

AI/ML 모델 리포트 관리

고급 웹 프로그래밍 기말고사 프로젝트 계획서
소프트웨어전공 - 202020881 - 나성곤

목차

1. 개요
2. Framework 구성
3. Project 구조
4. UI/UX 정의서
5. Database 설계
6. 주요 class 역할 정의서
7. Sequence Diagram
8. Test 방법

개요

- **목적**

- AI 모델, ML 모델을 테스트한 정량적 자료를 사용자가 리포트에 기입을 하여 관리자가 관리
- 모델의 정량적 평가를 가지고 성능을 비교하여 더 좋은 모델을 한눈에 확인하는 것이 목적입니다.
- 웹 사이트 이용자는 User(사용자), Admin(관리자) 가 있습니다.
- 관리자는 사용자를 관리합니다.
- 관리자는 모델을 관리합니다.
- 사용자는 리포트를 작성하고, 관리자는 리포트를 관리합니다.

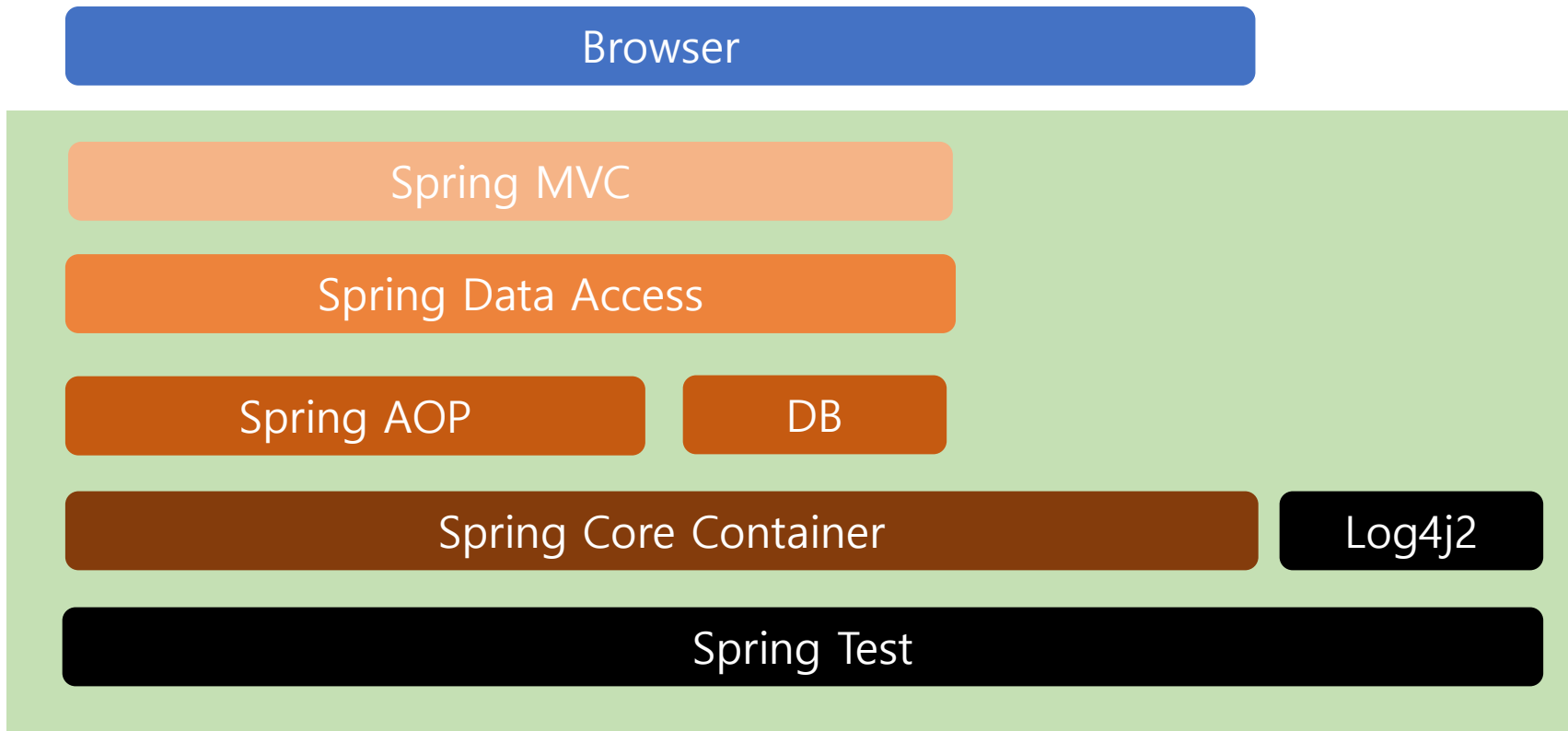
- **상세 정보**

- 사용자는 생성(회원가입), 조회, 수정, 삭제(소프트)가 가능합니다.
 - (소프트) : 데이터 상태 변경
- 사용자는 리포트를 생성 할 수 있으며 조회,수정,삭제(소프트)가 가능합니다.
- 사용자는 모델 리스트를 조회할 수 있으며, 해당 모델의 리포트를 생성합니다.
- 관리자는 관리자에 대해 생성, 조회, 수정, 삭제가 불가능 합니다.
- 관리자는 사용자를 생성, 조회, 삭제(하드)가 가능합니다.
- 관리자는 모델을 생성, 조회, 삭제(하드)가 가능합니다.
 - (하드) : 데이터 삭제
- 관리자는 리포트의 조회, 삭제(하드)가 가능합니다.

Framework 구성

- Spring MVC
- Spring Test
- Spring Data Access
- Spring AOP
- 예외처리
- Log4j 2

Framework Stack Architecture



Project 구조 Java Package 구성도

base package - kr.ac.kku.cs.wp.wsd.nsg716

기능구분				
Level 1 (기능 구분)	Level 2 (mvc)	Level 3	Level 4	설명
기능 구분 package				
user				사용자 정보 관리
	controller			
	dao			
	dto			
	entity			Model,report entity 포함
Level 1 (기능 구분)	Level 2 (mvc)	Level 3	Level 4	설명
model				모델 정보 관리
	controller			
	dao			
	dto			
Level 1 (기능 구분)	Level 2 (mvc)	Level 3	Level 4	설명
report				리포트 정보 관리
	controller			
	dao			
	dto			

공통 기능				
Level 1 (기능 구분)	Level 2 (mvc)	Level 3	Level 4	설명
공통 package				
support				library 기능 확장 및 정의
	sql			
	aop			
tools				Utility
	message			
	secure			
	tags.			Usercard
		custom		
aaa				로그인 정보 Controller
	controller			
web				기본 뷰 Controller
	controller			
Level 1 (기능 구분)	Level 2 (mvc)	Level 3	Level 4	설명
공통 package				
				연결 테스트(기본)
resources				정적 자원 매핑

Project 구조

Web Directory 구성도

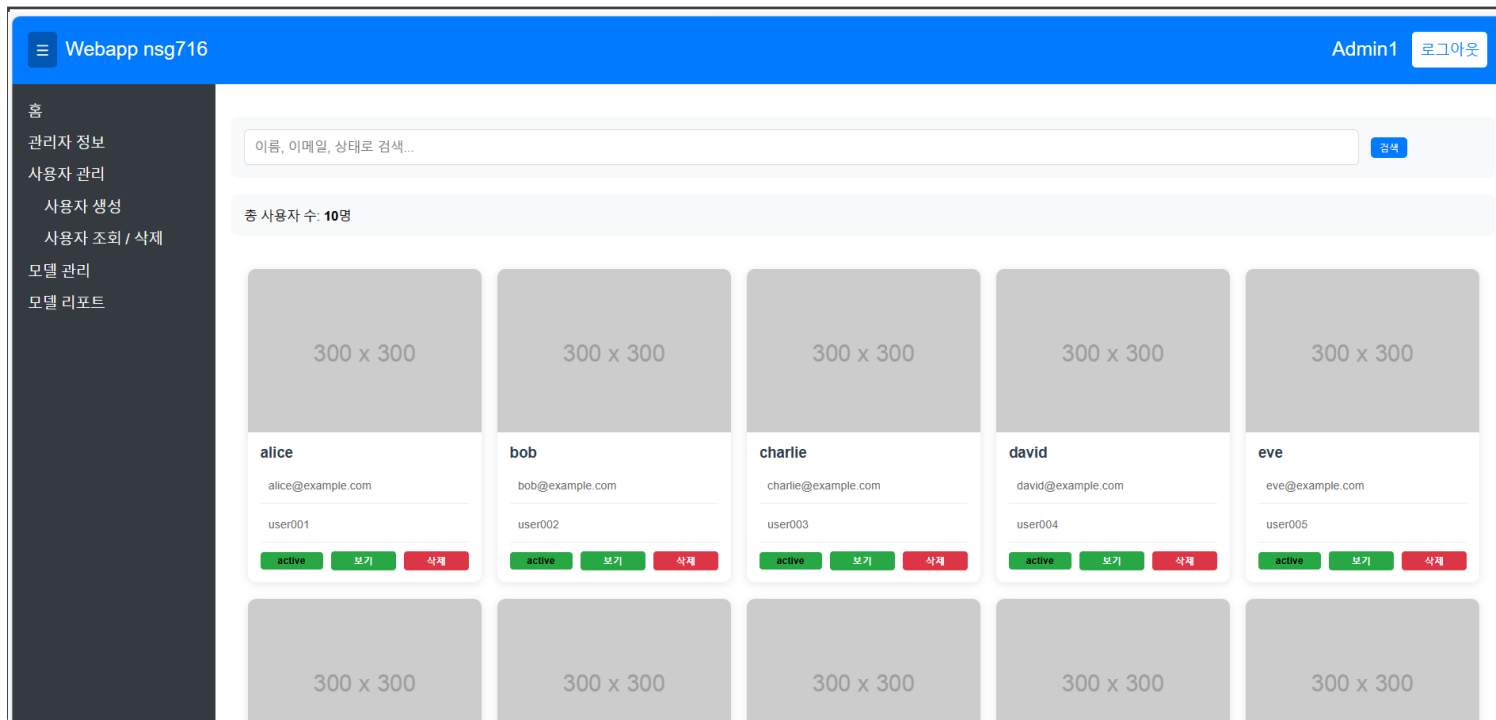
Level 1	Level 2	Level 3	Level 4	설명	Level 1	Level 2	Level 3	Level 4	설명
webapps					src/main/resources				
	WEB-INF			web application 설정, 라이브러리, classes 등이 있는 보안 디렉토리		static			스프링MVC 정적 자원 파일
		tag/template					image		
		tlds					js		
		view		View 구성					
			admin						
			auth						
			error						
			model						
			report						
			User						

