

응급실 혼잡도 분석을 통한 의료서비스 극대화

손하지

Tel : 010-7987-3197

Github : <https://github.com/doHand/portfolio>

Project : <http://43.203.78.27:8080/>



CONTENTS

1. 프로젝트 기획서

2. 사용 기술 목록

3. 사용 기술 경험

4. 개발 스케줄표

5. 빅데이터 분석 결과서

6. 요구사항 정의서

7. 화면 설계서

8. 주요 기능 및 소스코드

9. UseCase Diagram

10. Class Diagram

11. Sequence Diagram

12. ER Diagram

13. Software Architecture

14. 개발 향후 계획

15. 프로젝트 소감

사용 기술 목록

개발 환경



[Spring
framework]
3.9.18.ver



[Eclipse]
2019.09.ver



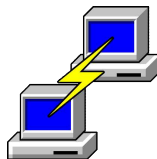
[Visual Studio
Code]



[R]
4.3.2.ver

ORACLE

[Oracle]
19.2.1.ver



[PuTTY]
0.70.2.ver



[Docker]
24.0.5.ver



[AWS EC2]
(Ubuntu)

사용 언어



[jdk 11.0.9]



[JSP]



[HTML,CSS,JS]



[MyBatis]



[SQL]

OS



ubuntu

[Ubuntu]



[Window]

웹 서버



[Tomcat]
9.0.84.ver

사용 기술 경험

JAVA

- JAVA 1.8 기반 미니프로젝트
- 주소록관리 시스템 (입력, 출력,검색, 삭제, 역순)
- Single Linked List 구조 (노드 삽입, 역순, 삭제)
- 컬렉션프레임워크(List, Set, Map)
- Multi-Thread와 동기화 메소드

HTML / CSS

프로젝트

- 전체 웹사이트 페이지와 네비게이션 바를 반응형으로 제작
- 메인 페이지, 응급실 정보 페이지, 응급실 현황 페이지의 구조와 각 섹션 제작

JavaScript

프로젝트

- 동적으로 웹페이지 생성
- 사용자 요청에 따라 다른 내용이 표시되는 웹페이지 생성
- Form태그를 활용해 데이터 전송 처리, 데이터베이스와의 상호작용 작업 수행
- MVC 패턴을 이용한 동적 웹페이지 제공

Bootstrap

프로젝트

- 전체적인 홈페이지 반응형으로 제작
- 그리스 시스템을 활용한 레이아웃 조정
- 버튼, 폼, 네이게이션 등과 같은 다양한 UI 구성

JS

프로젝트

- DOM을 조작하여 웹페이지의 요소를 동적으로 업데이트
- Fetch API를 사용하여 병원 추천, 지역 정보, 의료 정보를 비동기적으로 업데이트하면서 동적으로 표시
- CSS와 함께 애니메이션 효과 적용
- Chart.js를 이용하여 데이터를 시각적으로 표현

DataBase

프로젝트

- 의료 데이터에 대한 SQL 처리 및 테이블 관리, 시퀀스 작성
- Oracle 데이터베이스를 이용한 다양한 쿼리문 실습
- 사용자 인증 및 권한관리 실습
- 데이터 검색 및 조회 시퀀스 작성
- 데이터 저장 및 관리 시퀀스 작성

R

프로젝트

- 디지전트리를 이용한 응급실 병원 추천 예측
- rpart함수를 사용하여 트리 모델 생성
- predict 함수를 사용하여 예측
- prp 함수를 사용하여 트리 시각화
- mean 함수를 사용하여 정확도 계산

Git

프로젝트 관리를 위한 기본적인 깃 명령어 사용 경험 (gitbash 사용)

소스트리(Sourcetree)활용 경험

프로젝트 업로드, 삭제, 머지 실습

사용 기술 경험

Servlet

프로젝트

- HTTP 요청 처리 후 요청에 따라 적절한 작업 수행 (GET, POST 사용)
- JSON 콘텐츠를 생성하여 DB에서 정보를 검색하거나 다른 소스로부터 데이터를 가져와 웹페이지 동적으로 처리
- JDBC API 사용하여 데이터베이스와 연동하여 데이터베이스 작업 수행

AWS

프로젝트

- EC2 프리티어 인스턴스 생성
- Pem key 생성 후 puTTYgen을 통해 개인키 (.ppk) 생성
- puTTY를 활용하여 ssh 환경 접속
- webapps에 파일질라를 통해 프로젝트 war파일 전송
- 탄력적 Ip 주소 설정
- 인바운드 규칙 설정
- 도메인 구매 후 Route53 DNS 호스팅 설정

Spring

프로젝트

- 객체 간의 의존성을 외부에서 설정하여 객체 간의 결합도를 낮춤
- Spring MVC 아키텍처를 기반으로 변경
- Model을 활용한 데이터 처리
- myBatis와 연동하여 데이터 처리

Docker

프로젝트

- oracle 설치 후 AWS 환경에 DB 연결
- Docker 이미지를 통한 파일 pull
- 이미지 인스턴스를 컨테이너로 등록
- Docker repository에 이미지 push
- AWS에 DB 컨테이너 환경 설정
- Timezone 설정

Mybatis

프로젝트

- SQL 쿼리를 Xml 파일에 작성하여 매핑 구성을 정의
- SQL 쿼리의 매개변수를 자동으로 매핑하여 필요한 파라미터 설정
- 동적 SQL을 작성하여 코드 중복을 최소화하고 유연성을 높임

Linux / Ubuntu

- 버추얼박스를 통한 우분투 활용 경험
- 간단한 명령어 (ls, mkdir, mv, rm, sudo 등) 활용 경험
- Vim 을 통한 파일 편집 경험
- Chown, chmod을 이용한 파일 소유자, 권한 변경 경험
- 패키징툴을 이용한 프로그램 설치 (자바, 오라클, 톰캣 등) 및 프로젝트 배포 경험

기획 의도

사회 >

응급실 7곳 뱅뱅이 80대 숨져... 호흡곤란 신생아도 3시간 헤맸다

의료 공백 악몽 현실화

김준호 기자 우정식 기자

업데이트 2024.02.28. 15:20

가

YTN

YTN브랜드채널

정치 경제 사회 전국 국제 과학 문화 스포츠 연예 날씨 이슈 시리즈 TV프로그램

119구급대원도 '응급실 뱅뱅이'...병원 찾느라 '발 동동'

2024.03.02. 오전 05:52



기술은 나날이 발전되고 있지만,
중증환자가 제때 치료받지 못하는 일명

'응급실 뱅뱅이'

의 문제는 점점 심각해지고 있습니다.

