendation";
}
```

병원 리스트를 가져옴

모델 전달

```
<!-- 동물병원 리스트 -->
<table id="hospitalTable">
  <tbody id="hospitalBody">
    <tr th:each="hospitals : ${hosList}" class="hospitalTR" hidden="true">
      <td th:text="${hospitals.name}"></td>
      <td th:text="${hospitals.address}"></td>
    </tr>
  </tbody>
</table>
```

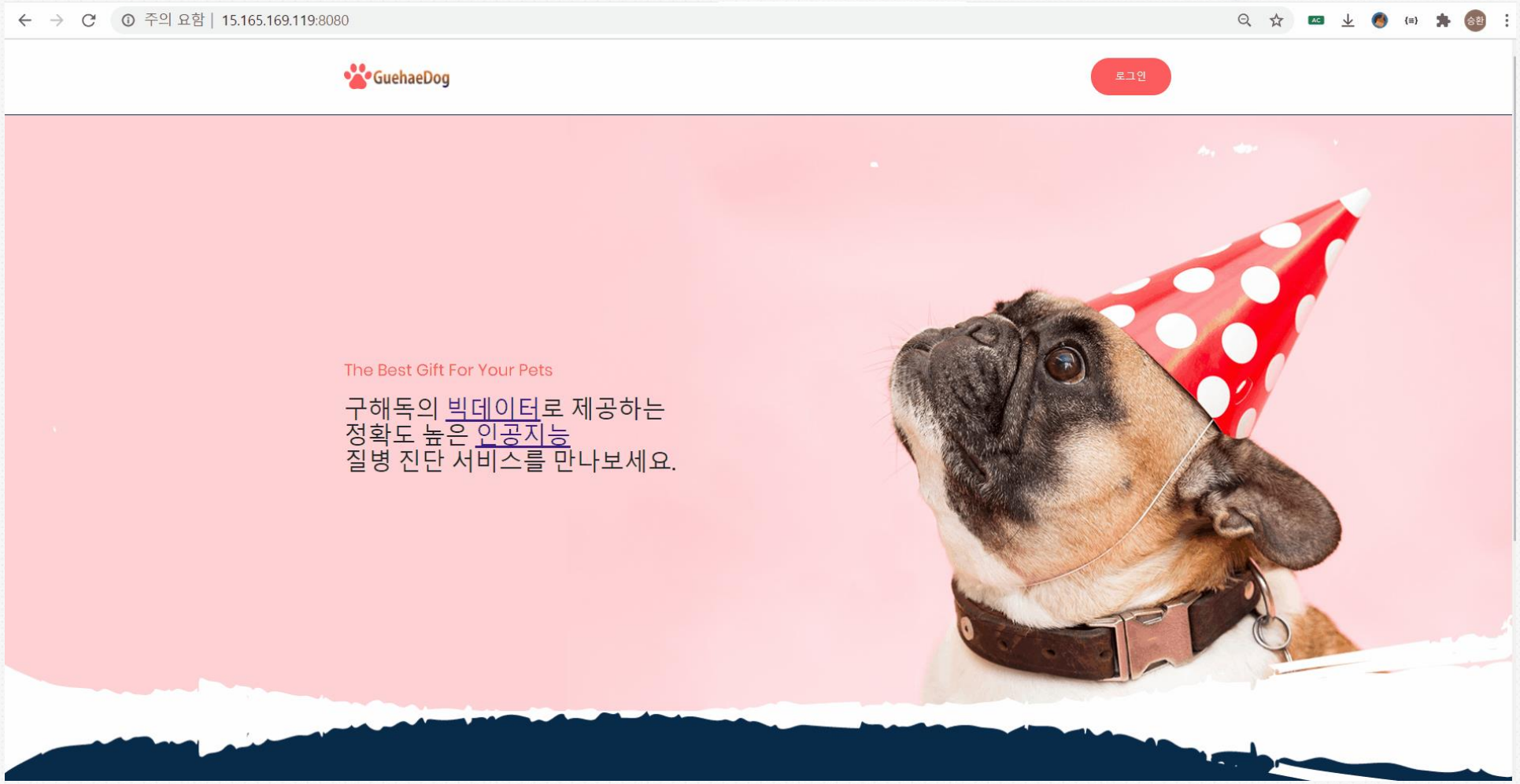
hidden 속성으로
병원 정보를 가져옴

```
el.innerHTML = itemStr;
el.className = 'item';

if(hosName.includes(places.place_name) && hosAddr.includes(places.road_address_name)) {
    el.children[1].onmouseover = function(e) {
        e.currentTarget.children[0].style.color = "blue";
    }
    el.children[1].onmouseout = function(e) {
        e.currentTarget.children[0].style.color = "black";
    }
}
else {
    el.children[1].children[0].onclick = null;
    el.style.cursor = "default";

    let idxDiv = 4;
    if(!places.road_address_name) {
        idxDiv = 3;
    }

    for(let j = 0; j < idxDiv; j++) {
        el.children[1].children[j].style.color = "#D3D3D3";
    }
}

    return el;
}..
```

병원이 등록된 경우

병원이 미 등록된 경우

05 | 프로젝트 시연


주의 요함 | 15.165.169.119:8080
GuehaeDog
로그인
The Best Gift For Your Pets
구해독의 빅데이터로 제공하는
정확도 높은 인공지능
질병 진단 서비스를 만나보세요.