UI/UX 정의서

(관리자) 사용자 목록 조회 화면 기능 정의

- (관리자) 사용자 목록 조회
 - 조회조건을 입력하고 정보 필터링
- (관리자) 사용자 상태에 따른 색상 변화
 - active : 초록색, inactive : 빨간색
- (관리자) 보기를 클릭하면 사용자 정보가 뜬다.
- (관리자) 삭제 클릭 시
 - 사용자 데이터가 삭제 (데이터 삭제)

(관리자) 사용자 목록 조회 화면

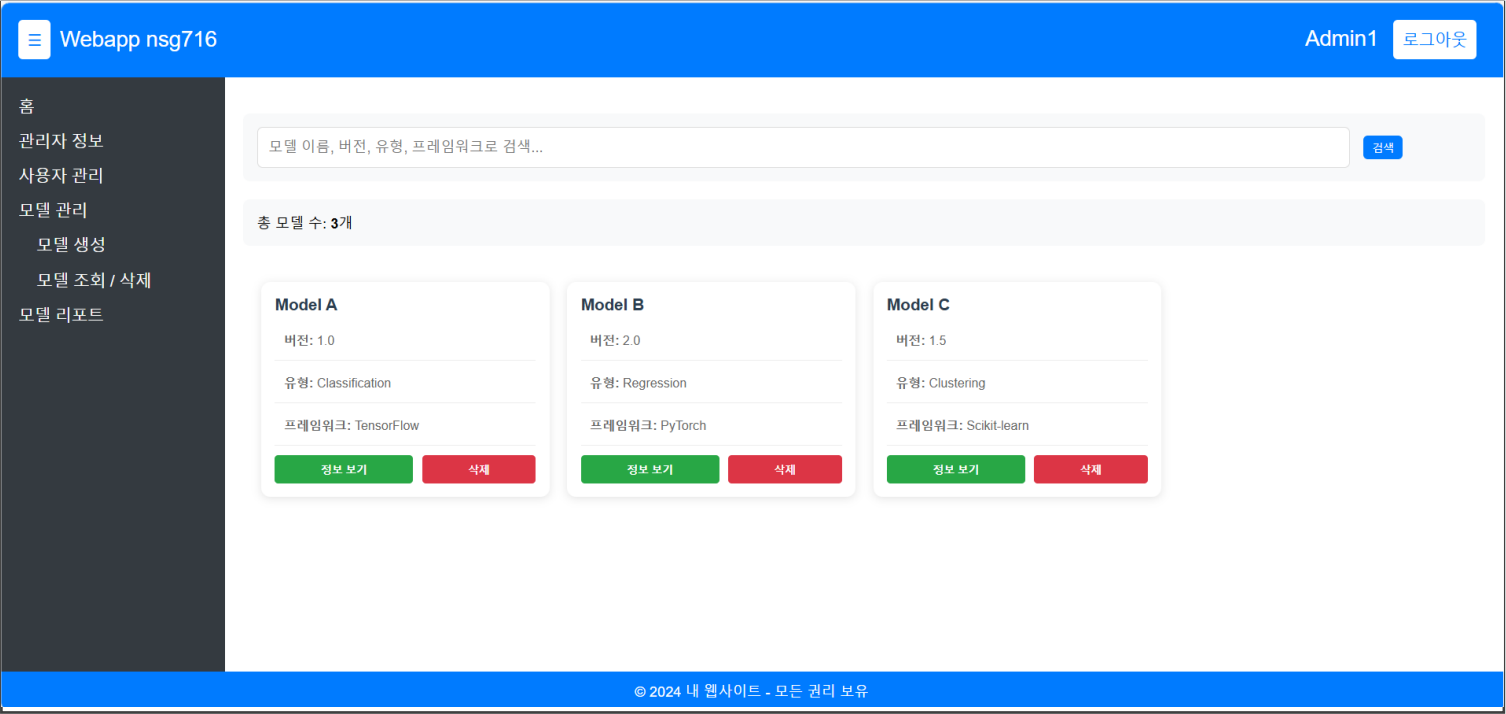


UI/UX 정의서

(관리자) 모델 목록 조회

- (관리자) 모델 목록 조회
 - 관리자 계정으로 로그인을 하면
 - 모델관리
 - 모델생성
 - 모델 조회/삭제
- (관리자) 모델 목록에서 조건을 입력
 - 이름, 버전, 유형, 프레임워크로 검색
- (관리자) 모델 목록 – 정보 보기
 - 더 자세한 정보를 확인
- (관리자) 모델 목록 – 삭제
 - 데이터 하드 삭제 가능

(관리자) 모델 목록 조회



UI/UX 정의서

(관리자) 모델 생성

- (관리자) 모델 생성
 - 모델 ID, 모델 이름, 버전, 모델 타입, 프레임워크, 설명 입력
 - 모델 ID는 기존에 입력한 것은 입력이 불가능하도록 설정 – 모델 생성시 확인 가능

(관리자) 모델 생성

The image shows a UI mockup of a form titled "모델 데이터 생성" (Model Data Generation) for administrators. The form is centered on a light gray background. It contains several input fields: "모델 ID" (Model ID), "모델 이름" (Model Name), "버전" (Version), "모델 타입" (Model Type), "프레임워크" (Framework), and "설명" (Description). Each field is a white rectangle with a thin gray border. The "설명" field is a larger text area. At the bottom of the form is a blue button with the text "생성하기" (Generate). The form is titled "모델 데이터 생성" in bold black text.

모델 데이터 생성

모델 ID

모델 이름

버전

모델 타입

프레임워크

설명

생성하기

UI/UX 정의서

(관리자) 모델 목록 상세 조회

- (관리자) 모델 목록 상세 조회
 - 모델 조회/삭제 – 정보보기 클릭
 - 데이터에 있는 모델에 대한 정보를 확인
 - 모델ID
 - 관리자 ID
 - 모델이름
 - 버전
 - 모델 타입
 - 프레임워크
 - 설명
 - 생성일
 - 수정일
- (관리자) 정보 수정 클릭
 - 해당 모델의 정보를 수정가능

(관리자) 모델 목록 상세 조회

Webapp nsg716Admin1로그아웃

홈관리자 정보사용자 관리모델 관리모델 생성모델 조회 / 삭제모델 리포트

모델 정보

모델 ID: model001

관리자 ID: admin001

모델 이름: Model A

버전: 1.0

모델 타입: Classification

프레임워크: TensorFlow

설명: Classification model for images

생성일: 2024-12-06T15:58:42

수정일: 2024-12-06T15:58:42

정보 수정

UI/UX 정의서

(관리자) 모델 정보 수정

- (관리자) 모델 정보 수정
 - 모델 조회/삭제 - 정보 보기 - 모델 수정
 - 데이터 수정 가능
 - 수정 가능 데이터 목록
 - 모델이름
 - 버전
 - 모델 타입
 - 프레임워크
 - 설명
 - 수정일의 경우 수정 성공 시 자동으로 수정
- (관리자) 수정 가능한 데이터 목록
 - 데이터를 공백으로 두면 에러 메시지 발생
- (관리자) 수정 완료 클릭
 - 수정 성공 시 홈 화면 이동
 - 수정 실패 시 에러메시지와 함께 다시 뜬다.