원인으로는
응급실 과밀화 와 전문의 부족



낮아지고 있는 의료 만족도와 신뢰도

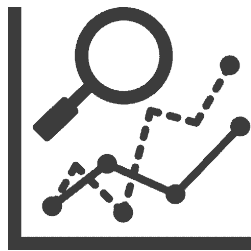
개발 목표



1. 실시간 응급실 정보 제공 및 추천 서비스

1.1 응급실 실시간 정보 제공

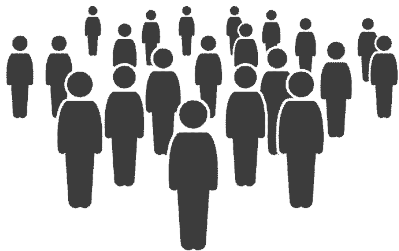
1.2 디시전 트리를 통한 위치 기반 응급실 추천 서비스



2. 혼잡도 예측 서비스

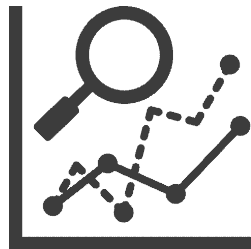
2.1 다중 선형 회귀 분석을 통한
지정 시기 및 지역의 응급실 혼잡도 예측

개발 목표



3. 인적자원 재배치

3.1 서울특별시와 경기도를 비교해
효율적인 응급의학과 전문의 재배치 정보 제공



4. 응급실 관련 정보 제공

4.1 의료 형평성을 위한 인적자원 재배치와
응급실 혼잡도 데이터 시각화 서비스

Project 1

[illegible]

Project 2

[illegible]

개발 스케줄표

Project 3

주제: 응급실 혼잡도 분석을 통한 의료 서비스 극대화 1팀 3차														
구분	담당자	작업(산출물)	시작일	종료일	작업기간	2월				3월				
						26	27	28	29	1	2	3	4	5
손하지		Servlet으로 구현된 혼잡도 페이지 분석	2024-02-26	2024-02-28	3									
		[Model]VO 구현	2024-02-29	2024-02-29	1									
		[MyBatis] MyBatis 이용하여 Repository 구현	2024-03-01	2024-03-01	1									
		[Service]지역 기반 검색 기능 구현	2024-03-02	2024-03-03	2									
		[Service]날짜 기반 검색 기능 구현	2024-03-03	2024-03-04	2									
		[Controller] MainController 구현	2024-03-04	2024-03-04	1									
		[View,JSP] 의사 재배치 페이지 spring으로 연결 후 Chart.js 수정	2024-03-05	2024-03-05	1									
		[View,JSP] 응급실 정보 페이지 spring으로 연결	2024-03-05	2024-03-05	1									

요구사항 분석서

목차

1. 기능적 요구사항

1.1 메인기능

2. 응급실 혼잡도 예측 기능

2.1 날짜 검색 기능

2.2 정보 제공 기능

2.3 지도 표시 기능

3. 인적자원 재배치 분석 기능

3.1 인적자원 예측 결과 표시

3.2 OECD 평균 자료 제공

1. 요구 기능

0. 공통 기능

- 메뉴 클릭을 통해 다른 기능으로 이동할 수 있어야 한다.
 - 응급실 정보, 응급실 현황, 혼잡도 예측, 인적자원 재배치, 로그인으로 메뉴가 구성될 수 있어야 한다.
- 로그인 클릭 시, 메인화면으로 이동할 수 있어야 한다.
- 홈페이지 최 하단에 로고를 통해 메인화면으로 이동할 수 있어야 한다.

1. 메인 기능

- 움직이는 화면(캐러셀)이 있어야 하고 각 화면을 이동할 수 있는 버튼이 있어야 한다.
 - 캐러셀 화면은 총 3개의 화면으로 구성될 수 있어야 한다.
 - 캐러셀 화면 왼쪽에는 페이지에 대한 소개가 있어야 한다.
- 주요 기능(응급실 정보, 응급실 현황, 혼잡도 예측, 인적자원 재배치)으로 이동할 수 있는 버튼이 있어야 한다.
 - 페이지 형태로 각각의 주요 기능을 표현 할 수 있어야 한다.
 - 버튼 상단 부에 각 기능에 관한 이미지와 간략한 설명을 적을 수 있어야 한다.

2. 응급실 혼잡도 예측 기능

2-1 날짜 검색 기능

- 날짜를 선택할 수 있어야 한다.
 - 년도와 월을 선택할 수 있어야 하고, 드롭다운 형태로 제공되어야 한다.
- 검색 버튼을 클릭하기 위해서는 날짜가 선택되어 있어야 한다.

2-2 정보 제공 기능

- 날짜 검색 버튼을 클릭 시, 혼잡도 예측 결과가 여유, 적정, 혼잡 3가지 단계를 통해 텍스트로 구현되어야 한다.
- 일정한 지역과 날짜를 기반으로 히스토리에 계산되어 저장된 다음 선택한 월의 응급실 관련 자료를 막대 그래프로 보여줄 수 있어야 한다.
 - 60분 내 응급실 이용률, 관내 응급실 이용률, 응급실 30분 내 접근 불가능한 인구 비율을 그래프로 표현할 수 있어야 한다.
- 크게 보기 버튼을 클릭 시, 그래프를 확대해서 보여줄 수 있는 팝업창 기능이 표현될 수 있어야 한다.

2-3 지도 표시 기능

- 지도 위에 오버레이를 클릭 시, 지역과 전국을 비교한 정보가 막대 그래프로 표시될 수 있어야 한다.
- 지도의 확대, 축소가 불가능 해야 한다.

- 오버레이 위에 마우스 오버 시, 색상이 변환될 수 있어야 한다.

3. 인적자원 재배치 예측 기능

3-1 인적자원 예측 결과 표시

- 서울시와 경기도의 응급의학전문회의 남녀 비율이 도넛 차트로 표현될 수 있어야 한다.
 - 각 그래프는 다른 색상을 사용하여 표현하여야 한다.
- 인적자원 재배치 분석 결과를 왼쪽 하단에 표현할 수 있어야 하고, 밑에 버튼을 삽입하여 클릭 시 계산식이 드롭다운 형태로 표시될 수 있어야 한다.
- 응급의학과 전문의 재배치 예측 결과가 수치화 되어 텍스트로 구현되어야 한다.

3-2 OECD 평균 자료 제공

- 평균 OECD 의사, 병상 수와 한국 의사, 병상 수가 텍스트로 구현되어야 한다.
- 1,000명 기준 한국과 OECD를 비교한 자료가 인구 대비 병상, 1인당 병원 외래 진료(건), 임상 의사 수 (만원사 포함)가 테이블로 표시될 수 있어야 한다.
- 인구 1,000명 기준으로 19개국의 (한국, 멕시코, 일본, 캐나다, 프랑스 등) OECD 국가 임상 의사 수가 막대 그래프로 구현 될 수 있어야 한다.

3-3 우리나라 의사 수 비교

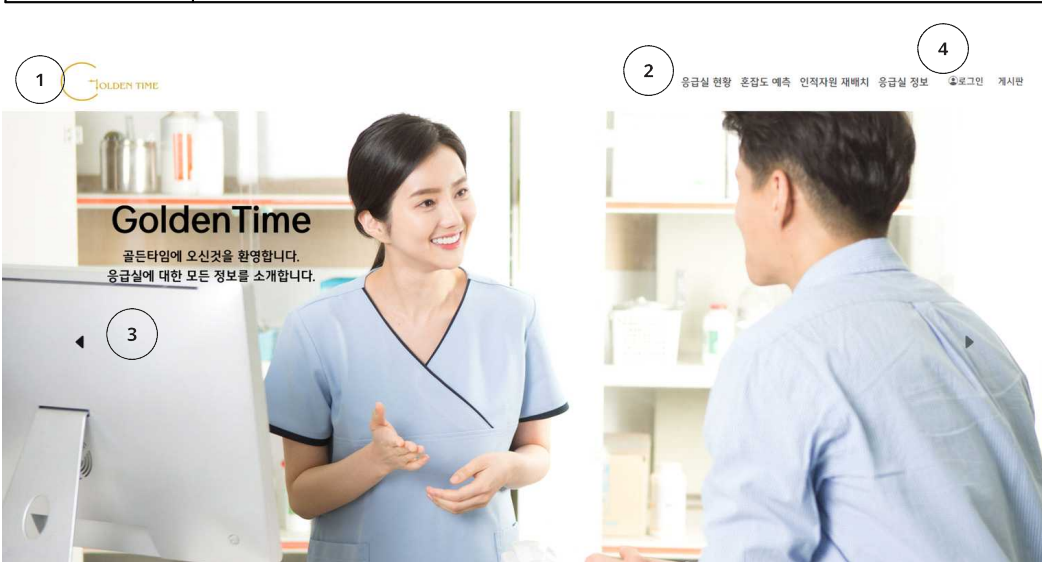
- 인구 1,000명당 평균 의사 수와 2.3명 기준 부속 의사 수가 라인과 바를 혼합한 혼합 그래프로 구현 될 수 있어야 한다.
 - 라인과 바 그래프는 다른 색상을 사용하여 표현해야 한다.
- 서울 & 경기 재배치 예측 결과 보러 가기 버튼을 클릭 시, 예측 데이터를 제공하는 상단 부분으로 이동 될 수 있어야 한다.

