

나만의 자소서 관리 Spec overflow

Open API를 활용한 빅데이터 전처리시각화


미니프로젝트 1조

2020-02-06 ~ 2020-03-18





Contents

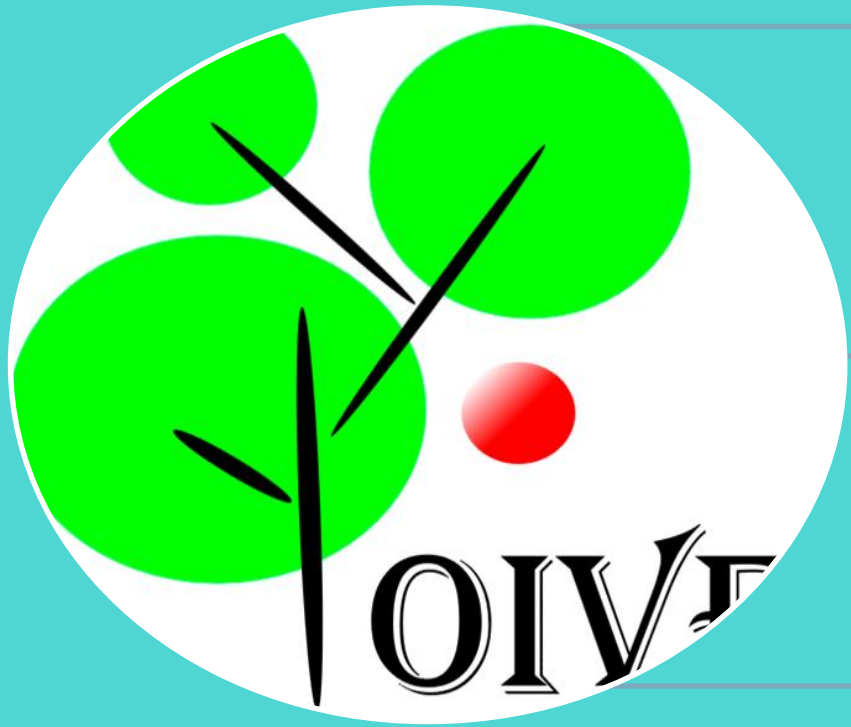
- ✕ 프로젝트 개요
 - 구현 기능
 - ✕ 시연
 - 후기
- 



프로젝트 개요

프로젝트 개요: 팀 구성

팀명 : oive



ONE team one spirit!!
성경의 Eve와 같은 창조의 근원을 상징합니다.



하지윤

- 프로젝트 형상관리
- 작업일정 관리
- DB 구축 및 관리
- 로그인, 파일업로드, 자기소개서 재활용 기능 등



김소연

- 회의록 작성
- 프로필 조회/수정/삭제



이영기

- 메인페이지 디자인
- 자기소개서 작성 기능 구현

프로젝트 개요 : 프로젝트 목표



사이트명

- spec overflow : 미래를 여는 넘치는 열정



목표

- 나만의 자기소개서를 작성할 수 있는 홈페이지



주요 기능

- 나의 스펙을 빠짐 없이 정리
- 기존 자소서 잘 정리하여 copy & paste로 빠르게 신규 자소서 작성



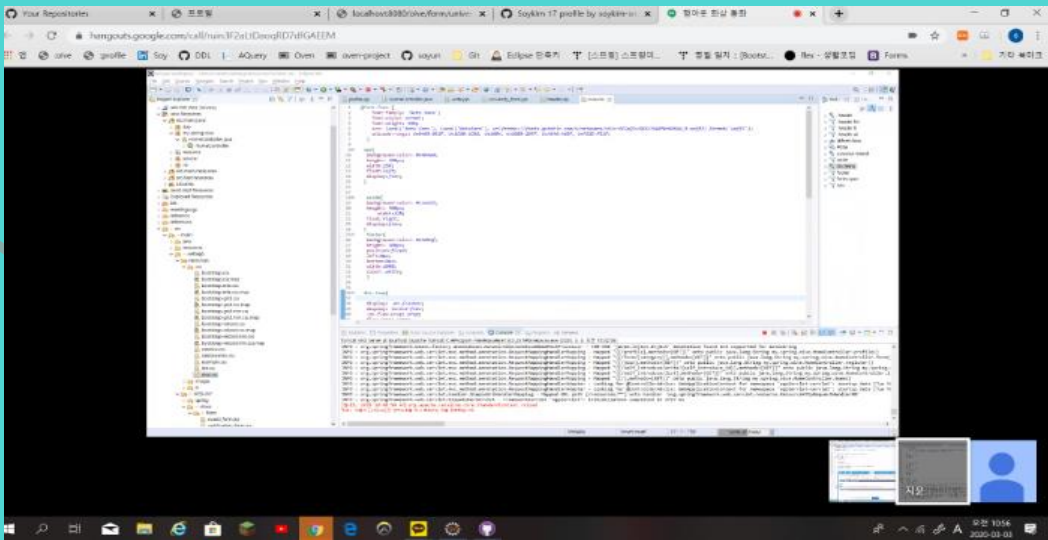
목적

- 신속한 입사지원으로 취업 성공!