(관리자) 모델 정보 수정

Webapp nsg716 Admin1 로그아웃

홈
관리자 정보
사용자 관리
모델 관리
 모델 생성
 모델 조회 / 삭제
모델 리포트

모델 정보 수정

모델 이름
Model A

버전
1.0

모델 타입
Classification

프레임워크
TensorFlow

설명
Classification model for images

수정 완료

모델 정보 수정

모델 이름
Model A

버전

모델 타입
Classification

프레임워크
TensorFlow

설명
Classification model for images

수정 완료

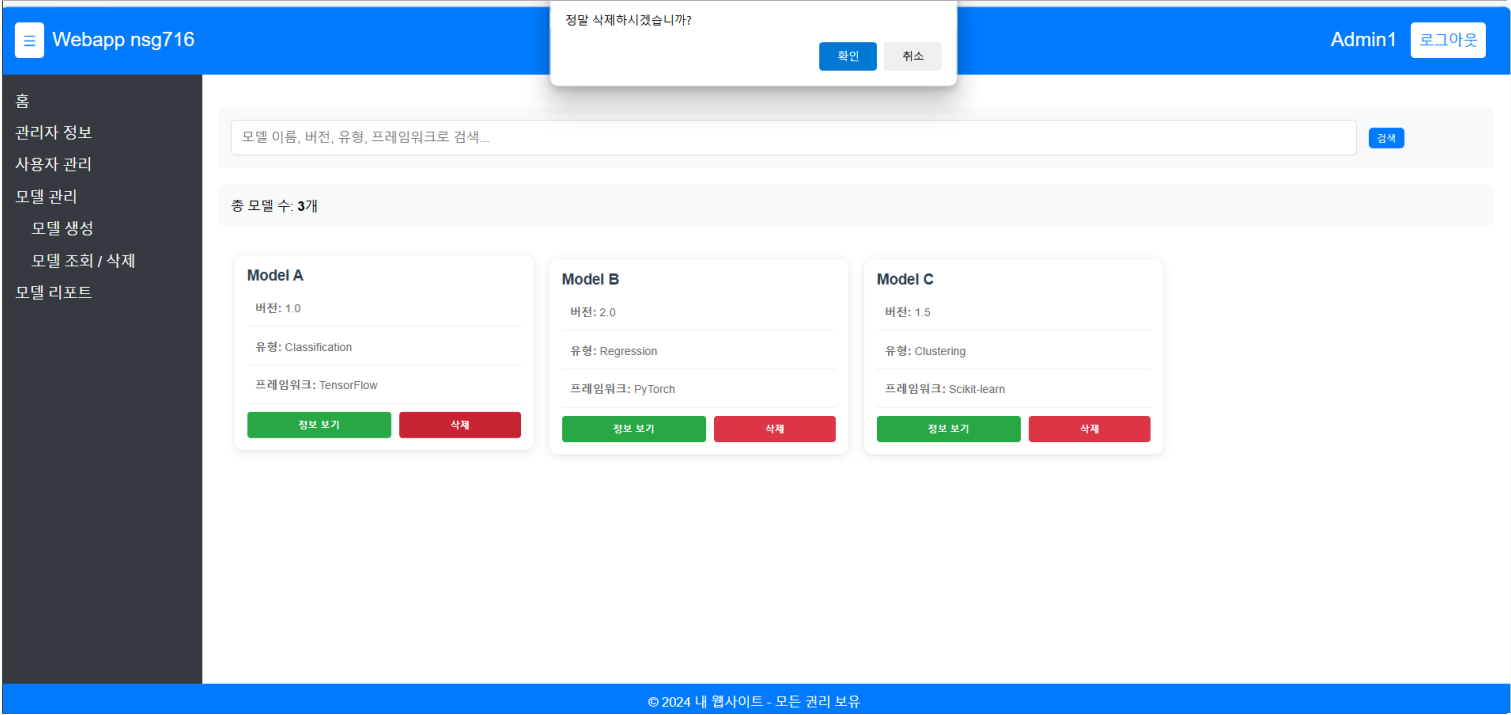
이 입력란을 작성하세요.

UI/UX 정의서

(관리자) 모델 삭제

- (관리자) 모델 삭제
 - 모델 삭제 – 하드 삭제 (데이터 삭제)
 - 선택한 모델에서 삭제 클릭 시 알림
 - “정말 삭제하시겠습니까?” 메시지 확인 시 삭제
 - 삭제 이후 모델 정렬 실행 삭제된 모델은 지워진 모습

(관리자) 모델 삭제



UI/UX 정의서

(관리자) 모델 삭제

(관리자) 모델 삭제 메시지 확인시

localhost:8080의 메시지

모델이 삭제되었습니다.

확인

(관리자) 모델 삭제 이후

모델 이름, 버전, 유형, 프레임워크로 검색...

총 모델 수: 2개

Model B

버전: 2.0

유형: Regression

프레임워크: PyTorch

정보 보기

삭제

Model C

버전: 1.5

유형: Clustering

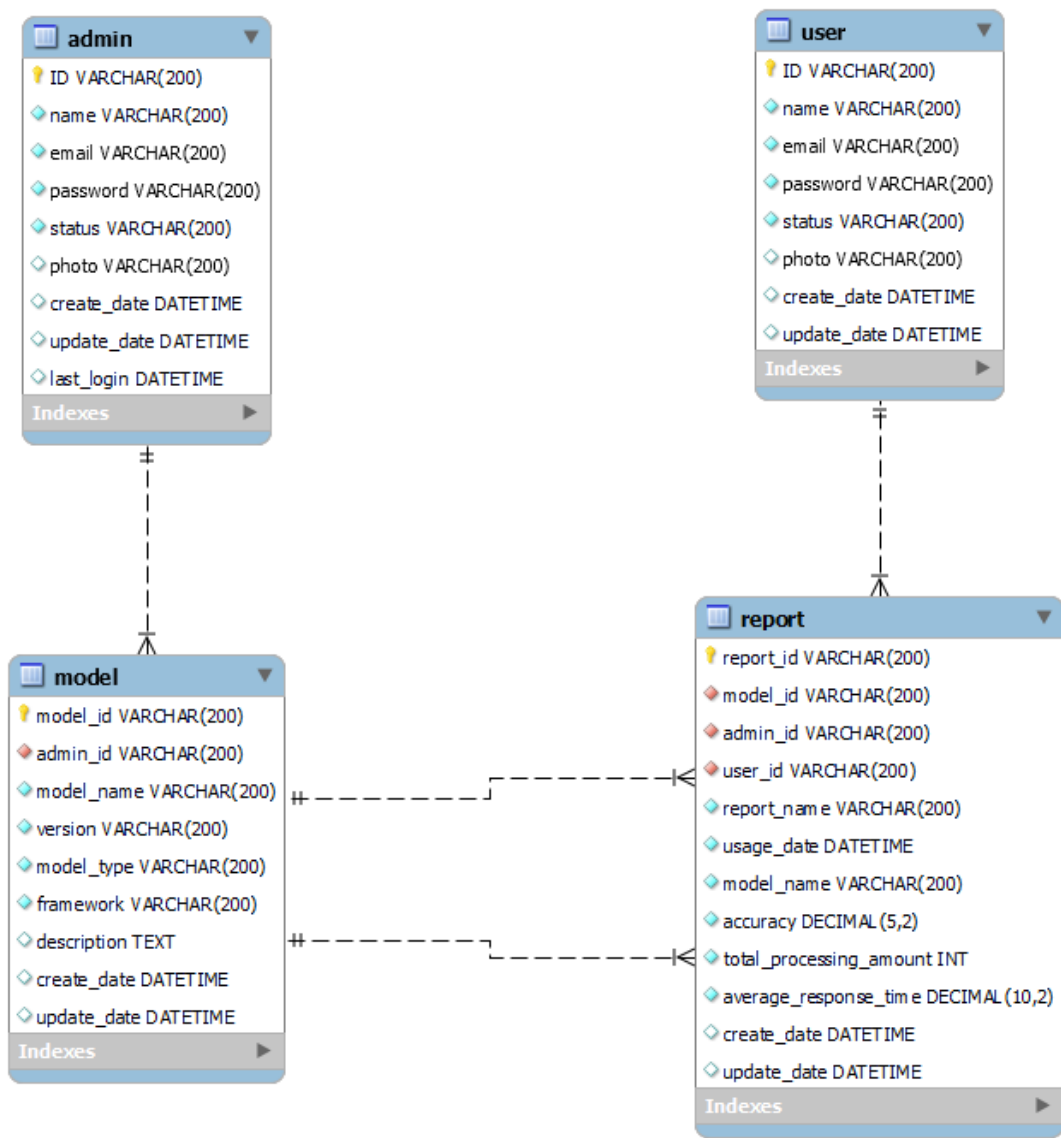
프레임워크: Scikit-learn

정보 보기

삭제

Database 설계

Table 관계도 (ERD)



Database 설계

Table 정의서

사용자 테이블 (User)			
컬럼 이름	데이터 유형	제약 조건	설명
ID	VARCHAR(200)	NOT NULL, PRIMARY KEY	사용자 고유 ID
name	VARCHAR(200)	NOT NULL	사용자 이름
email	VARCHAR(200)	NOT NULL	사용자 이메일
password	VARCHAR(200)	NOT NULL	사용자 비밀번호
status	VARCHAR(200)	NOT NULL	사용자 상태
photo	VARCHAR(200)	DEFAULT NULL	사용자 사진 URL
create_date	DATETIME	DEFAULT CURRENT_TIMESTAMP	생성일
update_date	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	수정일

관리자 테이블 (Admin)			
컬럼 이름	데이터 유형	제약 조건	설명
ID	VARCHAR(200)	NOT NULL, PRIMARY KEY	관리자 고유 ID
name	VARCHAR(200)	NOT NULL	관리자 이름
email	VARCHAR(200)	NOT NULL	관리자 이메일
password	VARCHAR(200)	NOT NULL	관리자 비밀번호
status	VARCHAR(200)	NOT NULL	관리자 상태
photo	VARCHAR(200)	DEFAULT NULL	관리자 사진 URL
create_date	DATETIME	DEFAULT CURRENT_TIMESTAMP	생성일
update_date	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	수정일
last_login	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	마지막 로그인

