

프로젝트명	PyP Trip
팀명 (팀원,팀원)	박승수, 유영
진행일자	2023.05.18~05.26

1. 목표

- Web Architecture를 이해하고 활용하여 Web Project를 설계하고 구현할 수 있다.
- Spring & MyBatis Framework, SpringBoot, Vue.js를 이해하고 활용할 수 있다.
- OAuth 로그인, REST API 등의 기술을 활용하여 MSA를 도입해 본다.
- 여행에 적합한 추가 기능을 적용해 본다.
- 다양한 보안사항을 고민해 본다.

2. 준비사항

1) 사용 데이터

- 전국관광지 정보

한국관광공사_국문관광정보 서비스_GW.

(<https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15101578>)

분류체계	관리처분기관 - 관광	제공기관	한국관광공사
관리처분체계	K7관광정보	관리처분처 연혁번호	003-7100-0004
API 유형	REST	데이터포맷	JSON+XML
분류코드	270	카테고리	여행/관광/관광정보
등록	2022-06-04	수정	2022-06-04
이용대상	무제한	신청가능 주체	/ 관광목적의 관광에 필요한 관광정보 제공 가능
이용제한	제한없음 / 관광목적 / 관광목적		
이용제한	제한없음 / 관광목적 / 관광목적		
접근성	제한없음 / 관광목적 / 관광목적		

- 관광지 사진정보

한국관광공사_관광사진정보 서비스_GW.

<https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15101914>

- 전국 전기차 충전소정보
한국환경공단_전기자동차 충전소 정보
<https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15076352>
- 날씨정보
지역별 날씨 정보
<https://openweathermap.org/api>
- 한국천문연구원_출몰시각 정보
지역별 일출 일몰 정보
<https://www.data.go.kr/tcs/dss/selectApiDataDetailView.do?publicDataPk=15012688>

2)

2) 개발언어/프로그램

- Java/STS/Tomcat/MySQL/VSCode

3) 필수 라이브러리 / 오픈소스

- Spring Framework (SpringBoot)
- MyBatis framework
- Vue.js / JavaScript / Bootstrap-Vue

4) 다양한 알고리즘

- DFS
- BFS
- 문자열 알고리즘
- 동적 프로그래밍

3. 요구사항

사용자에게 한국의 다양한 관광지, 먹거리, 축제, 행사 등을 소개하여 지역관광 활성화를 위한 소개 페이지를 구축하려 한다. 한국관광공사에서 제공하는 국문관광정보서비스_GW의 다양한 상세 기능정보 API를 활용하여 지역별 관광지 data를 분석하고 화면에 표시한다. 또한 여행계획을 위한 스케줄과 여행경로 공유 등 사용자 편의 기능을 구현하고 나만의 숨은 관광지를 소개하는 페이지와 자유롭게 토론이 가능한 게시판 등을 구현해본다.

보안의 대해 깊게 고민하며 CSRF 방어, 비밀번호 해쉬, SALT, FileUpload/download 취약점 방어, 세션 하이제킹 방어, 세션 중복문제 해결과 같은 기능들을 구현한다. 또한 여행자끼리의 정보를 공유하기 위해 자신의 위치 기반으로 가까운 게시판을 보여주는 기능들을 구현한다.