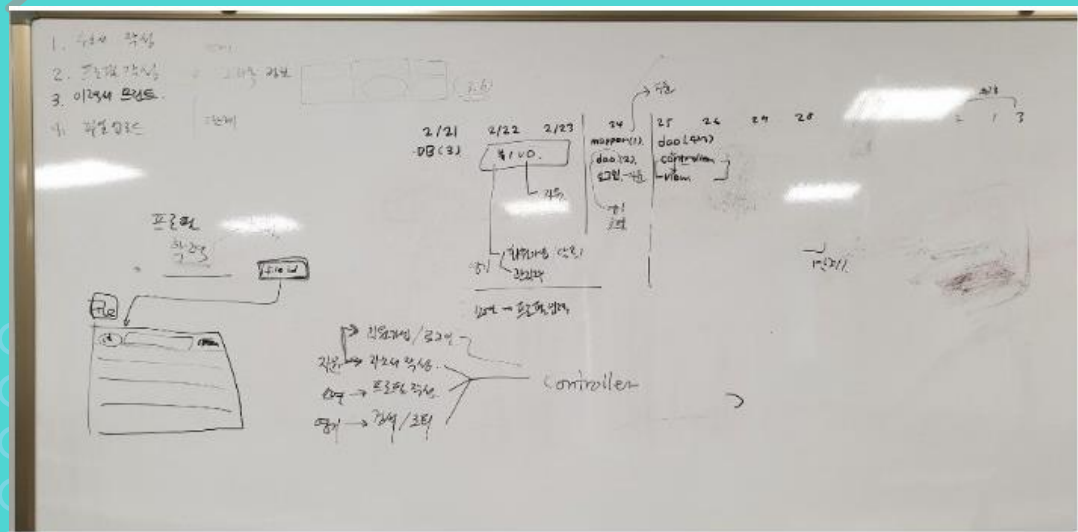
Spec overflow : presented by oive

개발 일정

단계	주요 활동	2월 2주	2월 3주	2월 4주	2월 5주	3월 1주	3월 2주	3월 3주
		6 7	10 11 12 13 14	17 18 19 20 21	24 25 26 27 28	2 3 4 5 6	9 10 11 12 13	16 17 18 19
기본 계획	주제 설정							
	참고 자료 조사							
	시스템 설계							
구축 단계	DB 정의							
	로그인 기능							
	디자인 작업							
	프로필 작성 기능							
	자소서 작성 기능							
	서브 기능 구축							
완료 단계	시스템 테스트							
	완료보고서 작성							



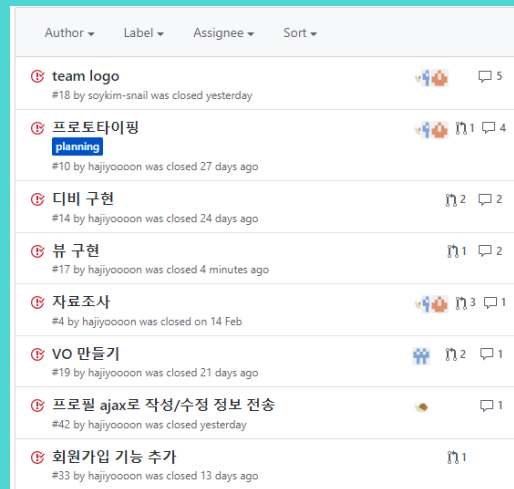
온라인 원격회의



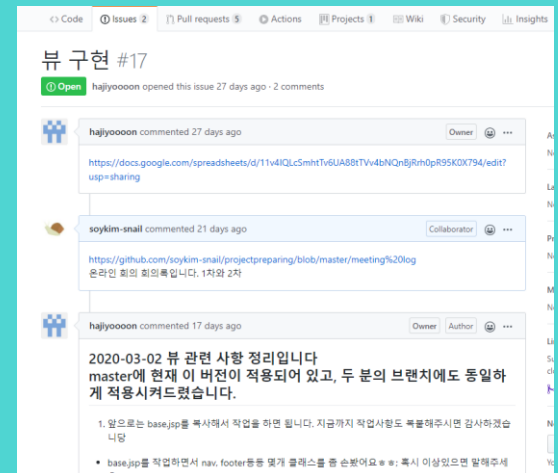
작업일정 회의



Kakao oven 프로토타입 협동 작업



Github 협업



프로젝트 작업 모습

사용 기술

Front-end



- HTML5
- JavaScript (AJAX)
- CSS (Bootstrap 4)

Server/DB

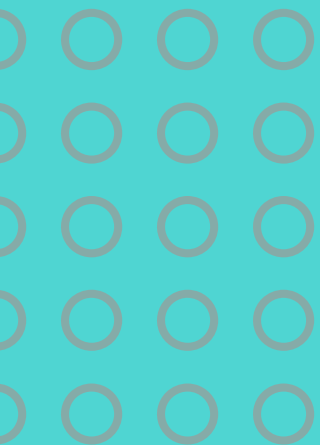


- Apache tomcat 9.0
- Oracle 11g (sql development)
- Spring framework (5.0.2)
- Mybatis, Spring MVC
- JAVA SE 8.0