모델 테이블 (Model)			
컬럼 이름	데이터 유형	제약 조건	설명
model_id	VARCHAR(200)	NOT NULL, PRIMARY KEY	모델 고유 ID
admin_id	VARCHAR(200)	NOT NULL, FOREIGN KEY (Admin.ID) (ON UPDATE CASCADE ON DELETE CASCADE)	관리자 ID
model_name	VARCHAR(200)	NOT NULL	모델 이름
version	VARCHAR(200)	NOT NULL	모델 버전
model_type	VARCHAR(200)	NOT NULL	모델 유형
framework	VARCHAR(200)	NOT NULL	프레임워크
description	TEXT		모델 설명
create_date	DATETIME	DEFAULT CURRENT_TIMESTAMP	생성일
update_date	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	수정일

리포트 테이블 (Report)			
컬럼 이름	데이터 유형	제약 조건	설명
report_id	VARCHAR(200)	NOT NULL, PRIMARY KEY	리포트 고유 ID
model_id	VARCHAR(200)	NOT NULL, FOREIGN KEY (model_id) REFERENCES Model(model_id) ON UPDATE CASCADE ON DELETE CASCADE	모델 ID
admin_id	VARCHAR(200)	NOT NULL, FOREIGN KEY (admin_id) REFERENCES Model(admin_id) ON UPDATE CASCADE ON DELETE CASCADE	관리자 ID
user_id	VARCHAR(200)	NOT NULL, FOREIGN KEY (user_id) REFERENCES User(ID) ON UPDATE CASCADE ON DELETE CASCADE	사용자 ID
report_name	VARCHAR(200)	NOT NULL	리포트 이름
usage_date	DATETIME	NOT NULL	사용 일자
model_name	VARCHAR(200)	NOT NULL	모델 이름
accuracy	DECIMAL(5, 2)	NOT NULL	정확도
total_processing_amount	INT	NOT NULL	총 처리량
average_response_time	DECIMAL(10, 2)	NOT NULL	평균 응답 시간
create_date	DATETIME	DEFAULT CURRENT_TIMESTAMP	생성일
update_date	DATETIME	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	수정일

주요 class 역할 정의서

HibernateUtil

Hibernate ORM을 설정하고, 세션 팩토리를 관리하는 유틸리티 클래스.

AdminDAO

관리자(Admin) 객체에 대한 데이터 접근을 정의하는 인터페이스로, CRUD(Create, Read, Update, Delete) 작업과 관련된 메소드를 포함.

AdminDAOImpl

AdminDAO 인터페이스의 구현체로, 관리자(Admin) 객체에 대한 데이터베이스 접근 및 조작을 처리하는 클래스.

UserDTO

사용자 정보를 담기 위한 데이터 전송 객체(DTO)로, 사용자 관련 데이터를 안전하게 전송하고 검증하는 데 사용됨.

UserDAO

사용자(User) 객체에 대한 데이터 접근을 정의하는 인터페이스로, 사용자 관련 CRUD 작업 및 인증 관련 메소드를 포함.

UserDAOImpl

UserDAO 인터페이스의 구현체로, 사용자(User) 객체에 대한 데이터베이스 접근 및 조작을 처리하는 클래스.

주요 class 역할 정의서

ModelDAO

모델(Model) 객체에 대한 데이터 접근을 정의하는 인터페이스로, CRUD(Create, Read, Update, Delete) 작업과 관련된 메소드를 포함

ModelDAOImpl

ModelDAO 인터페이스의 구현체로, 모델(Model) 객체에 대한 데이터베이스 접근 및 조작을 처리하는 클래스.

ModelDTO

모델(Model) 객체에 대한 데이터 전송을 정의하는 클래스. 모델 ID, 이름, 버전, 타입, 프레임워크, 설명, 생성일 및 수정일 정보를 포함하며, 유효성 검사를 위한 어노테이션을 사용.

ReportDAO

리포트(Report) 객체에 대한 데이터 접근을 정의하는 인터페이스. CRUD(Create, Read, Update, Delete) 작업과 관련된 메서드를 포함.

ReportDAOImpl

ReportDAO 인터페이스의 구현체로, 리포트(Report) 객체에 대한 데이터베이스 접근 및 조작을 처리하는 클래스.

ReportDTO

리포트(Report) 객체에 대한 데이터 전송을 정의하는 클래스. 리포트 ID, 모델 ID, 관리자 ID, 사용자 ID, 리포트 이름, 사용 날짜, 모델 이름, 정확도, 총 처리량, 평균 응답 시간, 생성일 및 수정일 정보를 포함하며, 유효성 검사를 위한 어노테이션을 사용.

주요 class 역할 정의서

SignupController

사용자 회원가입을 처리하는 컨트롤러로,
사용자 입력 검증 및 데이터베이스에 사용자 정보를 저장하는 기능을 제공.

LoginController

사용자 및 관리자 로그인과 로그아웃을 처리하는 컨트롤러로, 인증 기능을 제공

InfoController

사용자 및 관리자 정보를 조회, 수정, 삭제하는 기능을 제공하는 컨트롤러로, 인증된 사용자에게 대한 정보를 관리.

UserController

사용자 관리 기능을 제공하는 컨트롤러로, 사용자 목록 조회 및 필터링 기능을 포함.

ModelController

모델 관리 기능을 제공하는 컨트롤러로, 모델 목록 조회 및 필터링 기능을 포함.

ModelManagementController

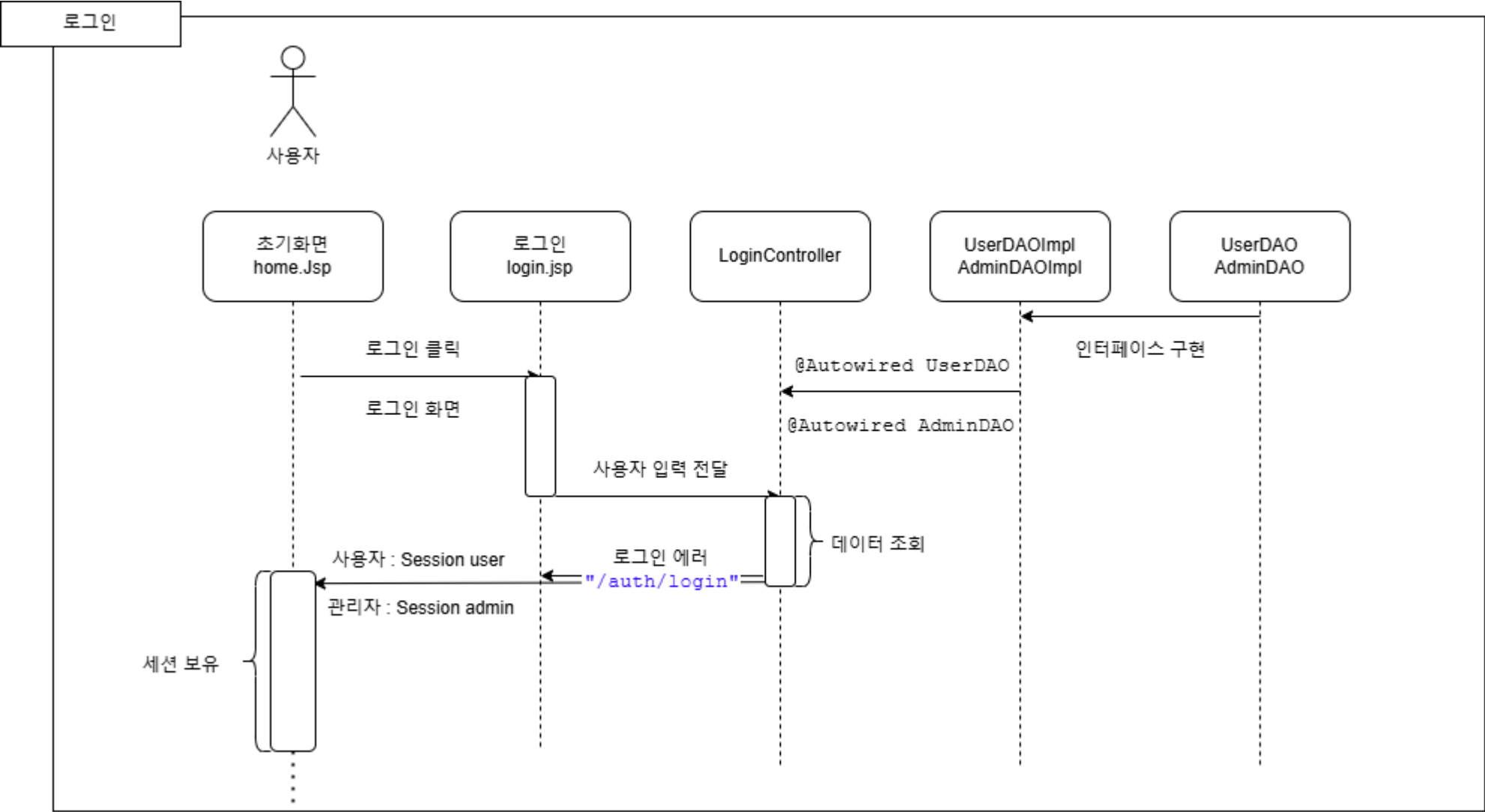
모델 관리 기능을 제공하는 컨트롤러로, 모델 생성, 조회, 삭제 등의 기능을 포함.