Usecase Form

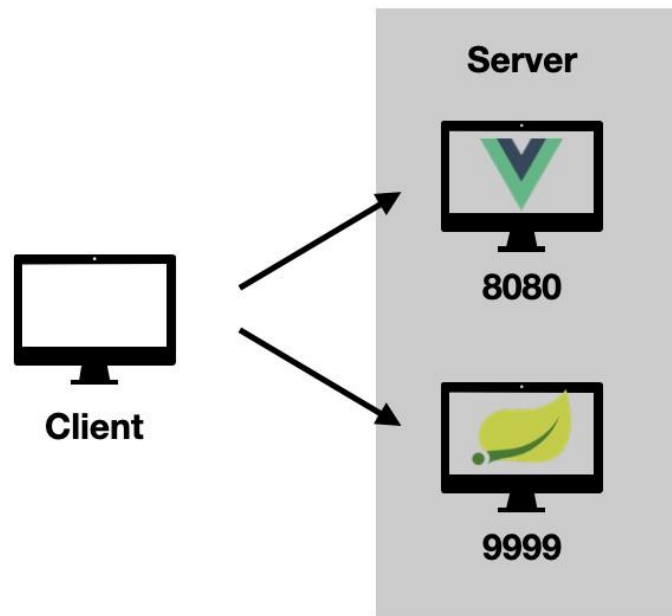
no	요구사항명	요구사항 상세	우선순위	구현여부(O/X)
F01	지역별 관광지 정보 수집	한국관광공사의 지역별 관광지 정보를 얻어와 화면에 표시	필수	O
F02	관광지, 숙박, 음식점 조회	관광지 정보를 지역별 원하는 컨텐츠 별 조회	필수	O
F03	문화시설, 공연,여행코스,쇼핑 조회	관광지 정보를 지역별 원하는 컨텐츠 별 조회	필수	O
F04	여행 계획 경로 설정	조회한 관광지를 활용하여 여행 계획, 여행 경로를 저장	추가	X
F05	회원 주도의 핫플레이스 등록	지도와 사진을 활용한 핫플레이스 등록	추가	X
F06	관광지 관련 뉴스 정보 크롤링	관광지 정보를 크롤링하여 DB에 저장	심화	O
F07	회원관리	회원가입/수정/조회/탈퇴	필수	O
F08	인증관리	로그인/로그아웃/비밀번호 찾기	필수	O
F09	공지사항	공지사항 등록/수정/삭제/조회	심화	X
F10	공유게시판	게시판 등록/수정/삭제/조회	심화	O
F11	관광지 날씨	관광지의 기간별 날씨 출력		X
F12	관광지 사진	관광지별 추천 사진 출력		O
F13	일출, 일몰 시각	관광지별 일출, 일몰 시간 출력		X
F14	전기자동차 충전소	전기자동차의 충전소의 위치 및 충전 상태 출력		X
비기능적 요구사항				
NF1	공공데이터의 정확	공공데이터 API를 활용함으로 인		O

	성	한 정확성이 요구됨		
NF2	가용성	언제나 (어떤 디바이스로든) 서비스 가능해야 함		O
NF3	응답성	조회에 대한 결과를 빠르게 응답해야 함		O
NF4	사용자 편의성	웹 사이트에 대한 사전 지식이 없어도 쓰기 편해야 함		O
NF5	안전성	비밀번호 해싱		O
NF6	안전성	CSRF 방어		O
NF7	안전성	FileUpload/download 취약점 방어		O
NF8	안전성	세션 하이재킹 방어		O
NF9	안전성	SALT, 암호화 키 관리		O
NF10	안전성	세션 중복 처리		O
NF11	안전성	유효성 검사		O