Project tools

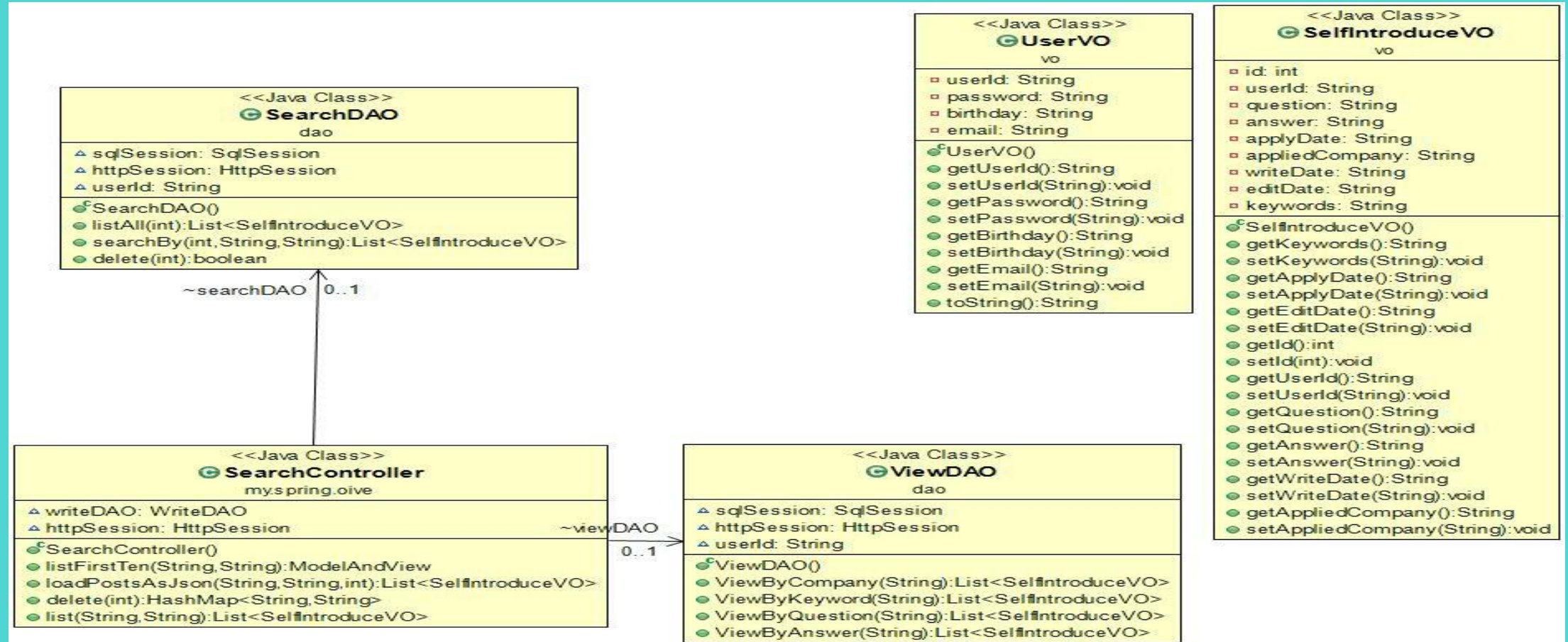


- Eclipse IDE
- Github, Github desktop
- Google hangover