ReportController

리포트 관리 기능을 제공하는 컨트롤러로, 리포트 생성, 조회, 삭제 등의 기능을 포함.

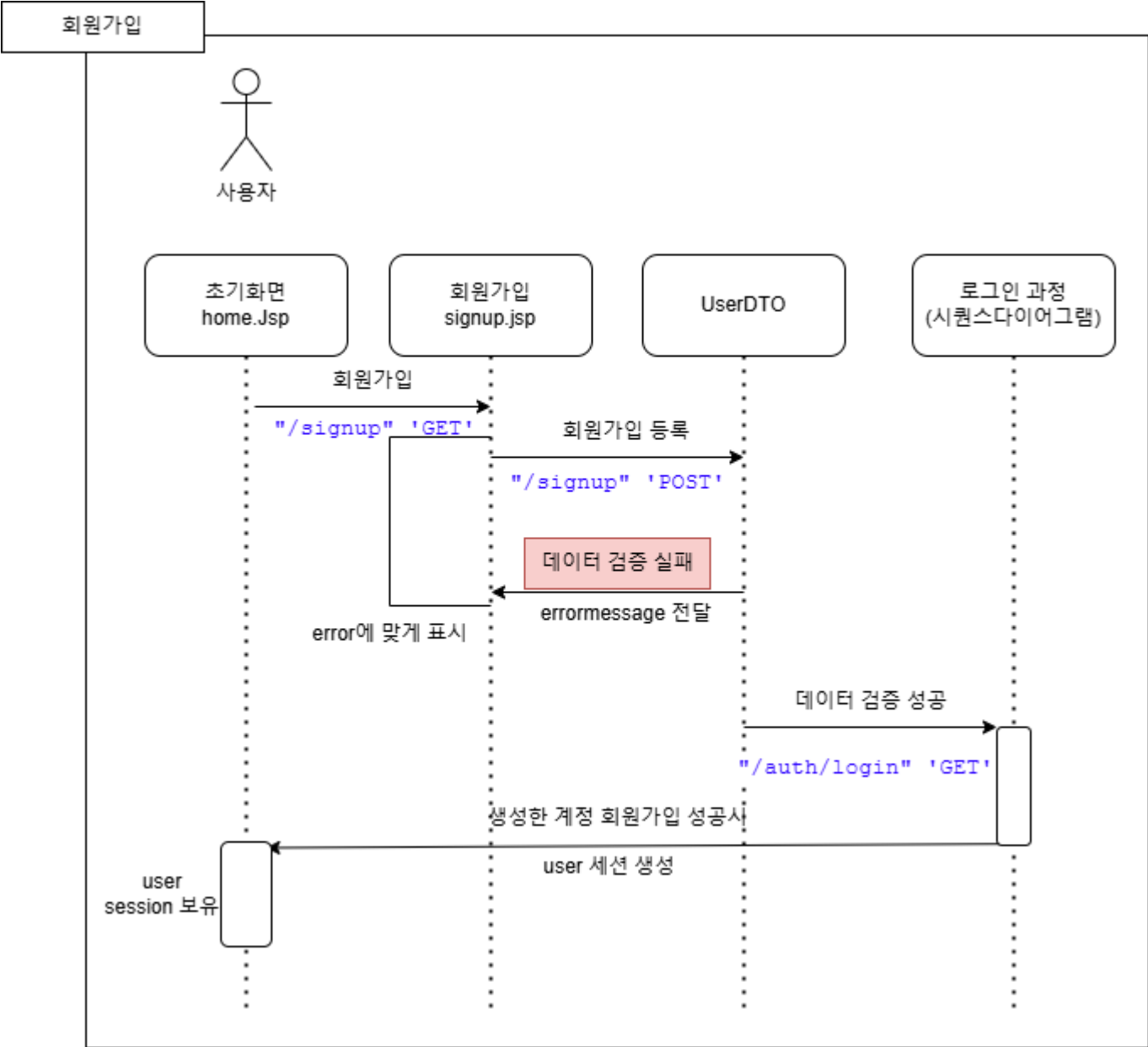
Sequence Diagram

로그인



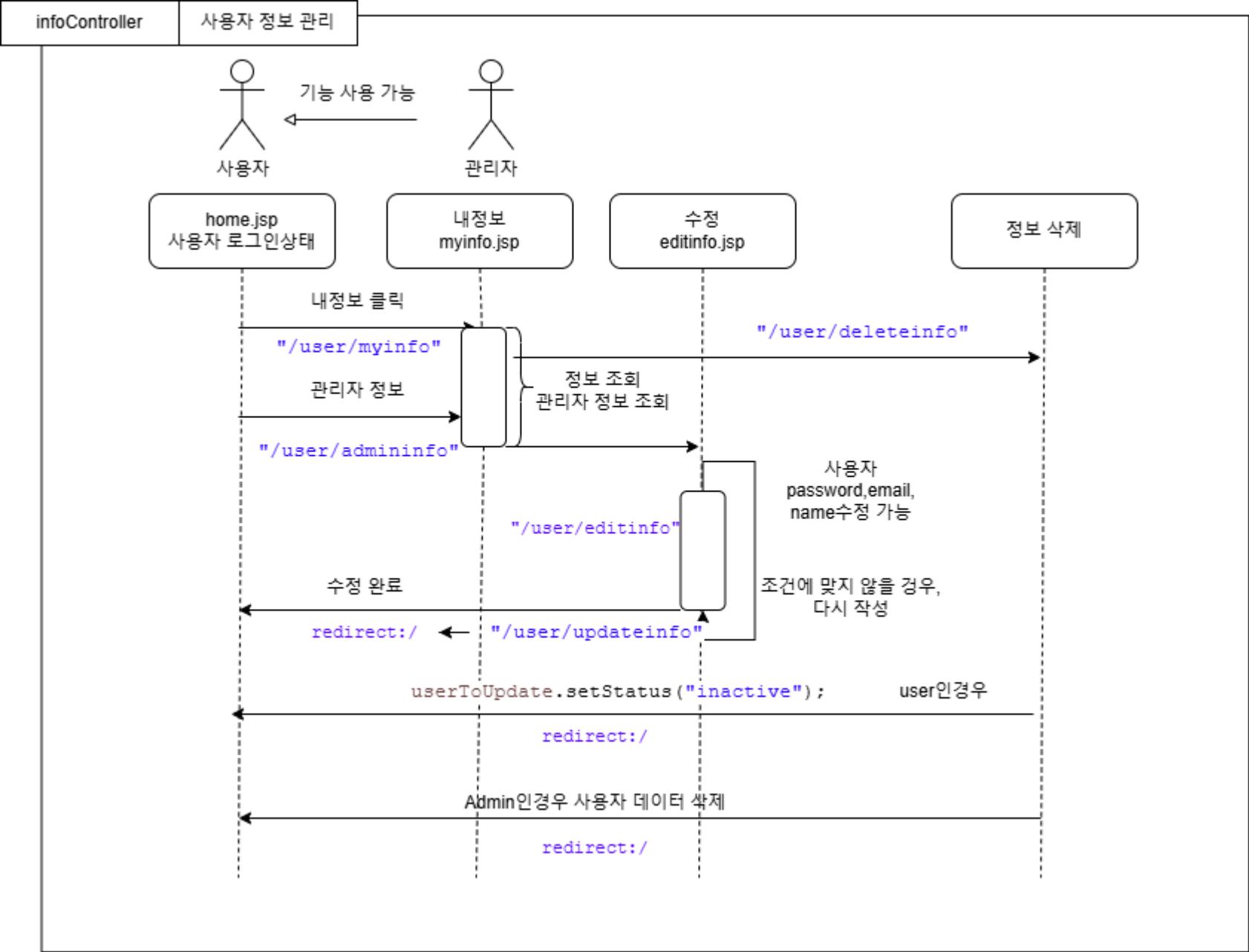
Sequence Diagram

회원가입



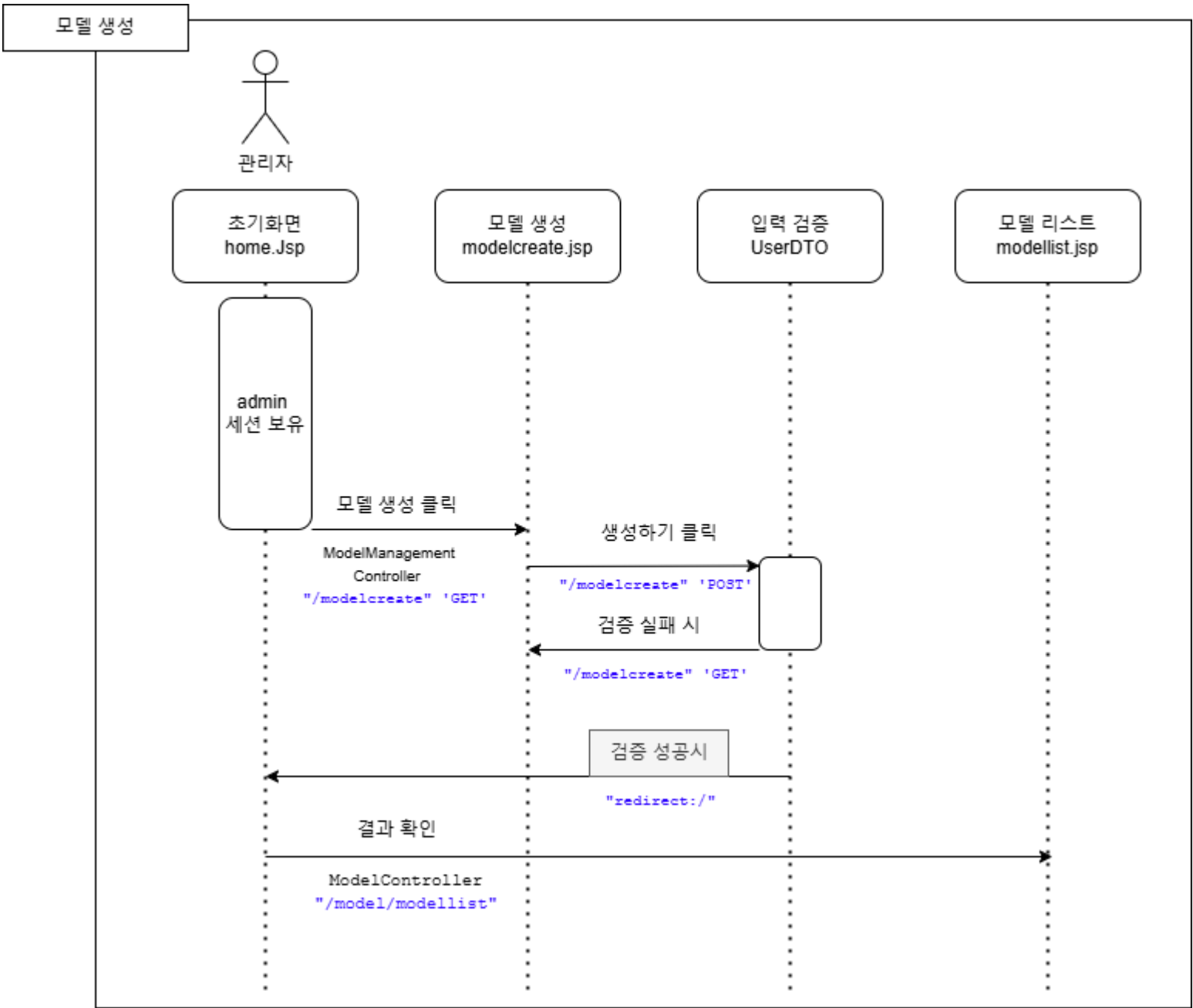
Sequence Diagram

사용자 정보 관리



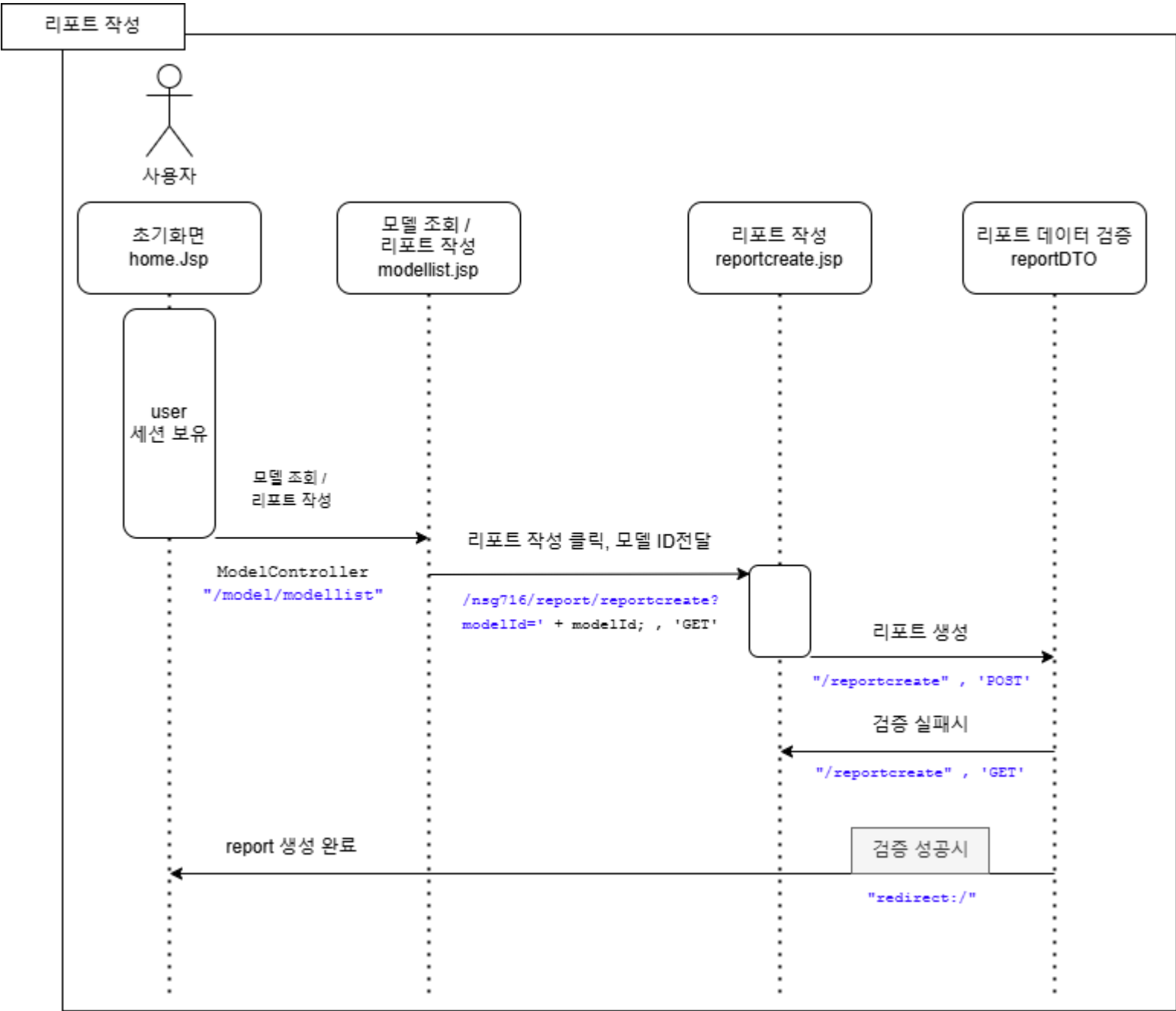
Sequence Diagram

모델 생성



Sequence Diagram

리포트 작성



Database 설계

Table 생성 (SQL 코드)

```
use wp_202020881;
-- 사용자 테이블 생성
CREATE TABLE User (
  ID VARCHAR(200) NOT NULL PRIMARY KEY,
  name VARCHAR(200) NOT NULL,
  email VARCHAR(200) NOT NULL,
  password VARCHAR(200) NOT NULL,
  status VARCHAR(200) NOT NULL,
  photo VARCHAR(200) DEFAULT NULL,
  create_date DATETIME DEFAULT CURRENT_TIMESTAMP,
  update_date DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

-- 관리자 테이블 생성
CREATE TABLE Admin (
  ID VARCHAR(200) NOT NULL PRIMARY KEY,
  name VARCHAR(200) NOT NULL,
  email VARCHAR(200) NOT NULL,
  password VARCHAR(200) NOT NULL,
  status VARCHAR(200) NOT NULL,
  photo VARCHAR(200) DEFAULT NULL,
  create_date DATETIME DEFAULT CURRENT_TIMESTAMP,
  update_date DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  last_login DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

-- 모델 테이블 생성
CREATE TABLE Model (
  model_id VARCHAR(200) NOT NULL PRIMARY KEY,
  admin_id VARCHAR(200) NOT NULL,
  model_name VARCHAR(200) NOT NULL,
  version VARCHAR(200) NOT NULL,
  model_type VARCHAR(200) NOT NULL,
  framework VARCHAR(200) NOT NULL,
  description TEXT,
  create_date DATETIME DEFAULT CURRENT_TIMESTAMP,
  update_date DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  FOREIGN KEY (admin_id) REFERENCES Admin(ID) ON UPDATE CASCADE ON DELETE CASCADE
);

-- 리포트 테이블 생성
CREATE TABLE Report (
  report_id VARCHAR(200) NOT NULL PRIMARY KEY,
  model_id VARCHAR(200) NOT NULL ,
  admin_id VARCHAR(200) NOT NULL,
  user_id VARCHAR(200) NOT NULL ,
  report_name VARCHAR(200) NOT NULL,
  usage_date DATETIME NOT NULL,
  model_name VARCHAR(200) NOT NULL,
  accuracy DECIMAL(5,2) NOT NULL,
  total_processing_amount INT NOT NULL,
  average_response_time DECIMAL(10,2) NOT NULL,
  create_date DATETIME DEFAULT CURRENT_TIMESTAMP,