요구사항 명세서

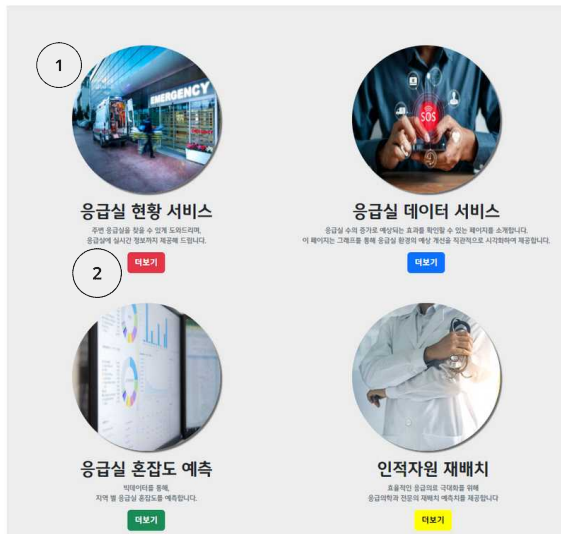
요구사항 명세서 - Code-911-2
혼잡률 분석을 통한 의료서비스 극대화

RQ-ID	화면 명	요구사항 명	요구사항 상세
RQ-ID-000-01	전체 페이지	상단 메뉴바	Home, 응급실 정보(응급실 위치 제공), 응급실 현황(실시간 응급 정보 제공), 혼잡도 예측(혼잡도 예측 정보 제공),인적자원 재배치(의사 수 현황, 재배치 분석 정보 제공),로그인으로 구분되어 있고 각 구분된 영역 클릭 시 해당 기능으로 이동
RQ-ID-000-02	전체 페이지	로고	웹 페이지 왼쪽 상단 클릭 시, 첫 화면으로 이동할 수 있는 로고를 삽입
RQ-ID-000-03	전체 페이지	웹 페이지 최하단 링크	웹 페이지 로고를 클릭하면 첫 화면으로 이동할 수 있는 링크를 최하단부에 삽입
RQ-ID-001-01	메인화면	화면 이동 버튼	각 페이지에 대한 설명과 버튼을 삽입하여 클릭 시 해당 페이지로 이동
RQ-ID-001-02	메인화면	메인 화면 이미지	자동으로 움직이는(캐러셀) 이미지를 상단 부분에 설정, 수동으로 이미지를 조정할 수 있는 버튼을 양옆에 삽입
RQ-ID-001-03	메인화면	몸통 부분	몸통 부분에 각 기능으로 이동할 수 있는 박스 및 이미지를 삽입
RQ-ID-002-01	혼잡도 예측	날짜 선택	왼쪽 상단에 날짜를 선택할 수 있는 기능을 삽입
RQ-ID-002-02	혼잡도 예측	날짜 검색 기능	왼쪽 상단에 날짜를 입력한 뒤, 날짜 정보를 전송 할 수 있는 버튼을 삽입
RQ-ID-002-03	혼잡도 예측	예측 정보 제공	왼쪽 중반에 날짜 검색을 통한 응급실 혼잡도 예측 정보를 제공
RQ-ID-002-04	혼잡도 예측	시각화 자료 제공	응급실 이용 지표를 왼쪽 하단에 막대 그래프로 시각화하여 정보를 제공
RQ-ID-002-05	혼잡도 예측	시각화 자료 버튼	클릭시 시각화 자료를 크게 보여줄 수 있는 팝업창이 뜰 수 있도록 설정
RQ-ID-002-06	혼잡도 예측	지역 선택 지도	정보를 보고싶은 지역을 오버레이 형태로 제공하여 클릭 시 해당 지역의 응급실 혼잡도를 확인할 수 있는 시각화 자료 제공
RQ-ID-003-01	인적자원 재배치	버튼 삽입	클릭하여 계산식 확인하기'버튼을 삽입하여 클릭 시 계산식이 드롭다운 형태로 출력될 수 있어야한다
RQ-ID-003-02	인적자원 재배치	계산식 자료 제공	버튼 클릭 시 계산식이 보일 수 있어야 한다.
RQ-ID-004-01	응급실 정보	버튼 삽입	서울 & 경기 재배치 예측 결과 보러가기 버튼을 삽입하여 클릭 시 인적자원 재배치 페이지로 이동할 수 있어야 한다.
RQ-ID-004-02	응급실 정보	차트 삽입	마우스 오버 시 데이터가 나오는 차트가 제공 될 수 있어야 한다

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	메인페이지_상단부	페이지	1-1
화면경로	메인페이지				

		설명
 <p>1 GoldenTime</p> <p>2 응급실 현황 혼잡도 예측 인적자원 재배치 응급실 정보 로그인 게시판</p> <p>3</p> <p>4</p>		
	1	<ul style="list-style-type: none"> 모든페이지 상단에 있는 로고 클릭 시 메인 첫 화면으로 이동 로고는 화면상단에 항상 고정
	2	<ul style="list-style-type: none"> 메뉴는 크게 4가지(실시간 응급실정보, 혼잡도 예측, 의사재배치, 응급실 정보)로 구성 마우스 오버 시 구분될 수 있도록 빨간색으로 변경
	3	<ul style="list-style-type: none"> 캐러셀 화살표를 클릭시3개의 이미지가 순서대로 fade in형식으로 전환 캐러셀 화면이 자동으로 전환
	4	사용자가 로그인 할 시 아이디가 표시됨
	5	

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	메인페이지_하단부	페이지	1-2
화면경로	메인페이지				



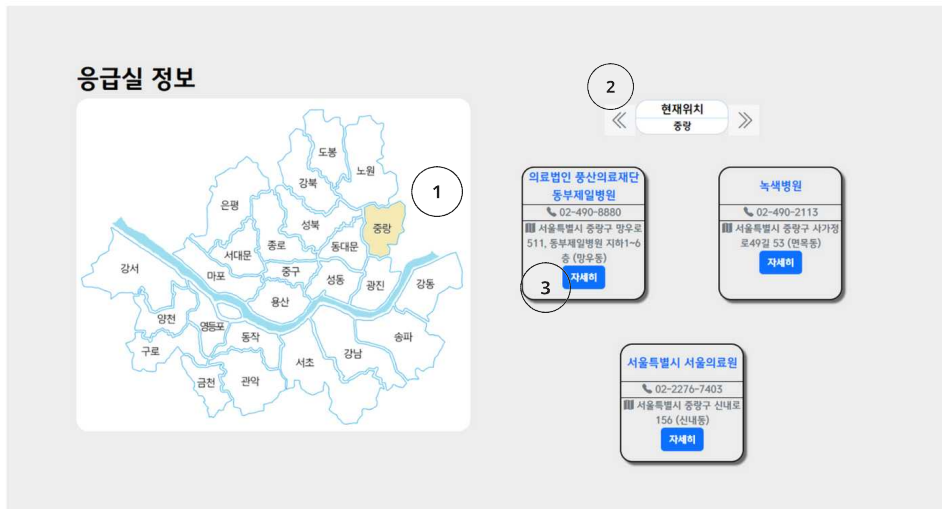
설명	
1	스크롤 내릴 시 해당 이미지 FADE-OUT
2	버튼에 마우스 오버 시 구분될 수 있도록 색상변경 버튼 클릭시 해당 페이지로 이동
3	
4	
5	
6	