- Kakao oven

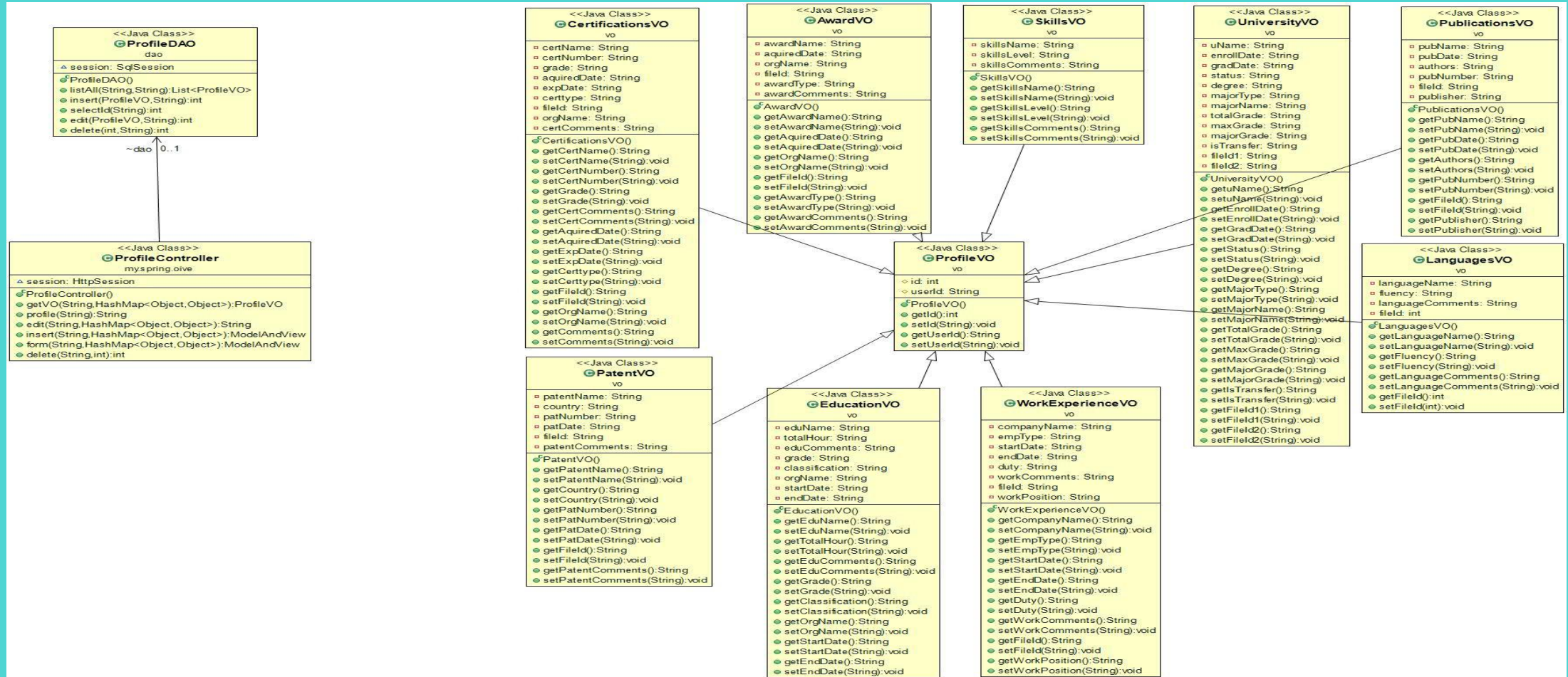


구현 기능

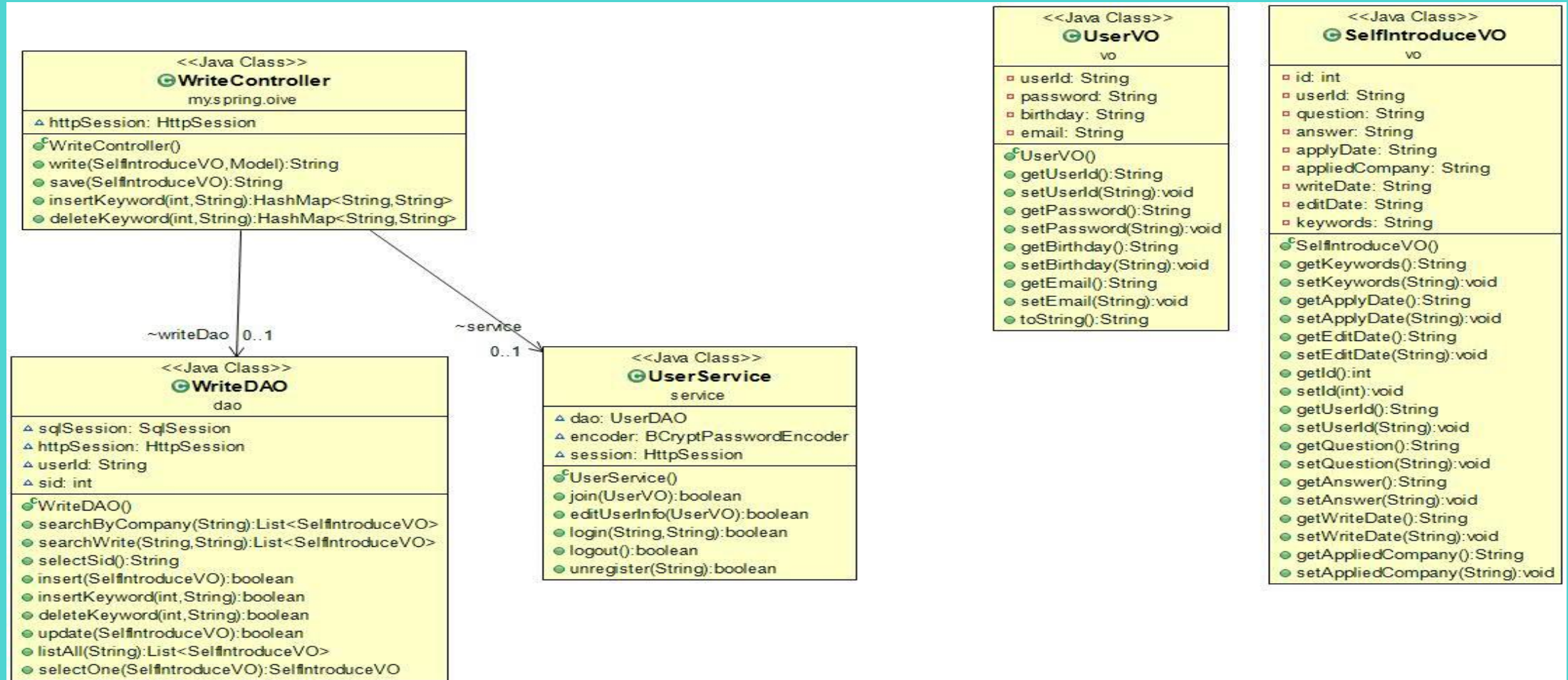
Class Diagram(user)