  update_date DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  FOREIGN KEY (model_id) REFERENCES Model(model_id) ON UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (admin_id) REFERENCES Model(admin_id) ON UPDATE CASCADE ON DELETE CASCADE,
  FOREIGN KEY (user_id) REFERENCES User(ID) ON UPDATE CASCADE ON DELETE CASCADE
);

INSERT INTO User (ID, name, email, password, status, photo) VALUES
('user001', 'Alice', 'alice@example.com', 'password123!', 'active', NULL),
('user002', 'bob', 'bob@example.com', 'password123!', 'active', NULL),
('user003', 'charlie', 'charlie@example.com', 'password123!', 'active', NULL),
('user004', 'david', 'david@example.com', 'password123!', 'active', NULL),
('user005', 'eve', 'eve@example.com', 'password123!', 'active', NULL),
('user006', 'frank', 'frank@example.com', 'password123!', 'active', NULL),
('user007', 'grace', 'grace@example.com', 'password123!', 'active', NULL),
('user008', 'hank', 'hank@example.com', 'password123!', 'active', NULL),
('user009', 'ivy', 'ivy@example.com', 'password123!', 'active', NULL),
('user010', 'jack', 'jack@example.com', 'password123!', 'active', NULL);

INSERT INTO Admin (ID, name, email, password, status, photo) VALUES
('admin001', 'Admin1', 'admin1@example.com', 'adminpass', 'active', NULL),
('admin002', 'Admin2', 'admin2@example.com', 'adminpass', 'active', NULL),
('admin003', 'Admin3', 'admin3@example.com', 'adminpass', 'active', NULL);

INSERT INTO Model (model_id, admin_id, model_name, version, model_type, framework, description) VALUES
('model001', 'admin001', 'Model A', '1.0', 'Classification', 'TensorFlow', 'Classification model for images'),
('model002', 'admin002', 'Model B', '2.0', 'Regression', 'PyTorch', 'Regression model for sales prediction'),
('model003', 'admin003', 'Model C', '1.5', 'Clustering', 'Scikit-learn', 'Clustering model for customer segmentation');

INSERT INTO Report (report_id, model_id, admin_id, user_id, report_name, usage_date, model_name, accuracy, total_processing_amount, average_response_time) VALUES
('report001', 'model001', 'admin001', 'user001', 'Report A', '2024-01-01 10:00:00', 'Model A', 95.50, 100, 0.45),
('report002', 'model002', 'admin002', 'user002', 'Report B', '2024-01-02 11:00:00', 'Model B', 89.20, 200, 0.78),
('report003', 'model003', 'admin003', 'user003', 'Report C', '2024-01-03 12:00:00', 'Model C', 92.10, 150, 0.50);
```