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	응급실 현황 페이지_상단부	페이지	2-1
화면경로	메인페이지 > 응급실 현황 페이지				



응급실 현황 혼잡도 예측 인적자원 재배치 응급실 정보 로그인 계

설명	
1	지역 클릭 시 클릭한 지역의 색상이 노란 색으로 변환 되어야하며, 해당 지역에 대한 응급실 데이터가 POP-UP 될 수 있어야 함.
2	화살표 클릭 시, 랜덤하게 다른 지역으로 이동 현재 위치 하단 부분에 해당 지역에 대한 이름 변경
3	자세히 버튼 위에 마우스 오버 시 구분될 수 있도록 색상 변경
4	
5	



프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	응급실 현황 페이지	페이지	2-2
화면경로	메인페이지 > 응급실 현황 페이지				

실시간 응급실

병원 아이콘을 클릭하시면, 실시간 시작과 정보가 표시됩니다.



설명

1	현위치 버튼을 클릭하면 사용자의 현위치가 지도에 따라 한다.
2	처음으로 버튼을 클릭 시, 페이지의 상단부로 이동될 수 있어야 한다.
3	
4	
5	
6	

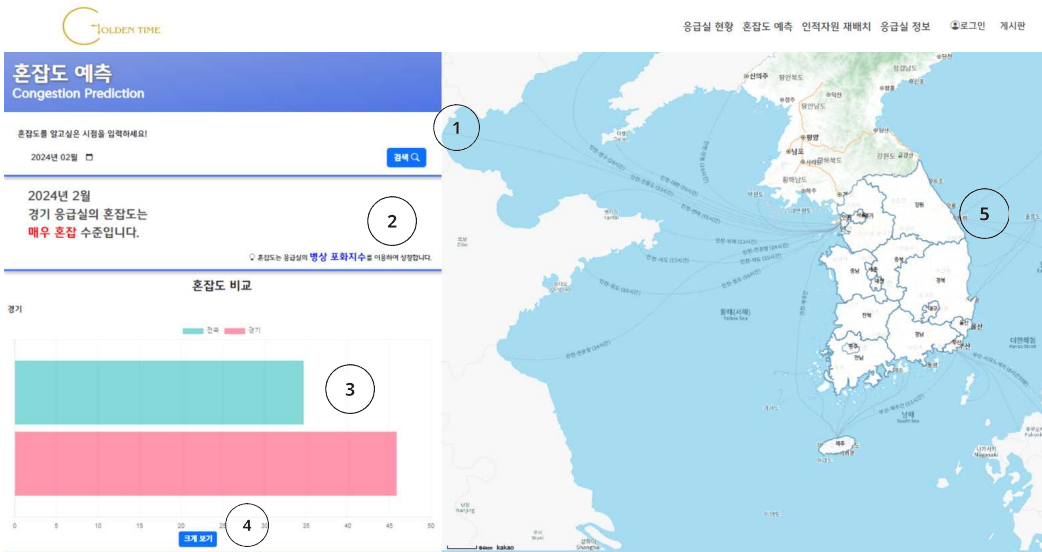
응급 메시지

제목:

응급/중증구분:

진료결과 메시지: 격리실 소독관련 전산반영체계 미구축으로 격리실 현황 실태와 다를 수 있음

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	혼잡도 예측 페이지	페이지	3
화면경로	메인페이지 > 혼잡도 예측 페이지				



설명	
1	<ul style="list-style-type: none"> 처음 페이지 로딩 시, 현재 년 월이 자동으로 입력됨 날짜를 입력하고 검색을 누를 시, 전국 기준 월기준 혼잡도가 예측
2	'병상포화지수' 클릭 시 병상 포화지수 계산식이 나타남
3	<p>월별 기준, 지역기준에 따라 차트가 바뀜</p> <p>월별 : 전국 혼잡도 데이터 예측 데이터</p> <p>지역별 : 전국과 해당 지역 데이터가 자동 비교됨</p>
4	<p>버튼 마우스 오버 시 구분할 수 있게 색상 변경</p> <p>버튼 클릭 시 해당 차트가 크게 보일 수 있도록 POP-UP</p>
5	지도 오버레이 위에 마우스 오버 시 파란색으로 색상 변경 클릭 시 혼잡도 예측 결과 출력

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	인적자원 재배치 페이지	페이지	4
화면경로	메인페이지 > 인적자원 재배치 페이지				

인적자원 재배치 분석 결과



서울 응급의학과 전문의 중 341명(34.3%)이 경기도로 이동하게 되면, 서울과 경기 모두 2PPH를 충족하게 되어 의료형평성이 개선될 수 있음을 알 수 있다.

클릭하여 계산식 확인하기

1

수요 FTE = Number of Patient / Capacity of 1 FTE

1. 서울 : 1,025,310명 / 3,528 = 291FTE

2. 경기 : 1,358,634명 / 3,528 = 385FTE

수요 전문의 수 = FTE / (전문의 중 응급실에 근무하는 비율 x 실제 근무시간)

1. 서울 : 291FTE / (0.8 x 0.7) = 519명

2. 경기 : 385FTE / (0.8 x 0.7) = 688명

FTE란? 응급의학과 전문의 1명을 의미하는 것이 아니라 전체 업무량의 기본단위이다.

설명

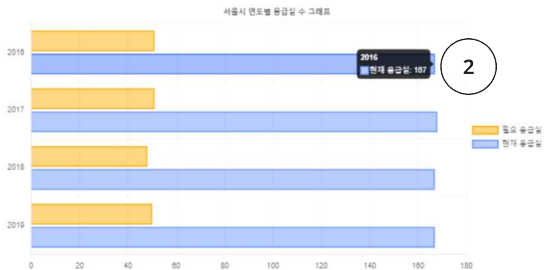
1	버튼을 누를 시 계산식이 나타남
2	진한 글씨 위에 마우스 오버 시 밑줄이 나타나는 애니메이션이 실행됨
3	차트 위에 마우스 오버 시 데이터가 나타남
4	
5	
6	

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	응급실 정보 페이지	페이지	5-1
화면경로	메인페이지 > 응급실 정보페이지				



응급실 현황 혼잡도 예측 인적자원 재배치 응급실 정보

1 서울시 응급실 수와 병상포화지수



이 그래프는 서울에 위치한 응급실의 수와 인구수 대비 필요한 응급실의 수를 표현한 그래프입니다.

버튼을 눌러 차트를 확인해보세요

1

응급실 병상포화지수

설명

1

차트 위에 마우스 오버 시 데이터가 나타남

2

버튼을 클릭하면 해당 데이터가 출력됨

3

4

5

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	응급실 정보 페이지	페이지	5-2
화면경로	메인페이지 > 응급실 정보페이지				



응급실 현황 혼잡도 예측 인적자원 재배치 응급실 정보 ④로

2 서울시 인구수 변화율



서울시 인구수 변화율을 시각화한 그래프입니다.