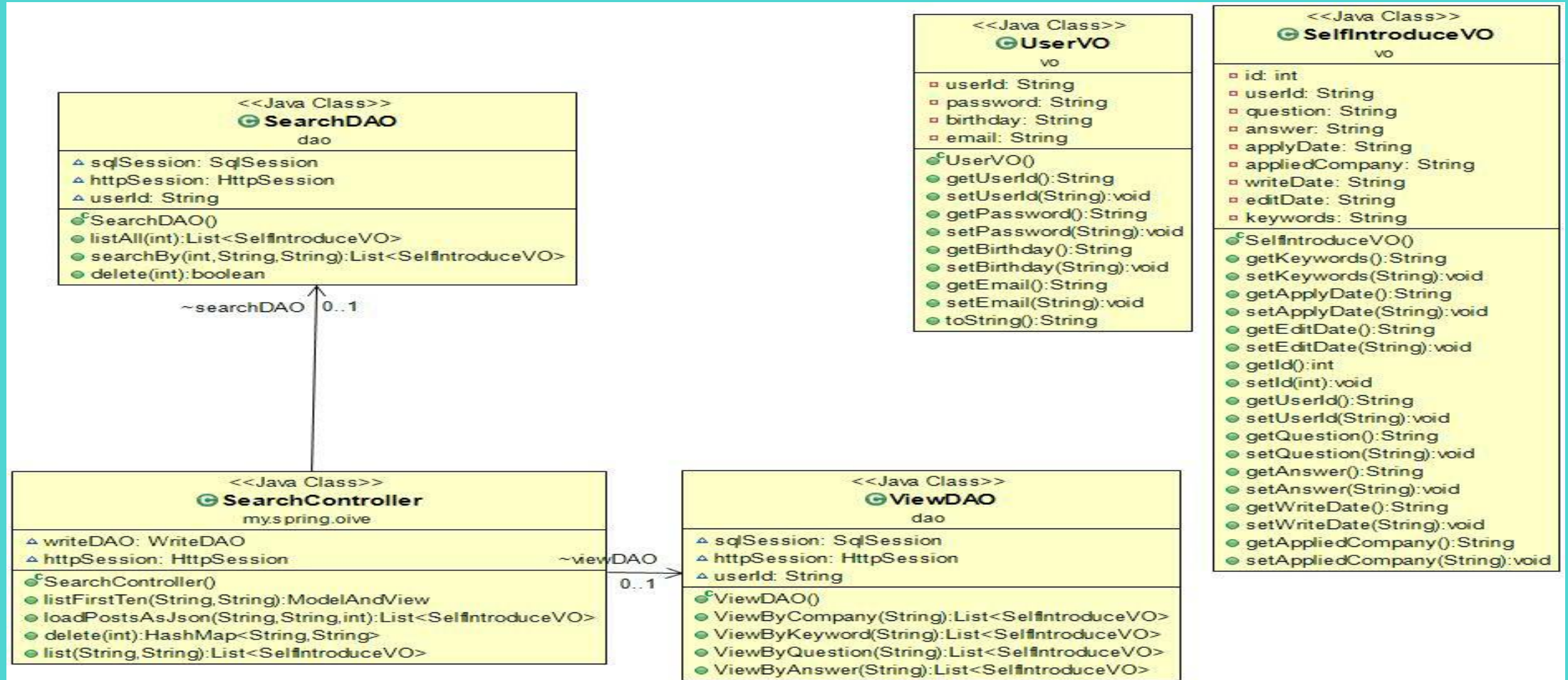
Class Diagram(profile)



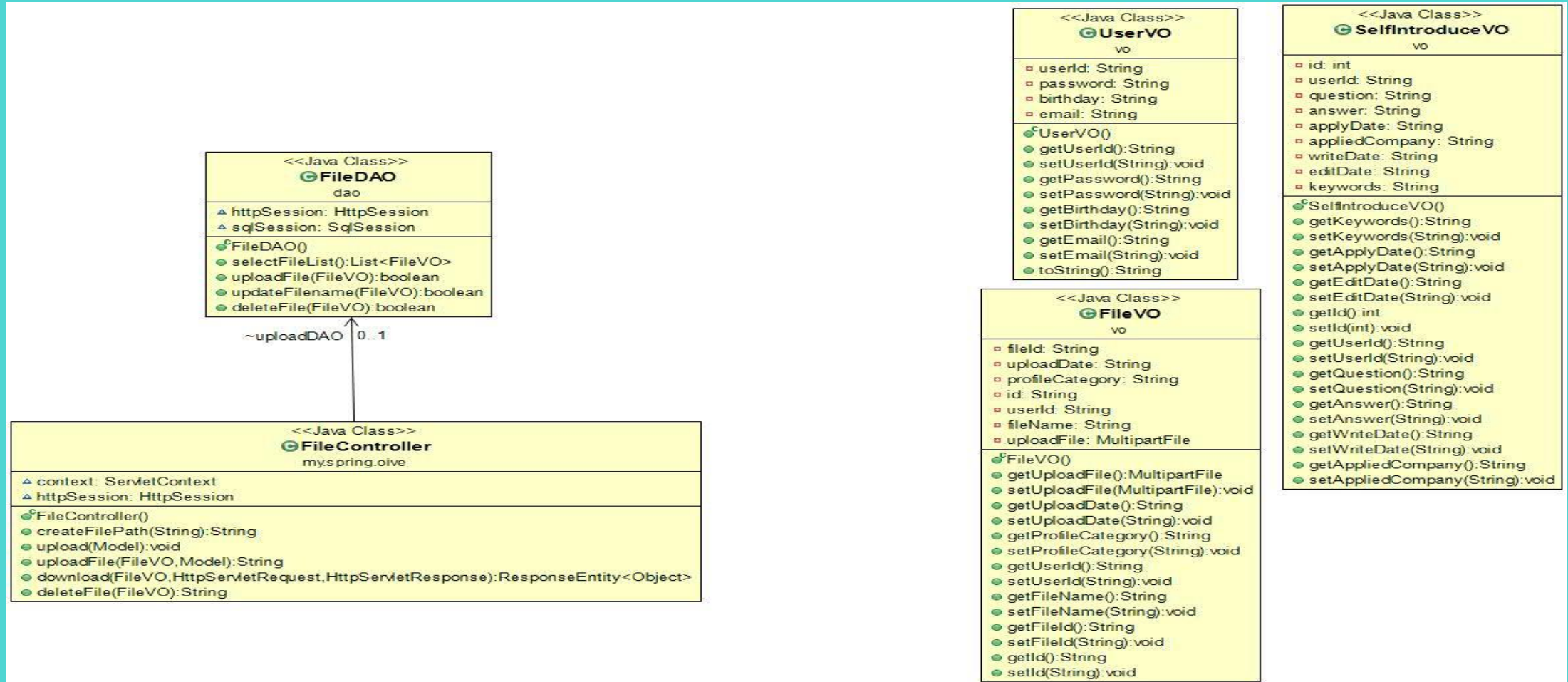
Class Diagram(write)