Test 방법 단위 테스트 방법

사전 작업 - 자유롭게 테스트 가능

- 기초 데이터 생성
 - schema - wp_202020881
 - wp_202020881
 - id : nsg716
 - password : 202020881
 - 메일에 첨부한 기말고사 SQL 데이터 생성.sql 실행
- **RestControllerTest**
 - 폴더 위치: /src/test/java
 - 패키지: kr.ac.kku.cs.wp.wsd.nsg716.controller
- testTestEndpoint() - /Test 엔드포인트 테스트
- testLogging() - 로깅 기능 테스트
- **ViewControllerTest**
 - 폴더 위치: /src/test/java
 - 패키지: kr.ac.kku.cs.wp.wsd.nsg716.web.controller
- testHomeGet() - GET /home 엔드포인트 테스트
- testHomePost() - POST /home 엔드포인트 테스트

Test 방법 단위 테스트 방법

Unit 테스트

- **LoginControllerTest**
 - 폴더 위치 - /src/test/java
 - package
 - kr.ac.kku.cs.wp.wsd.nsg716.aaa.controller
 - testLoginAsUserSuccess() - 사용자로 로그인 성공 테스트
 - testLogout() - 사용자 로그아웃 테스트
 - testLoginAsAdminSuccess() - 관리자 로그인 성공 테스트
 - testLogoutAsAdmin() - 관리자 로그아웃 테스트
 - testLoginFailure() - 로그인 실패 테스트
 - testLoginPageAccess() - 로그인 페이지 접근 테스트
- **UserControllerTest**
 - 폴더 위치: /src/test/java
 - 패키지: kr.ac.kku.cs.wp.wsd.nsg716.user.controller
 - testUserListAccess() - 사용자 목록 접근 테스트
 - testFilterUserListByName() - 사용자 이름으로 필터링 테스트
 - testFilterUserListById() - 사용자 ID로 필터링 테스트
 - testFilterUserListWithEmptyFilter() - 빈 필터 파라미터로 사용자 리스트 테스트
 - testFilterUserListNullParams() - Null 파라미터로 사용자 리스트 테스트

Test 방법 단위 테스트 방법

Unit 테스트

- **InfodataControllerTest**

- 폴더 위치: /src/test/java
- 패키지
 - kr.ac.kku.cs.wp.wsd.nsg716.aaa.controllertest
- testUpdateInfo_UserSuccess() - 사용자 정보 업데이트 성공 테스트
- testDeleteInfo_UserSelf() - 사용자 본인 삭제 테스트
- testGetAdminInfo_AdminSession() - 관리자 세션에서 관리자 정보 조회 테스트
- testDeleteInfo_AdminDeleteUser1() - 관리자가 사용자1 삭제 테스트
- testDeleteInfo_AdminDeleteUser2() - 관리자가 사용자2 삭제 테스트
- testEditInfo_UserSession() - 사용자 세션에서 내 정보 조회 테스트
- testMyInfoGet_UserSession() - 사용자 세션에서 내 정보 조회 테스트
- testDeleteInfo_NoSession() - 세션 없는 상태에서 사용자 삭제 시도 테스트
- testGetAdminInfo_NoSession() - 세션 없는 상태에서 관리자 정보 조회 시도 테스트
- testMyInfoGet_AdminSession() - 관리자 세션에서 내 정보 조회 시도 테스트
- testMyInfoGet_NoSession() - 세션 없는 상태에서 내 정보 조회 시도 테스트

- SignupPageAccess() - 회원가입 페이지 접근 테스트
- testSignupAdminSuccess() - 관리자 회원가입 성공 테스트
- testSignupSuccess() - 일반 사용자 회원가입 성공 테스트
- testSignupError_AdminInId() - 관리자 ID로 회원가입 시도 테스트
- testSignupError_DuplicateId() - 중복된 ID로 회원가입 시도 테스트
- testSignupError_ValidationFailures() - 유효성 검사 실패 시 회원가입 시도 테스트

Test 방법 단위 테스트 방법

Unit 테스트 - Model

- ModelControllerTest
 - 폴더 위치: /src/test/java
 - 패키지: kr.ac.kku.cs.wp.wsd.nsg716.model.controller
 - testModelListAccessByAdmin() - 관리자로 모델 리스트 접근 테스트
 - testModelListAccessByUser() - 일반 사용자로 모델 리스트 접근 테스트
 - testModelListAccessUnauthorized() - 미인증 사용자의 모델 리스트 접근 테스트
 - testFilterModelListByModelName() - 모델 이름으로 필터링 테스트
 - testFilterModelListByFramework() - 모델 프레임워크로 필터링 테스트
 - testFilterModelListWithEmptyFilter() - 빈 필터 파라미터로 모델 리스트 테스트
 - testFilterModelListNullParams() - Null 파라미터로 모델 리스트 테스트
-
- ModelManagementControllerTest
 - 폴더 위치: /src/test/java
 - 패키지: kr.ac.kku.cs.wp.wsd.nsg716.model.controller
 - testShowModelCreatePage_WithAdminSession() - 관리자 세션으로 모델 생성 페이지 접근 테스트
 - testShowModelCreatePage_WithoutSession() - 세션 없는 상태에서 모델 생성 페이지 접근 테스트
 - testCreateModel_Success() - 모델 생성 성공 테스트
 - testCreateModel_DuplicateModelId() - 중복된 모델 ID로 모델 생성 시도 테스트
 - testInfoModel_ValidModel() - 유효한 모델 정보 조회 테스트
 - testDeleteModel_Success() - 모델 삭제 성공 테스트
 - testDeleteModel_ModelNotFound() - 존재하지 않는 모델 삭제 시도 테스트

Test 방법 단위 테스트 방법

Unit 테스트 – report

- ReportControllerTest
 - 폴더 위치: /src/test/java
 - 패키지: kr.ac.kku.cs.wp.wsd.nsg716.report.controller
- testShowReportCreatePage_WithoutSession() - 보고서 생성 페이지 접근 테스트 - 세션 없음
- testCreateReport_Success() - 보고서 생성 테스트 - 성공 시나리오
- testCreateReport_DuplicateReportId() - 보고서 생성 테스트 - 중복된 보고서 ID
- testReportList_WithAdminSession() - 보고서 목록 조회 테스트 - 관리자 세션
- testFilterReportList_WithParameters() - 보고서 필터링 테스트 - 파라미터 사용
- testReportInfo_Success() - 보고서 상세 정보 조회 테스트 - 성공
- testUpdateReportInfo_Success() - 보고서 정보 업데이트 테스트 - 성공
- testDeleteReport_Success() - 보고서 삭제 테스트 - 성공
- testFilterReportList_NullParams() - 보고서 필터링 테스트 - Null 파라미터