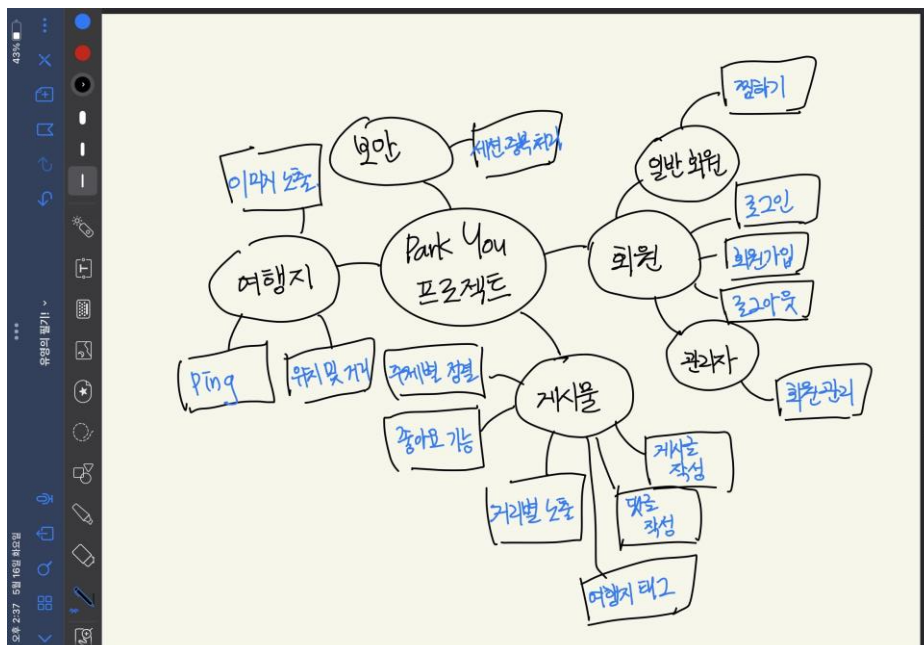
4. 결과 (산출물)