이를 통해 인구수가 점차 감소한다는 사실 알 수 있으며, 응급실을 늘리고자 할때 이를 염두해야 합니다.

설명	
1	차트 위에 마우스 오버 시 데이터가 나타남
2	버튼을 클릭하면 해당 데이터가 출력됨
3	
4	
5	

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	응급실 정보 페이지	페이지	5-3
화면경로	메인페이지 > 응급실 정보페이지				



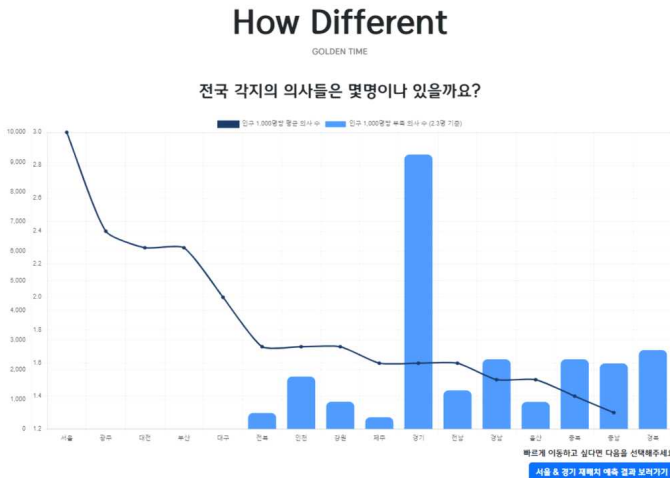
1



설명	
1	차트 위에 마우스 오버 시 차트 데이터 표시
2	
3	
4	
5	

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	응급실 정보 페이지	페이지	5-4
화면경로	메인페이지 > 응급실 정보페이지				

1



2

Information

주소: 서울 마포구 선릉로 176(대흥동 12-20)

설명	
1	차트 위에 마우스 오버 시 데이터가 나타남
2	버튼 클릭 시 인적자원 재배치 페이지로 이동 버튼 위에 마우스 오버 시 색상 변환
3	
4	
5	

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	로그인 페이지	페이지	6
화면경로	메인페이지 > 로그인 페이지				



응급실 현황 혼잡도 예측 인적자원 재배치 응급실 정보 **로그인**

로그인

1

아이디

비밀번호

로그인

회원가입

2

3

4



Information

📍 주소 : 서울 마포구 신촌로 176(대흥동 12-20)
 ☎ 대표전화 : 02-718-8513
 ✉ 이메일 : GoldenTime@gmail.com
 🕒 근무시간 : 월 ~ 금 / AM 9:00 ~ PM 6:00

설명	
1	아이디는 영어로 입력되어야 함
2	비밀번호는 가려져서 표시
3	로그인 버튼 위에 마우스 오버 시 색상 변경 버튼 클릭 시 로그인 가능/불가능 여부 판단 후 홈페이지로 이동
4	회원가입 버튼 클릭 시 회원가입 페이지로 이동
5	로그인 성공 시, '로그인' 대신 아이디로 변경

프로젝트명	응급실 혼잡도 분석을 통한 의로서비스 극대화	설명	회원가입 페이지	페이지	7
화면경로	메인페이지 > 회원가입 페이지				



응급실 현황 본업노 배속 입석사전 선택시 응급실 정보

회원가입

- 아이디를 입력해주세요.
- 비밀번호를 입력해주세요.
비밀번호를 한번 더 입력해주세요.
- 이름을 입력해주세요.
- 이메일을 입력해주세요.
- 핸드폰 번호를 입력해주세요.
- 주소를 검색해주세요. 주소 검색
- 회원가입
- 메인화면으로 돌아가기

설명	
1	아이디는 영어로 입력되어야 함
2	비밀번호는 가려져서 표시
3	이메일은 '0000@000.com' 형식으로 입력되어야 함
4	핸드폰 번호는 ' - ' 표시 없이 입력되어야 함
5	버튼 클릭 시 회원가입 가능/ 불가능 여부 판단
6	버튼 클릭 시 메인화면으로 이동

프로젝트명	응급실 혼잡도 분석을 통한 의료서비스 극대화	설명	게시판 페이지	페이지	8
화면경로	메인페이지 > 게시판 페이지				

게시판

번호	이름	제목	날짜	조회수
11	asdf	111	2024-01-22 00:00:00.0	15
4	123	test1	2024-01-26 10:30:16.0	3
5	이준수	- test1.1	2024-01-26 10:30:41.0	2
3	1234	1234	2024-01-26 09:50:51.0	2
2	123	test2	2024-01-26 09:48:00.0	6
1	123	1234	2024-01-26 09:23:26.0	10

1

1

작성하기

2

설명

1	작성하기 버튼 클릭 시 작성페이지로 이동
2	조회된 횟수만큼 카운팅되어서 표시
3	
4	
5	
6	



Information

📍 주소 : 서울 마포구 신촌로 176(대흥동 12-20)
 ☎ 대표전화 : 02-718-8513
 ✉ 이메일 : GoldenTime@gmail.com
 🕒 근무시간 : 월 ~ 금 / AM 9:00 ~ PM 6:00

프로젝트명	응급실 혼잡도 분석을 통한 의로서비스 극대화	설명	게시판 작성 페이지	페이지	9
화면경로	메인페이지 > 게시판 작성 페이지				

OPEN TIME

응급실 현황 혼잡도 예측 인적자원 재배치 응급실 정보

게시판

1

작성자

손하지

2

제목

내용

3

작성

4

취소

설명

1	작성자의 이름은 기본 정보에 등록되어 있는 이름으로 자동으로 표시됨
2	제목과 내용이 반듯이 들어가야 함
3	작성 버튼 클릭 시 해당 내용이 업로드 마우스 오버 시 구분될 수 있도록 색상 변경
4	취소 버튼 클릭 시 이전으로 돌아가감 마우스 오버 시 구분될 수 있도록 색상 변경
5	
6	

[Controller]

@Controller

```
public class MainController {
```

@Autowired

```
private IService iService;
```

@Autowired

: 요청 처리를 위해 필요한 인터페이스를 활용해 의존성을 주입

@RequestMapping("/congestion")

```
public String congeston(Model model) {
```

Model 객체를 활용해 데이터 처리

```
    System.out.println("월선택이 없다.");
```

```
    LocalDate currentDate = LocalDate.now();
```

```
    Integer serverYear = currentDate.getYear();
```

```
    Integer serverMonth = currentDate.getMonthValue(); // month 파라미터를 년도와 월로 분리
```

```
    String stringDate = String.format("%04d-%02d", serverYear, serverMonth); // 현재 년도,월 "yyyy-mm" 형식으로 변환
```

```
    model.addAttribute("stringDate", stringDate); //stringDate값을 model을 사용해서 View단으로 전송
```

```
    model.addAttribute("serverYear", serverYear); //serverYear 값을 model을 사용해서 View단으로 전송
```

```
    model.addAttribute("serverMonth", serverMonth); //serverMonth 값을 model을 사용해서 View단으로 전송
```

```
    return "goldenTime/congestion";
```

view의 이름을 지정하여 데이터 전송

```
}
```

@GetMapping("/monthSearch") //form태그로 가져온 데이터 처리하는 컨트롤러

```
public String handleMonthSearchRequest(@RequestParam("month") String month, Model model) {
```