JUnit 테스트 종료

Test 방법 통합 테스트 방법

테스트 순서

0. 접속 테스트

0-1. 홈

0-2. 비로그인 접속

1. 회원가입

2. 사용자 로그인

2-1. 사용자 - 정보

2-2. 사용자 - 정보 수정

2-3. 사용자 - 상태변경

2-3-1. 사용자 - 재로그인 및 정보

2-4. 사용자 - 모델 리스트

2-4-1. 모델 리포트 작성

2-5. 사용자 - 리포트 리스트 조회

2-6. 사용자 - 리포트 정보

2-6-1 사용자 - 리포트 수정

2-7. 사용자 - 로그아웃

3. 관리자 로그인

3-1. 관리자 - 관리자 정보

3-2. 관리자 - 사용자 생성

3-3. 관리자 - 사용자 리스트

3-3-1. 관리자 - 사용자 조회

3-3-2. 관리자 - 사용자 삭제

3-4. 관리자 - 모델 리스트

3-4-1. 관리자 - 모델 생성

3-4-2. 관리자 - 모델 조회

3-4-3. 관리자 - 모델 수정

3-4-4. 관리자 - 모델 삭제

3-5. 관리자 - 모델 리포트

3-5-1. 관리자 - 리포트 조회

3-5-2. 관리자 - 리포트 수정

3-5-2. 관리자 - 리포트 삭제

3-6. 관리자 - 로그아웃

Test 방법 통합 테스트 방법

순서	테스트 번호	확인내용	방법	확인여부
1	0	접속 테스트	/localhost:8080/nsg716/Test접속	
2	0-1	홈 접속 테스트	/localhost:8080/nsg716/ 접속	
3	0-2	비로그인 접속	localhost:8080/nsg716/user/deleteinfo 접속	
4			로그인 되지 않았기에 /login 접속	
5			log - Unauthorized access attempt to delete user information 확인	
6	1	회원가입	/localhost:8080/nsg716/ 접속	
7			메뉴에서 회원가입 클릭	
8		오류 테스트	아이디 : user001 이름 : sktjdrhs 이메일 : sktjdrhs@example.com 비밀번호: password123 입력	
9			아이디 중복 확인 , 비밀번호 오류 메시지 확인 , 이름, 이메일 필드 입력 유지 확인	
10			아이디 : admin001, 비밀번호 password123! 입력	
11			아이디 오류 확인, 비밀번호 오류 메시지 사라지는지 확인	
12			아이디 : user999 비밀번호 password123! 입력	
13	2	사용자 로그인	로그인 페이지 이동	
14		사용자 로그인 실패	user999, password123 입력	
15			오류 메시지 확인	
16		사용자 로그인 성공	user999, password123! 입력	
17			홈으로 이동 확인	
18	2-1	사용자 – 내 정보	사이드바 내 정보 클릭	
19			내 정보 확인 – 아이디,이름,이메일,비밀번호 확인, 상태 확인	
20	2-2	사용자 – 정보 수정	정보수정 클릭	
21			정보 수정 페이지 이동 확인	

Test 방법 통합 테스트 방법

순서	테스트 번호	확인내용	방법	확인여부
22		아이디 클릭해보기	아이디 클릭 – 수정 안되면 성공	
23		사용자 – 정보 수정 오류 확인	이름 : SKTJDRHS 비밀번호 : 맨 뒷자리 1개 제거 - 수정 완료 클릭	
24			오류 메시지 확인	
25			비밀번호 password123!으로 설정	
26		사용자 이름 변경 확인	홈으로 이동 – 로그아웃 옆에 이름 변경 확인	
27			내 정보 다시 들어가서 수정일 변경 확인	
28	2-3	사용자 상태변경	상태 변경 클릭	
29			세션 할당 해제 - 로그인 취소 상태 확인	
30	2-3-1	사용자 재로그인 및 정보	아이디 user999 비밀번호 : password123!으로 재로그인	
31			내 정보 클릭 – 상태 inactive 변경 확인	
32		상태에 따른 메뉴 이동 제한	모델 / 리포트 관리 – 모델 조회/ 리포트 작성 클릭 모델 리포트 – 리포트 조회 클릭 – 둘다 막히는지 확인	
33		로그아웃	로그아웃 클릭	
34	2-4	사용자 – 모델 리스트	아이디 user001 , 비밀번호 password123! 으로 로그인	
35			모델 / 리포트 관리 – 모델 조회/ 리포트 작성 클릭	
36		모델 리스트 확인	이름, 버전,유형, 프레임워크를 입력하여 검색 확인	
37			Model A – 리포트 작성 클릭	
38	2-4-1	모델 리포트 작성	리포트 작성	
39		오류 확인	리포트 아이디 report004 이름 : report D 사용날짜 : 현재 날짜 및 시간 입력 정확도 : 87.5 총 처리량 : 100 평균응답시간 : 10 입력	
40			홈으로 이동 확인	

Test 방법 통합 테스트 방법

순서	테스트 번호	확인내용	방법	확인여부
41	2-5	리포트 리스트 조회	모델 리포트 – 리포트 조회	
42			report D를 확인, 생성날짜 일치 확인	
43			검색 – A 입력 시 검색 조건 확인	
44	2-6	리포트 정보	report D 보고서 정보 확인	
45			보고서에 모델 생성한 관리자 아이디, 보고서 생성 사용자 아이디 등 정보 확인	
46			밑에 정보수정 클릭	
47			보고서 정보 수정	
48		오류 확인	항목 공백, 정확도, 총 처리량, 평균응답시간 음수 시 오류 발생	
49			수정 성공 시 홈으로 이동	
50			모델 리포트 – 리포트 조회 클릭	
51			자세히 보기 클릭 – 수정 확인	
52	2-7	사용자 로그아웃	로그아웃 클릭	
53	3	관리자 로그인	아이디 : admin001 비밀번호 : adminpass	
54			로그인 성공 확인	
55	3-1	관리자 – 관리자 정보	내 정보 확인 – 최근 접속 시간 확인 (로그아웃 갱신)	
56			로그아웃 이후 다시 로그인 시 갱신 확인	
57	3-2	관리자 – 사용자 생성	회원가입과 같은 방식	
58			아이디 user990 이름 sktjdrhs123 이메일 sktjdrhs123@example.com 비밀번호 : password123! 입력	
59			사용자 생성 성공 확인 – 세션이 관리자인지 확인 – 홈 이름 확인	
60	3-3	관리자 – 사용자 리스트	사용자 관리 – 사용자 조회 / 삭제 클릭	
61			밑에 SKTJDRHS이름 사용자 확인	

Test 방법 통합 테스트 방법

순서	테스트 번호	확인내용	방법	확인여부
62		상태에 따른 표시 방법	active – 초록색 , inactive - 빨강색	
63			이름, 이메일 상태로 검색	
64	3-3-1	관리자 – 사용자 조회	SKTJDRHS – 보기 클릭	
65			해당 사용자의 정보 확인	
66			사용자 관리 – 사용자 조회 / 삭제 클릭	
67	3-3-2	관리자 – 사용자 삭제	아이디 : sktjdrhs123 삭제	
68			리스트에서 삭제 확인	
69	3-4	관리자 모델 리스트	모델 관리 - 모델 조회 / 삭제 클릭	
70			모델 이름,버전, 유형,프레임워크 검색 확인	
71	3-4-1	관리자 – 모델 생성	모델 관리- 모델 생성 클릭	
72		오류 메시지 확인	(모델 아이디 : model001 모델 이름 : Model D 버전 : 2.5 모델 타입 Classification 프레임워크 : TensorFlow 설명 미 입력)입력	
73			오류 확인 – 모델 아이디 : model005 로 수정	
74			모델 생성 완료 – 홈 화면 이동	
75			모델 관리 – 모델 조회 / 삭제 클릭	
76			모델 생성 확인	
77	3-4-2	모델 조회	Model D – 정보 보기 클릭	
78			모델 정보 확인 – 관리자 ID는 로그인한 아이디가 들어갑니다.	
79	3-4-3	모델 수정	모델 정보 – 모델 수정 클릭	
80			설명 모델 5입니다. 입력	
81			수정 성공 시 홈 화면 이동	
82			모델 관리 – 모델 조회 / 삭제 Model D 정보보기 클릭	

Test 방법 통합 테스트 방법

순서	테스트 번호	확인내용	방법	확인여부
83			설명 추가 확인 및 수정일 확인	
84	3-4-4	관리자 – 모델 삭제	모델 관리 – 모델 조회 / 삭제 클릭	
85			Model D 삭제 클릭	
86			삭제된 이후 모델 리스트 재정렬 확인	
87	3-5	관리자 – 모델 리포트	모델 리포트 – 리포트 조회 / 삭제 클릭	
88			리포트 이름,모델 이름으로 검색 확인	
89	3-5-1	관리자 – 리포트 조회	Report C 자세히 보기 클릭	
90			사용자가 작성한 리포트의 상세 정보 확인	
91	3-5-2	관리자 – 리포트 수정	보고서 정보 – 정보 수정 클릭	
92			정확도 95로 수정	
93			수정 성공 시 홈 화면 이동	
94			모델 리포트 – 리포트 조회 / 삭제 - Report C - 자세히 보기	
95			수정 내용 및 수정일 변경 확인	
96	3-5-3	관리자 – 리포트 삭제	모델 리포트 – 리포트 조회 / 삭제 클릭	
97			Report C – 삭제 클릭	
98			삭제 성공 시 리포트 리스트 갱신	
99	3-6	관리자 – 로그아웃	로그아웃 클릭	
100			로그인 화면 이동	