1. 설계서

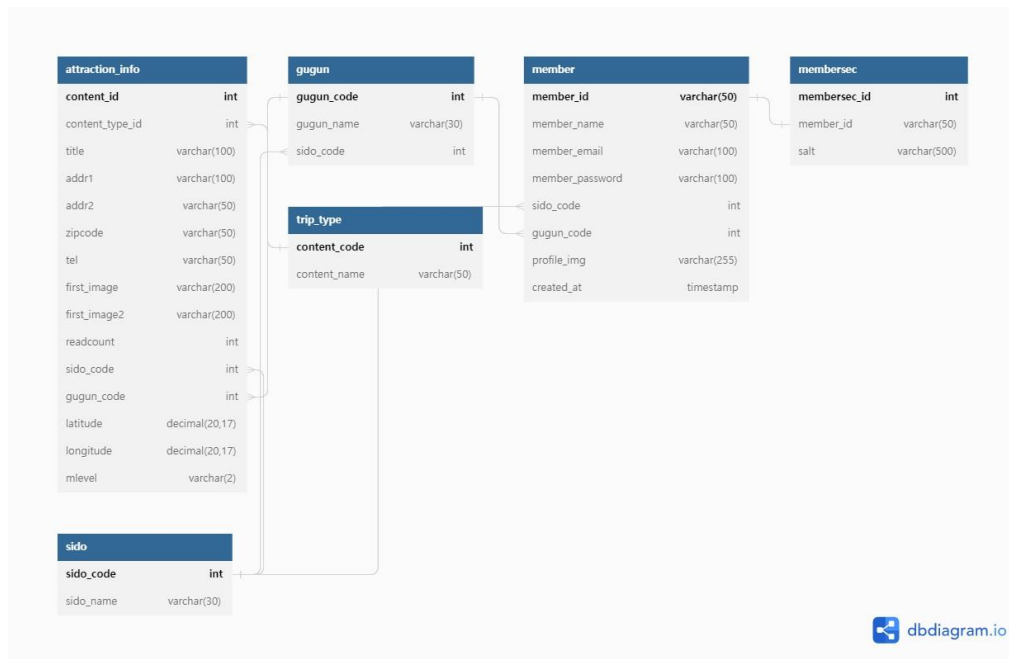
(1)Architecture Diagram



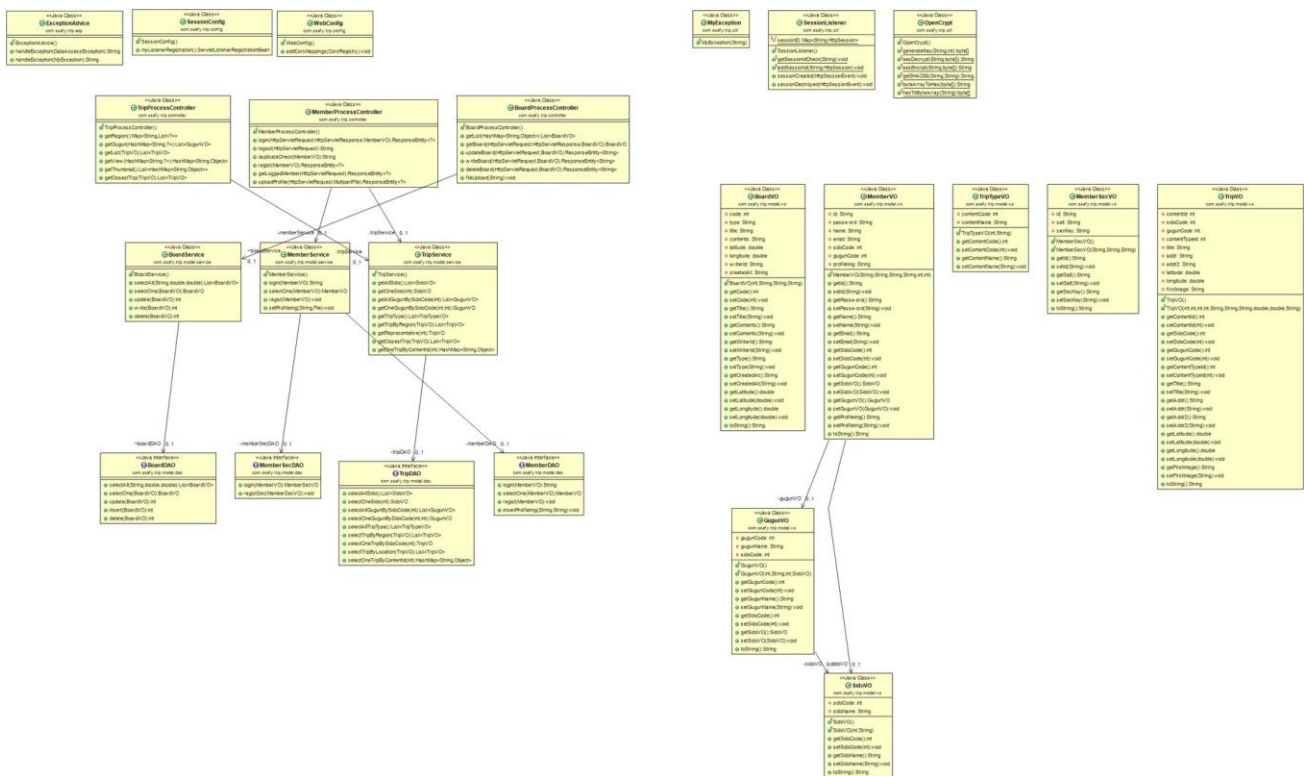
2. UseCase Diagram



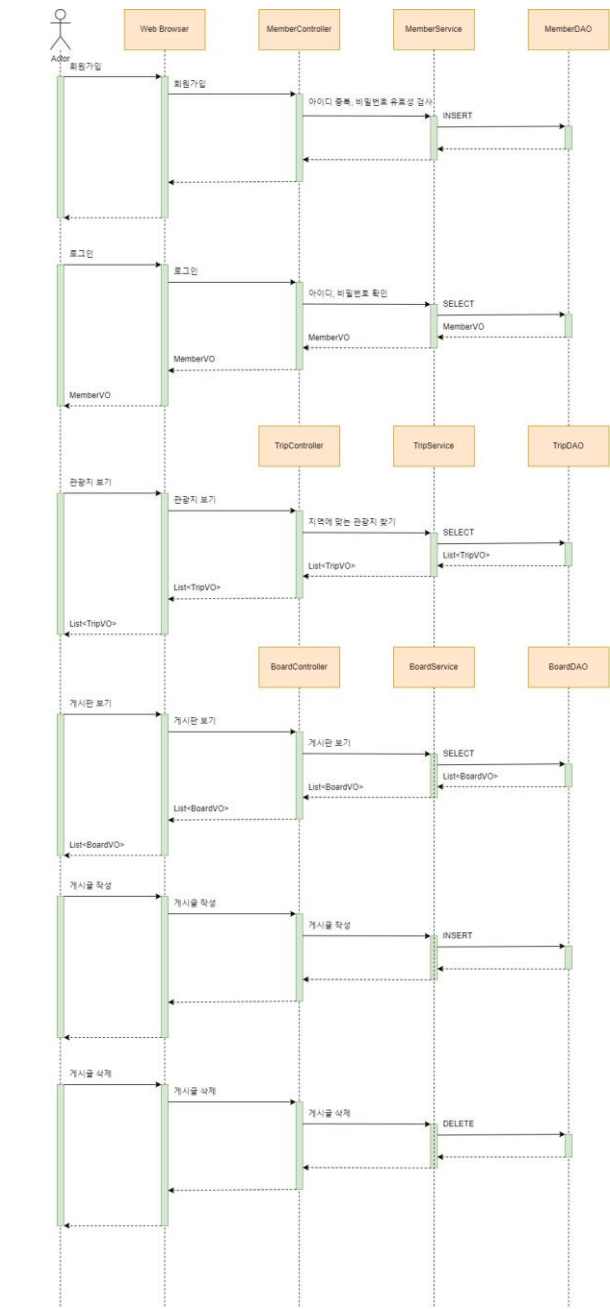
3. ERD



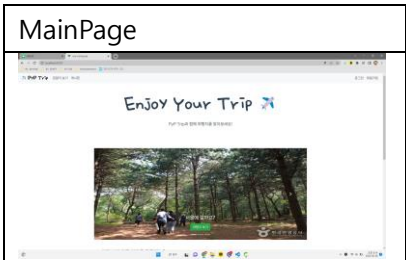
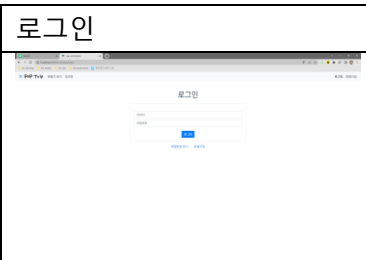