@GetMapping

: 요청 url을 'GET' 방식으로 mapping하여 필요한 서비스 매칭

```
    System.out.println("month 메서드 실행");
```

```
    model.addAttribute("stringDate", month); // 월을 StringDate라는 변수에 저장
```

```
    iService.executeMonth(model, month); // executeMonth 메서드 실행
```

```
    return "goldenTime/congestion";
```

```
}
```

@GetMapping("/region") // Fetch Api로 데이터 받아와서 처리 후 재 전송

```
public @ResponseBody List< Object> getRegressionJSONList(@RequestParam("region") String region, @RequestParam("month")
```

```
String month, Model model) {
```

```
    return iService.executeRegion(region, month); // executeRegion 메서드 실행 후 json형식으로 return
```

```
}
```

```
}
```

[Service]

@Service

```
public class SSearchService implements ISService {
```

@Autowired

@Qualifier("ISRepository")

```
private ISRepository iSRepository;
```

@Qualifier

: 요청 처리를 위해 한정자 사용

```
public void executeMonth(Model model, String monthAndYear) {  
    int year = Integer.parseInt(monthAndYear.substring(0, 4)); // 'yyyy-mm' 형식에서 년 월 나누기  
    int month = Integer.parseInt(monthAndYear.substring(5));  
  
    RegressionVO regression = iSRepository.getRegressionInfo("nationwide"); // 전국 표준 데이터 가져오기  
    DataVO queryData = iSRepository.getDataInfo("nationwide", year, month);  
  
    float congestionValue = calculateCongestionValue(regression, queryData, year, month); // 회귀식 계산 결과값  
    String congestion = calculateCongestionLevel(congestionValue); // 회귀식 계산 결과 값 단계별로 나누기  
  
    model.addAttribute("congestion", congestion); // model에 데이터 담아서 전송  
    model.addAttribute("congestionValue", congestionValue);  
    model.addAttribute("region", "전국");  
    model.addAttribute("serverYear", year);  
    model.addAttribute("serverMonth", month);  
}
```

Model 객체를 활용해 데이터 처리

```
...  
private float calculateCongestionValue(RegressionVO regression, DataVO data, int year, int month) { // 회귀식 계산 메서드
```

```
    float RConstant = regression.getConstant(); // 상수  
    float RYear = regression.getYear(); // 년도 상수  
    float RMonth = regression.getMonth(); // 월 상수  
    float RPatent_119 = regression.getPatent_119(); // 이송건수 상수  
    float RPatent_ER = regression.getPatent_ER(); // 응급실 이용건수 상수  
    float RTA = regression.getTA(); // 교통사고 상수
```

```
    int patient_119 = data.getPatent_119(); // 이송 건수  
    int patient_ER = data.getPatent_ER(); // 응급실 건수  
    int TA = data.getTA(); // 교통사고 건수
```

```
    return RConstant + RYear * year + RMonth * month + RPatent_119 * patient_119  
        + RPatent_ER * patient_ER + RTA * TA;
```

```
}}
```

[Repository]

```
public interface ISRepository {  
    List<RegressionVO> getRegressionList(@Param("region")String region); //지역 기준 서비스 메서드  
  
    List<DataVO> getDataList(@Param("year") int year, @Param("month") int month); // 날짜 기준 서비스 메서드  
  
    RegressionVO getRegressionInfo(@Param("region") String region);  
    // 테이블 이름을 동적으로 변경하고 year를 고정하는 메서드  
    DataVO getDataInfo(@Param("region")String region,@Param("year") int year,@Param("month") int month);  
}
```

@Param

: 변수 이름을 명시해서 데이터 가져오기

[Mybatis]

```
<mapper namespace="com.mycompany.myapp.dao.ISRepository">
```

```
<!-- RegressionVO에 대한 resultMap 정의 -->
```

```
<resultMap id="getRegressionInfo"
```

```
  type="com.mycompany.myapp.model.RegressionVO">
```

```
  <result property="region" column="REGION" />
```

```
  <result property="constant" column="CONSTANT" />
```

```
  <result property="year" column="YEAR" />
```

```
  <result property="month" column="MONTH" />
```

```
  <result property="patient_119" column="PATIENT_119" />
```

```
  <result property="patient_ER" column="PATIENT_ER" />
```

```
  <result property="TA" column="TA" />
```

```
</resultMap>
```

```
<!-- RegressionVO 리스트를 조회하는 쿼리 -->
```

```
<select id="getRegressionList" resultMap="getRegressionInfo">
```

```
  SELECT *
```

```
  FROM REGRESSION
```

```
  WHERE REGION = #{region}
```

```
</select>
```

```
<!-- 특정 지역의 Regression 정보를 조회하는 쿼리 -->
```

```
<select id="getRegressionInfo" parameterType="string"
```

```
  resultType="com.mycompany.myapp.model.RegressionVO">
```

```
  SELECT *
```

```
  FROM REGRESSION
```

```
  WHERE REGION = #{region}
```

```
</select>
```

```
...
```

← mapper namespace

: xml의 설정과 인터페이스에 선언한 설정을 같이 사용함. 제약도 같이 적용

← Query 문을 입력해서 DB 정보 가져오기

[Model]

```
public class DataVO {  
    private int year; // 연도  
    private int month; // 월  
    private int patient_119; // 구급차 이송 건수  
    private int patient_ER; // 응급실 이송 건수  
    private int TA; // 교통사고 건수  
  
    public int getYear() {  
        return year;  
    }  
    public void setYear(int year) {  
        this.year = year;  
    }  
    public int getMonth() {  
        return month;  
    }  
    public void setMonth(int month) {  
        this.month = month;  
    }  
    public int getPatient_119() {  
        return patient_119;  
    }  
    public void setPatient_119(int patient_119) {  
        this.patient_119 = patient_119;  
    }  
    public int getPatient_ER() {  
        return patient_ER;  
    }  
    public void setPatient_ER(int patient_ER) {  
        this.patient_ER = patient_ER;  
    }  
    public int getTA() {  
        return TA;  
    }  
    public void setTA(int tA) {  
        TA = tA;  
    }  
}
```



private 변수에 대한 getter / setter 지정

[R]

의사결정 트리 그림

library(rpart.plot)

시각화 Rplot01

prp(emr_dtree, type = 5, extra = 104, roundint = FALSE,
main = "Decision Tree for Emergency Room Congestion Prediction", font=4)

의사결정 트리 모델로 예측

emr_dtree_pred <- predict(emr_dtree, newdata = emr, type = "class")

실제 결과와 예측 결과 비교

table(emr\$name, emr_dtree_pred, dnn = c("Actual", "Predicted"))

정확도 계산

accuracy <- mean(emr\$name == emr_dtree_pred)

cat("Prediction Accuracy:", accuracy, "\n")

rpart 함수 사용

emr_model <- rpart(name ~ distance + congestion + accident, data = emr)

예측

predict(emr_dtree, newdata = emr, type = "class")

head(predict(emr_dtree, newdata = emr, type = "class"))

tail(predict(emr_dtree, newdata = emr, type = "class"))

...