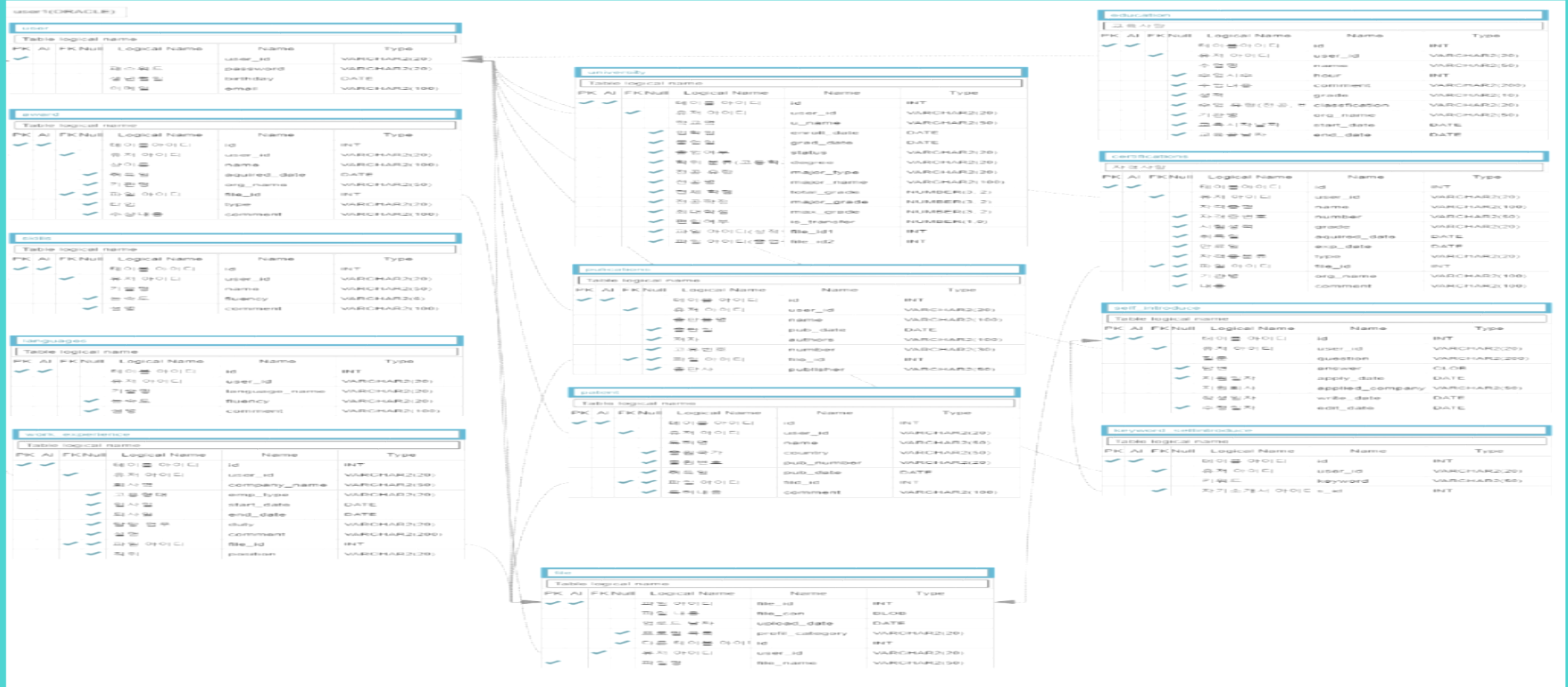
Class Diagram(view)



Class Diagram(file)



ER Diagram



기능 1

회원가입/로그인

기능



음... 우선 회원가입을 할 때
비밀번호는 암호화해서
저장되어야 할 것 같아요

```
@Autowired
```

```
BCryptPasswordEncoder encoder;
```

```
public boolean join(UserVO vo) {
```

```
    String encodedPassword = encoder.encode(vo.getPassword());
```

```
    vo.setPassword(encodedPassword);
```

▲ BCryptPasswordEncoder를 이용한 패스워드 암호화

```
    return dao.insert(vo);
```

```
}
```

USERID	PASSWORD	BIRTHDAY	EMAIL
test	\$2a\$10\$1S6mXAFMqIk4Ne9cDpnkBuM...	96/03/22	abc@abc.com
abc	\$2a\$10\$nvO30/2RboGJDsg0y5dQC.t...	16/03/13	abc@abc.com
test2	\$2a\$10\$08o8NEzdviAFtVLoKeZ0ie1...	20/03/12	ewf

▲ DB에 암호화되어 저장된 모습

기능 1

회원가입/로그인

기능



로그인하면 적어도 **30분**동안 로그아웃이 안되면 좋겠구요

로그인하지 않은 사용자가 데이터를 볼 수 없도록 해주세요

```
<interceptors>
  <interceptor>
    <mapping path="/**"></mapping>
    <exclude-mapping path="/"></exclude-mapping>
    <exclude-mapping path="/join"></exclude-mapping>
    <exclude-mapping path="/resources/**"></exclude-mapping>

    <beans:ref bean="authenticationInterceptor"/>
  </interceptor>
</interceptors>
```

▲Servlet-context에 메인페이지, 회원가입 페이지를 제외한 모든 페이지에 인터셉터 등록

```
@Override
public boolean preHandle(
    HttpServletRequest request, HttpServletResponse response,
    Object handler) throws ServletException {

    if(session.getAttribute("user") != null) {
        session.removeAttribute("user");
    }

    return true;
}
```

◀ 인터셉터에서는 세션에 저장된 유저 정보가 없으면 메인 페이지로 리다이렉트

기능 2 프로필 작성 기능



내 “스펙”의 전체 내용을 한 화면에 보게 해 주세요.