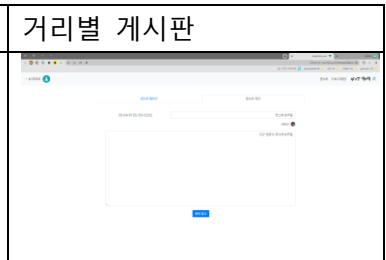
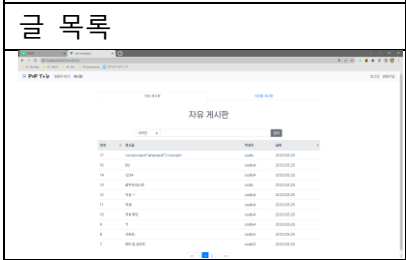
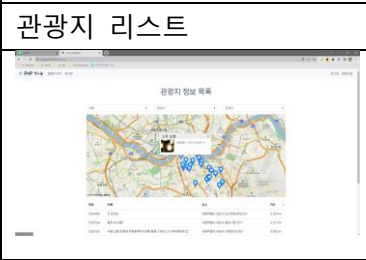
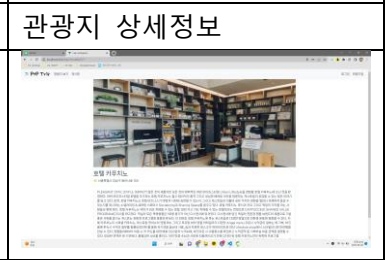
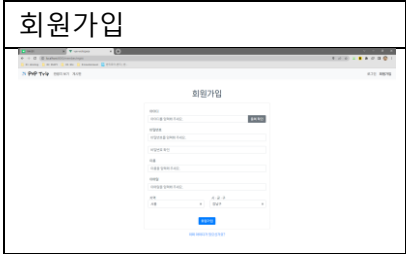
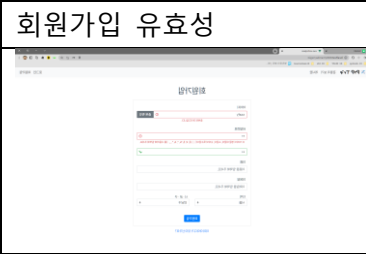
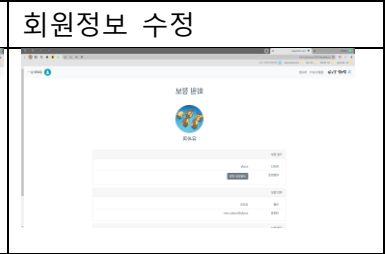
4. Class Diagram