교차타당성 오차를 최소화하는 Tree로 가지치기(pruning)

opt <- which.min(emr_dtree\$cptable[, "xerror"])

cp <- emr_dtree\$cptable[opt, "CP"]

cp

prune.emr_dtree <- prune(emr_dtree, cp = 0.05)

plotcp(emr_dtree)

summary(emr_dtree)

summary(prune.emr_dtree)

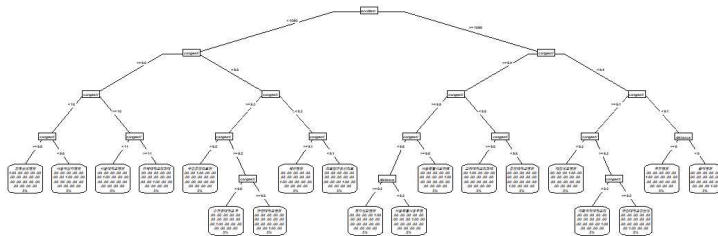
xerror 값 표준편차 구하기 -> 1.1

printcp(prune.emr_dtree)

plotcp(prune.emr_dtree)

[의사결정 트리 시각화 자료]

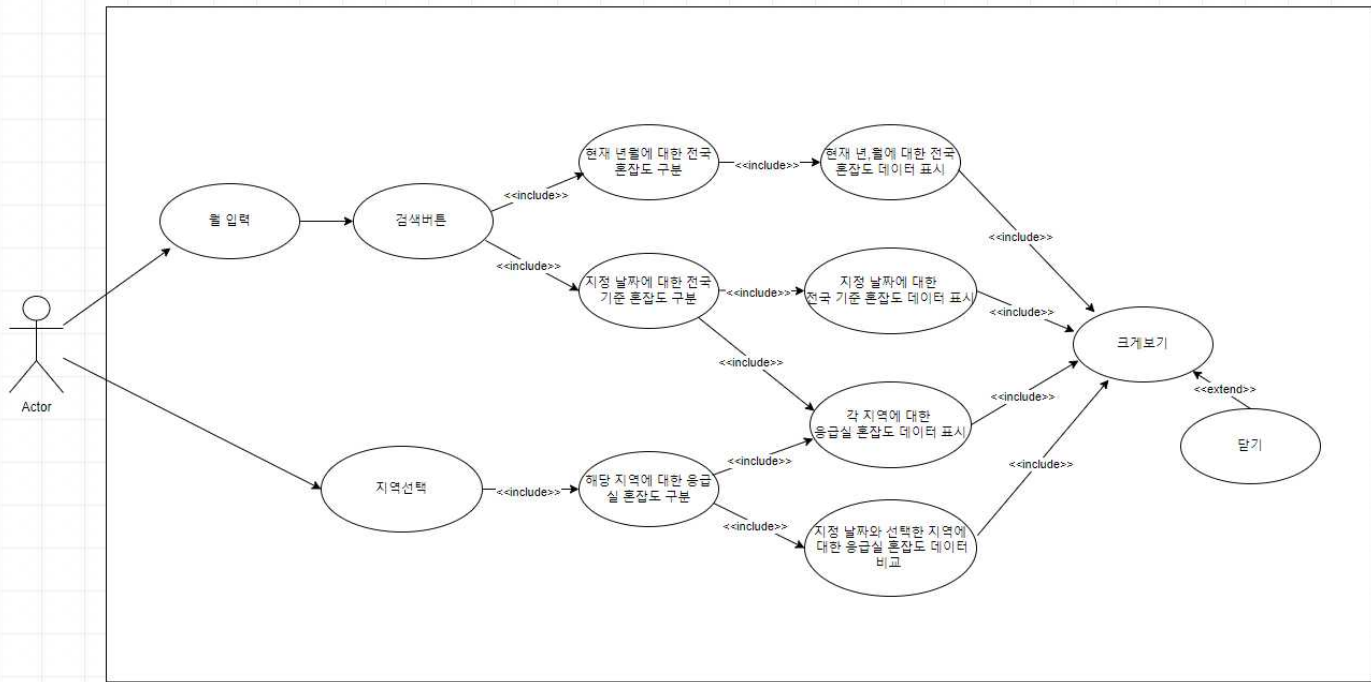
Decision Tree for Emergency Room Congestion Prediction



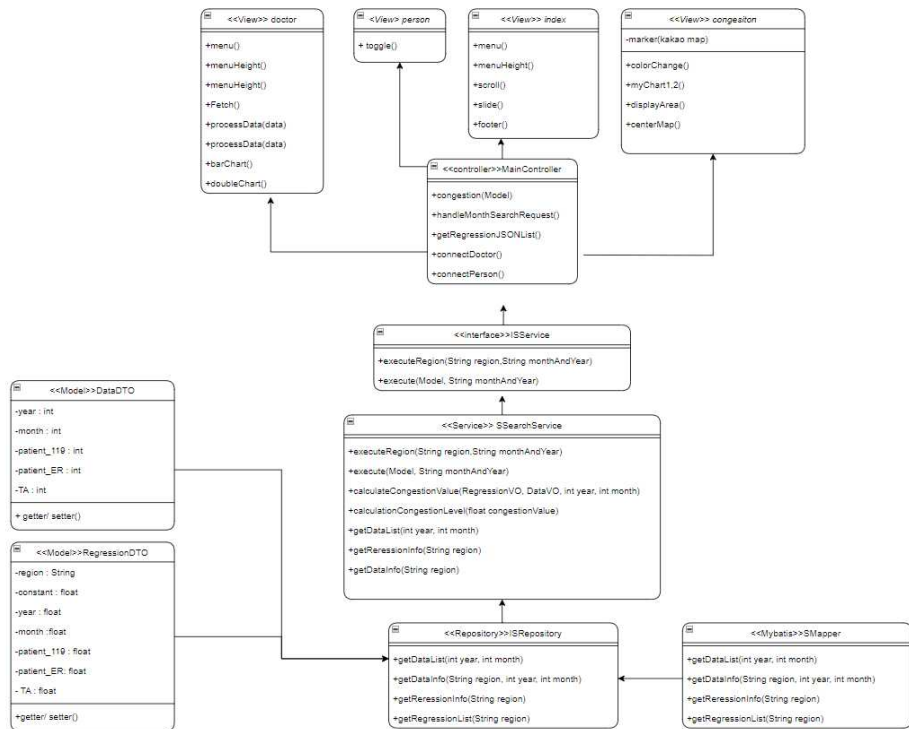
```
> # 예측
> predict(emr_dtree, newdata = emr, type = "class")
연세대학교의과대학세브란스병원      세란병원      서울직업자병원      3
      갑곡삼성병원      의료법인동신의료재단 동신병원      5      서울대학교병원      6
      국립중앙의료원 이화여자대학교의과대학부속목동병원      8      중앙대학교병원      9
      순천향대학교 부속 서울병원      서울특별시보라매병원      11      부민병원      12
      명지성모병원      한림대학교강남성신병원      14      부민병원      15
      명지성모병원      대림성모병원      17      한양대학교병원      18
      고려대학교의과대학부속구로병원      19
19 Levels: 갑곡삼성병원 고려대학교의과대학부속구로병원 국립중앙의료원 대림성모병원 명지성모병원 부민병원 ... 을지병원
```