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<div class="col-sm profile" style="display:inline-flex;flex-direction:column">
  <h2 style="">학력사항</h2>
  <h3 id="university">대학교</h3>
  <c:import url="/form/university"/>
  <div id="university-add"></div>

  <h3 id="education">교육사항</h3>
  <c:import url="/form/education"/>
  <div id="education-add"><!-- 이곳에 추가된 form이 붙음.--></div>

  <h2 id="certifications">자격증</h2>
  <c:import url="/form/certifications"/>
  <div id="certifications-add"><!-- 이곳에 추가된 form이 붙음.--></div>
</div>
```

spec overflow : [프로필](#) [자기소개서](#) [이력서만들기](#) [파일업로드](#) [내정보](#) [로그아웃](#)

학력 사항

대학교

교육사항

자격증

외국어

기술

수상이력

특허

논문·저서

경력

학력사항

대학교

학교명(필수입력)

코넬대학교

학과명

컴퓨터사이언스

전공 유형

단일전공

입학일

2006-01-03

종 학점

4.3

관련파일

업로드

파일삭제

학사

졸업(예정)일

2020-03-01

전공 학점

4.3

최고학점

4.7

편입

졸업

수정

삭제

추가

교육사항

기관명

교육기관명(필수입력)

교육시간

수업 종료(예정)일

교육유형

성적

교육기관명(필수입력)

수업명/또는 강좌명을 입력하세요.

교육시간

수업 시작일

연도-월-일

수업 종료(예정)일

연도-월-일

교육유형

선택하세요.

성적을 입력하세요.

JSTL(Jsp standard tag library)의 ‘import’ 기능을 사용하여, 여러 jsp파일에 작성되는 9개의 ‘스펙’ 항목들을 한 화면으로 불러온다.

기능 2 프로필 작성 기능



“스펙” 수정할 때 마다 페이지가 리로딩 되는 건 싫어요.

그러나 수정할 때 마다 데이터베이스에는 실시간 저장해 주셔야 해요.

```
function edit(id){
    var formElement = document.getElementById("form_"+id);
    var formData = new FormData(formElement);

    var xhr= new XMLHttpRequest();
    xhr.onload=function(){
        if(xhr.status==200){
            window.alert("성공하였습니다: "+ xhr.responseText);
        } else
            window.alert("실패하였습니다: "+ xhr.responseText);
    }
    xhr.open("POST", "/oive/edit", true);
    xhr.send(formData);
}
```

HTML의 기본적인 form submit 기능 대신에 Ajax 통신을 사용한다.
사용자가 정보를 수정/삭제/추가해도 페이지는 리로딩되지 않는다. DB는 즉시 수정된다.

기능 3 자기소개서 작성 기능(ver1)



자기소개서를 작성할 때 자기소개서를 대표하는 키워드를 입력할 수 있게 해주세요

▼키워드는 콤마(,)로 입력받아 컨트롤러에서 파싱하여 **DB**에 각각 입력한다.

```
String keyword = vo.getKeywords()
String[] array = keyword.split(",")

else if(action.equals("insert")) {
    boolean result = dao1.insert(vo)
    result = dao.insertKeyword(array)

    if(result) {
        model.addAttribute("msg", vo.getId()+"의 글이 입력되었습니다.")
        model.addAttribute("list", list)
    }else {
        model.addAttribute("msg", vo.getId()+"의 글이 입력에 실패했습니다.")
    }
}

return "write"
```

기능 3 자기소개서 작성 기능(ver2)



“저장”버튼을 누를 때마다 조회 화면으로 돌아가는게 불편하네요

ㅠ.ㅠ

▶
프론트에서도 AJAX 비동기 통신을 적용
하여 저장 후에도 페이지 로딩을 방지

```
@RequestMapping(value = "/self_introduce/save", method = RequestMethod.POST,  
    produces="application/json; charset=utf-8")  
@ResponseBody  
public String save(@RequestBody SelfIntroduceVO vo) {  
    vo.setUserId(((UserVO)httpSession.getAttribute("user")).getUserId());  
    boolean result = writeDao.update(vo);  
    return String.format("{ 'result' : '%s' }", result);  
}
```

▲ 컨트롤러의 응답 방식을 JSON으로 변경

```
var mySubmit = document.querySelector("input[type=submit]");  
mySubmit.addEventListener("click", (e) => {  
    var xhr = new XMLHttpRequest();  
  
    var data = {  
        'id': myForm.querySelector("input[name=id]").value ,  
        'question': myForm.querySelector("input[name=question]").value ,  
        'answer': myForm.querySelector("textarea[name=answer]").value ,  
        'appliedCompany': myForm.querySelector("input[name=appliedCompany]").value ,  
        'applyDate': myForm.querySelector("input[name=applyDate]").value  
    }  
  
    var jsonString = JSON.stringify(data);  
    console.log(jsonString);  
  
    xhr.open("POST", "save");  
    xhr.setRequestHeader("Content-Type", "application/json");  
    xhr.send(jsonString);  
})
```

기능 3 자기소개서 작성 기능(ver3)



저기 지금 키워드가 중복해서 들어
가는데요... 중복 안되게 해주세요

^^, ^^, ^^....., ; ; ; ; ; ; ; ; ; ;



기능 3 자기소개서 작성 기능(ver3)



사용자의 입력을 태그 형식으로 예쁘게 바꿔 주는 오픈소스

태그 입력, 삭제가 가능하고,
“중복 태그 방지” 기능까지 제공하는 착한 친구

```
var myInput = $('[name=tags]').tagify();
myInput.on('add', function(e, tagName){
  var xhr = new XMLHttpRequest();
  xhr.onload = function(e){
    ...
  };
  xhr.open('GET', 'write/keyword?sid=${ vo.id }&keyword='+tagName.data.value);
  xhr.send();
}),
```