5. Sequence Diagram



6. Ui 설계서

<p>MainPage</p> 	<p>로그인</p> 	<p>거리별 게시판</p> 
<p>글 목록</p> 	<p>관광지 리스트</p> 	<p>관광지 상세정보</p> 
<p>회원가입</p> 	<p>회원가입 유효성</p> 	<p>회원정보 수정</p> 

7.



PyP TRIP.

SSAFY 9기 서울 19반 박승수, 유명



프로젝트 목표





기능 geolocation API

사용자의 현재 위치 정보를 가져올 때 사용하는 자바스크립트 API



```
<select id="selectAll" parameterType="double" resultType="BoardVO">
  SELECT *, writer_id AS writerId, created_at AS createdAt FROM ${type}_board
</select>
<where>
  <if test="type == 'location'">
    AND (
      6371 * acos ( cos ( radians(#{latitude}) )
        * cos( radians( latitude ) )
        * cos( radians( longitude ) - radians(#{longitude}) )
        + sin ( radians(#{latitude}) ) * sin( radians( latitude ) )
      ) <= 5000
    )
  </if>
</where>
ORDER BY created_at DESC
</select>
```



보안

1. 유효성 검사
2. CSRF 검사
3. session 탈취 여부 검사
4. session 중복 검사
5. 업로드 file 형식 검사





보안

유효성 검사

```

}
this.save({
  uri: "member/duplicate-check",
  method: "post",
  data: { id: this.member.id },
});
then((response) => {
  if (response.data != "ok") {
    this.validateCheckId = false;
    this.validateCheckIdFeedback = "중복된 아이디";
  } else {
    this.validateCheckId = true;
  }
});
},
register() {
  if (this.validateCheckId != true) {
    this.modalMsg = "아이디 중복 확인을 하주세요.";
    this.$bvModal.show('bv-modal');
    return;
  }
  if (this.validateCheck.password != true) {
    this.modalMsg = "비밀번호를 다시 입력해 주세요.";
    return;
  }
  if (this.validateCheck.email != true) {
    this.modalMsg = "이메일을 다시 입력해 주세요.";
    return;
  }
}
}

```

Front

```

ic void regist(MemberVO member) throws MyExc
// 아이디 중복 검사
MemberVO selMember = selectOne(member);
if(selMember != null) throw new MyException(
// 비밀번호 중복 검사
String pwReg = "(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[!@#$%^&*~])";
if(!Pattern.matches(pwReg, member.getPassword())
throw new MyException("비밀번호 불일치");
}
if(member.getName().equals("")) {
  throw new MyException("이름을 입력하세요.");
}
String emailReg = "[a-zA-Z0-9+\\-\\.]+@[a-zA-Z0-9\\-\\.]+\\.([a-zA-Z]{2,6})";
if(!Pattern.matches(emailReg, member.getEmail())
throw new MyException("이메일 형식 불일치");
}
try {
  // 회원가입 시 입력한 비밀번호를 암호화 하기 위해 랜덤해서 salt 생성
  String salt = UUID.randomUUID().toString();
  System.out.println("salt : " + salt);
  // salt와 랜덤 생성된 비밀번호를 salt와 함께 암호화
  String hashPassword = OpenCrypt.encrypt(member.getPassword(), salt);
  member.setPassword(hashPassword);
  byte[] secKey = OpenCrypt.generateKey("memberDAO.regist(member);
  ...