예측 정확도 결과 : 89.47%

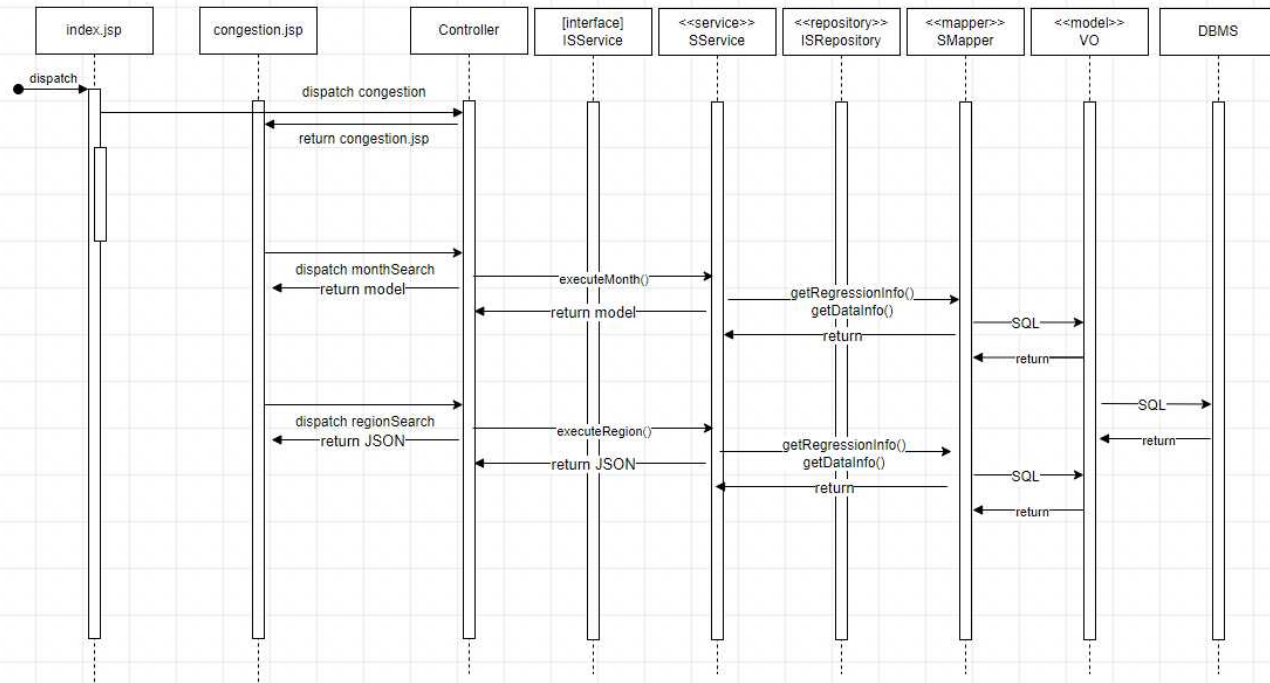
Usecase Diagram



Class Diagram



Sequence Diagram



ER Diagram

GOLDTIME.BUSAN	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.CHUNGBUK	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.CHUNGNAM	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.DAEGU	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.DAEJEON	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.GANGWON	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.GWANGJU	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.GYEONGBUK	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.GYEONGGI	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.GYEONGNAM	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.JEJU	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.JEONBUK	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.JEONNAM	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.NATIONWIDE	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.ULSAN	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

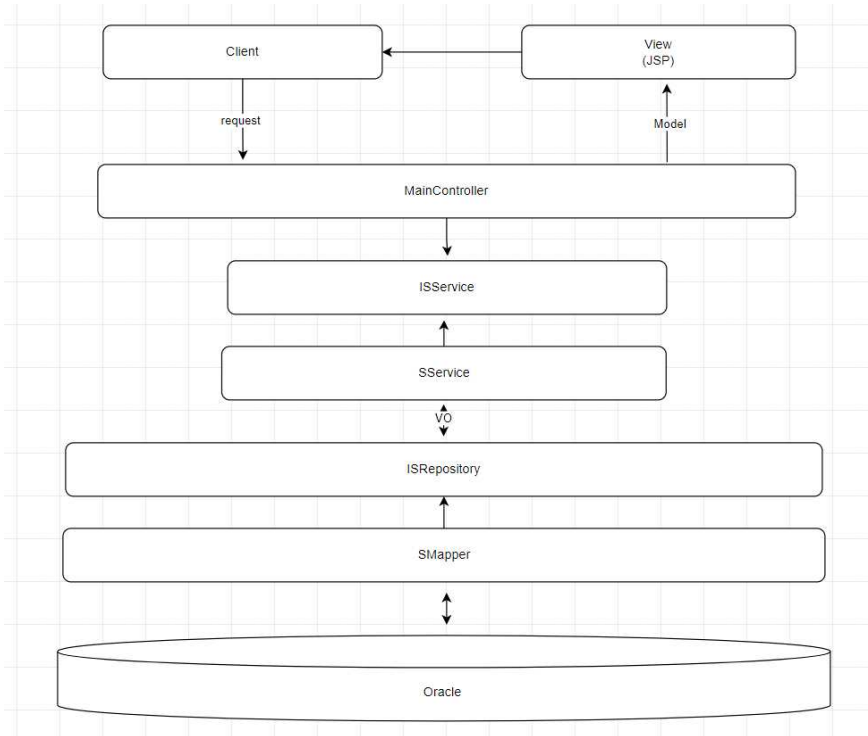
GOLDTIME.SEJONG	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.SEOUL	
YEAR	NUMBER (*,0)
MONTH	NUMBER (*,0)
PATIENT_119	NUMBER (*,0)
PATIENT_ER	NUMBER (*,0)
TA	NUMBER (*,0)

GOLDTIME.REGRESSION	
REGION	VARCHAR2 (20 BYTE)
CONSTANT	NUMBER
YEAR	NUMBER
MONTH	NUMBER
PATIENT_119	NUMBER
PATIENT_ER	NUMBER
TA	NUMBER

GOLDTIME.OECD	
CONTRIES	VARCHAR2 (20 BYTE)
RESULT	NUMBER (38,1)

Software Architecture



향후 개발 계획

- 현재 JSP로 되어있는 View를 React로 변환
 - DOM을 사용하여 UI와 렌더링 성능 향상
 - 단일 페이지 어플리케이션(SPA) 구현
- 현재 JS를 Node.js로 변환
 - 비동기 이벤트 기반 아키텍처를 사용하여 높은 성능과 개발 생산성 향상

프로젝트 수행소감

Spring 구동 방식의 이해 과정

반복되는 404에러에 스프링 구동 방식에 대한 이해가 필요했습니다.

구조에 대한 이해가 부족하다보니 오류가 난 부분을 정확하게 잡을 수 없었고, 오류를 잡지 못한 상황에서 다시 처음으로 돌아가서 서버가 구동되는 시점부터 어느 파일을 지나 어떻게 동작하는 지 확인하였습니다. 그 과정에서 web.xml, servlet-context.xml, View단과 Controller의 역할에 대해 이해하게 되었고, 페이지를 완성할 수 있었습니다.

이로 인해, 구동 방식의 흐름에 대해 이해할 수 있었고 개발에 대한 흥미가 높아짐을 느낄 수 있었습니다.

첫 소프트웨어 프로젝트에 대한 흥미

이번 프로젝트를 통해 소프트웨어 개발에 경험을 쌓고, 첫 발을 내딛었다는 점에서 가치가 있다고 생각합니다.

다양한 의견들이 오고 가면서 의견을 나누고, 다시 한번 생각하고 이해함을 통해 자바와 스프링에 대한 기본적인 개념을 더 탄탄하게 쌓게 되었습니다.

팀으로 진행되는 프로젝트이기 때문에 부담감이 있었지만 팀원들과 함께 서비스를 구현하면서 서로의 피드백을 통해 보다 더 효율적인 코드로 리팩토링할 수 있었습니다. 이를 통해, 보다 확장된 생각을 가지게 될 수 있었고, 팀프로젝트에 대한 중요성을 깨닫는 계기가 되었습니다.

감사합니다

Telephone

010-7987-3197

Github

<https://github.com/doHand/portfolio>

Project

<http://43.203.78.27:8080/>