```

Service

```

public String getId() {
  return id;
}
public void setId(String id) {
  if(id != null && !id.trim().equals("")) {
    this.id = id;
  }
}
public String getPassword() {
  return password;
}
public void setPassword(String password) {
  this.password = password;
}
public String getName() {
  return name;
}
public void setName(String name) {
  if(name != null && !name.trim().equals("")) {
    this.name = name;
  }
}
public String getEmail() {
  return email;
}
public void setEmail(String email) {
  if(email != null && !email.trim().equals("")) {
    this.email = email;
  }
}

```

VO

```

<?xml encoding="UTF-8"?>
<mapper
  xmlns="http://mybatis.org/dtd/mybatis-3-mapper.dtd"
  namespace="com.ozary.trip.model.dao.MemberDAO">
  <insert id="login" parameterType="MemberVO" resultType="int">
    SELECT name FROM member WHERE id=#{id} AND password=#{password}
  </insert>
  <insert id="register" parameterType="MemberVO">
    INSERT INTO member(id, password, email, name, email)
    VALUES (#{id}, #{password}, #{email}, #{name}, #{email})
  </insert>
  <insert id="insertProfileImg" parameterType="String">
    UPDATE member SET profile_img=#{profileImgPath}
  </insert>

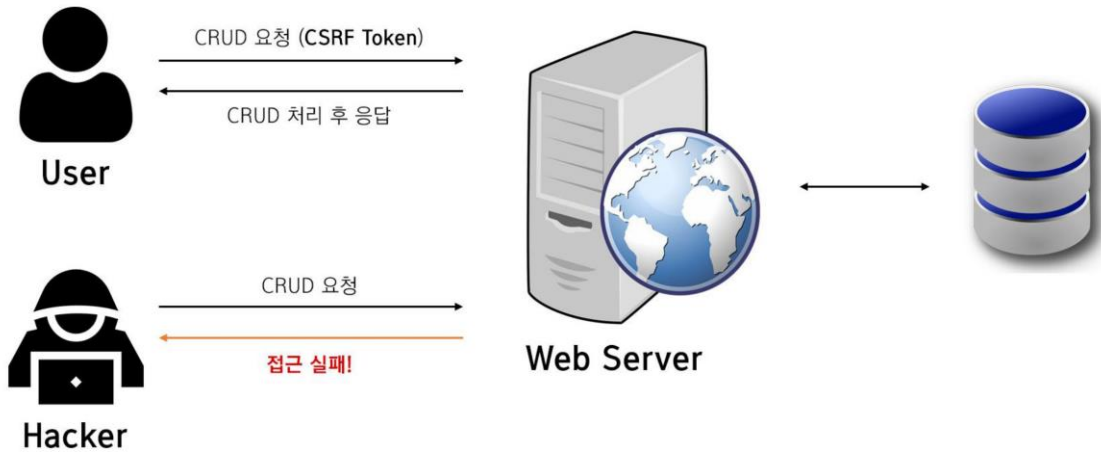
```

Mybatis



보안

CSRF 검사





보안

Session 탈취 검사



Request Header

Request Headers	
:Authority:	gist-queue-consumer-api.cloud.gist.build
:Method:	OPTIONS
:Path:	/api/v1/users
:Scheme:	https
Accept:	*/
Accept-Encoding:	gzip, deflate, br
Accept-Language:	ko-KR,ko;q=0.9,en-GB;q=0.8,en;q=0.7,en-US;q=0.6
Access-Control-Request-Headers:	content-type,x-cio-datacenter,x-cio-site-id,x-gist-encoded-user-token
Access-Control-Request-Method:	POST
Origin:	https://www.notion.so
Referer:	https://www.notion.so/
Sec-Fetch-Dest:	empty
Sec-Fetch-Mode:	cors
Sec-Fetch-Site:	cross-site
User-Agent:	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36



Session

key	value
***	***
userCheck	request.getHeader("user-agent")



보안

Session 중복 검사

ID: **SSAFY1**



ID: SSAFY2



ID: SSAFY3



ID: **SSAFY1**

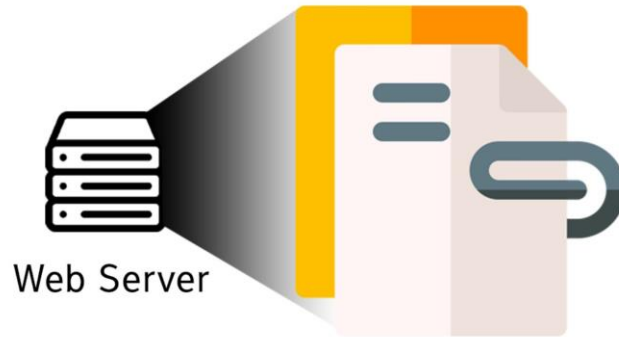


같은 ID의 JSESSIONID 중복 제거



보안

웹쉘 공격 방지



Web Server

`chmod -x`

Thanks! 

5. 프로젝트 핵심

1. 거리를 이용한 기능들을 추가 (자신과 거리가 가까운 게시물 순...)
2. CSRF, 세션탈취, FileUpload, 비밀번호 해쉬, SALT 와 같은 보안기능 추가
3. 세션 중복문제를 해결하기위한 기능 추가

6. 개발 후기

팀장(박승수)	<p>프로젝트에 필요한 CRUD 기능은 당연히 중요하지만, 그 외적으로 보안과 같은 기능들 또한 중요하다고 생각하며 프로젝트를 진행하였습니다.</p> <p>또한 session 중복 문제를 해결하면서 ConcurrentHashMap 을 사용하여 중복을 해결하였지만, 초당 만 건 정도의 요청이 들어갈 경우 동시성 문제가 발생한다는 것을 알았습니다.</p> <p>이를 해결하기 위해 다양한 고민을 하다 synchronized를 사용하는 방법으로 해결을 하였지만. 성능을 해치기 때문에, 더 좋은 로직은 없을지 고민을 더 해보고 싶어졌습니다.</p> <p>이처럼 깊게 공부할수록, 게시판 하나를 만드는 것도, 간단한 것이 아닌 점점 더 어려워졌고, 아는 만큼 보인다는 이야기를 깊게 공감하게 되는 시간이었습니다.</p>
팀원(유영)	<p>주로 개발하던 백 단 뿐만이 아니라 프론트 단의 데이터 노출 및 제어에 대해서 이해할 수 있는 시간이 되었습니다. 기능적인 부분에 중점을 둔 개발 후 발표 자료 제작 중에 느낀 점은, 기능적으로 작동하는 유저 접근 코드보다 비기능적인 기술이 추가된 코드가 더욱 복잡하고 고민을 많이 해야한다는 것이였습니다. 해당 부분에 힘을 쏟지 못한 점이 아쉽습니다. 페어프로젝트를 진행하면서 페어와 함께 웹 보안에 대해 고민하며 HTTP의 취약점 및 보안 강화 방법에 대해 고안해볼 수 있는 좋은 경험이 되었습니다. 다음엔 이 프로젝트를 유지보수하며 서버 혹은 데이터 공격의 강화, CRUD 실행 전 유효성 검사에 대한 고도화 작업을 하는 것도 도움이 될 것 같습니다.</p>