```
if(vo.getId() = 0) {
  String sid = writeDao.selectSid(); // 1. 자기소개서 id 찾기
  if(sid != null) {
    vo.setId(Integer.parseInt(sid)); // 2. 받은 id로 vo.setId()
    result = writeDao.insert(vo);
    uri = "redirect:write?id=" + vo.getId();
  }
}
else {
  vo = writeDao.selectOne(vo);
  if(vo != null) result = true;
}
```

▲ 새 자기소개서를 작성하는 경우 해당 자기소개서에 id를 부여하여 반환하고, 수정하는 경우 기존 자기소개서를 DB에서 받아오는 방식으로 자소서를 생성하는 순간 id가 부여되도록 함

◀ Tagify 에서 제공하는 on 메소드를 이용.
키워드가 추가될 때마다 비동기 통신으로 추가된 태그를 전달한다

기능 4 파일함 기능



성적증명서, 졸업증명서 등을 홈페이지에 올려서 관리하고 싶어요

올린 파일을 다운로드할 수 있으면 좋겠어요

```
public class FileVO {  
    private String fileId;  
    private String uploadDate;  
    private String profileCategory;  
    private String id;  
    private String userId;  
    private String fileName;  
    private MultipartFile uploadFile;  
}
```

FileVO 객체에서
MultipartFile 를 이용해
◀ 입력된 데이터를 처리

업로드 부분은 AJAX로 처리하여, ▶
전송 성공 시 페이지를 재로딩하고
실패할 시 메시지를 출력

```
function upload(){  
    var xhr = new XMLHttpRequest();  
    var data = new FormData(document.getElementById("f"));  
    xhr.onload=function(e){  
        var str = xhr.responseText;  
        var result = JSON.parse(str);  
        alert(result['msg']);  
    }  
    xhr.onerror=function(e){alert(e);};  
    xhr.open('POST','/oive/upload',true);  
    xhr.send(data);  
};
```


기능 4 파일함 기능



올린 파일을 다운로드할 수 있으면
좋겠어요

파일 로드 실패 시의 응답과 status code를 지정 ▶

응답 헤더와 바디를 직접 설정할 수 있는 ▶
ResponseEntity 타입으로 리턴

```
@RequestMapping(value="/download")
@ResponseBody
public ResponseEntity<Object> download(FileVO vo, HttpServletRequest request, HttpServletResponse headers) {
    HttpHeaders headers = new HttpHeaders();
    HttpStatus status = null;
    Object body = null;
    headers.setContentType(MediaType.parseMediaType("application/octet-stream"));
    headers.setContentDisposition(ContentDisposition.builder("attachment").filename(vo.getFilename(), Charset.forName("UTF-8")).build());

    try(InputStream in = new FileInputStream(file);) {
        headers.setContentLength(file.length());

        byte b[] = new byte[(int) file.length()];
        while ((in.read(b)) > 0) {}

        body = b;
        status = HttpStatus.OK;
    } catch (IOException e) {
        e.printStackTrace();
        headers.setContentType(new MediaType("application", "json", Charset.forName("UTF-8")));
        result = false;
        status = HttpStatus.INTERNAL_SERVER_ERROR;
    }

    ResponseEntity<Object> entity = new ResponseEntity<Object>(body, headers, status);

    return entity;
}
```

기능 4 파일함 기능



올린 파일을 다운로드할 수 있으면
좋겠어요

서버에서 데이터 전송에 성공하면 ▶
가상 URL을 생성해서 클릭하는 것으로
파일 다운로드 기능을 구현

서버에서 데이터 전송에 실패하면 ▶
Status code 에 따라 적절한 오류 메시지를 출력

```
function downloadFile(e, fileName){
    var xhr = new XMLHttpRequest();
    xhr.onload = function(){
        if(xhr.status == 200){
            if(true){
                var link = document.createElement('a');

                link.href = window.URL.createObjectURL(new Blob([xhr.response]));
                link.download = fileName
                link.click();
                link.href.revokeObjectURL();
                link.remove();
            }
        }
        else if(xhr.status == 400){
            window.alert("잘못된 인코딩 방식을 적용하여 파일을 다운로드할 수 없습니다.");
        }
        else if(xhr.status == 500){
            window.alert("파일을 읽는 중 오류가 발생했습니다. 다시 한 번 시도해주세요.");
        }
        else if(xhr.status == 403){
            window.alert("파일을 찾을 수 없습니다");
        }
    }
    xhr.responseType = 'blob';
    xhr.open('GET', 'download?fileName=' + fileName, true);
    xhr.send();
}
```

부가기능 1 야간모드

```
<html class="darkmode">
```

```
...  
<div class="col-sm">  
  <a href="javascript:toggleDarkmode();">자소서로 흑사한 눈을 편안하게</a>  
</div>
```

```
function toggleDarkmode(){  
  var html = document.querySelector("html");  
  if(html.className=="darkmode"){  
    html.removeAttribute("class");  
  }  
  else{  
    html.setAttribute('class', 'darkmode');  
  }  
}
```

◀Html의 클래스 속성을 **"darkmode"**로 부여하고
이를 제어할 수 있도록 **javascript** 이용

CSS에서 **.darkmode** 속성의 하위 클래스 별로
야간모드 작동 시 스타일을 지정▼

```
.darkmode .form-control {  
  display: inline-block;  
  width: 100%;  
  height: calc(1.5em + 0.75rem + 2px);  
  padding: 0.375rem 0.75rem;  
  font-size: 1rem;  
  font-weight: 400;  
  line-height: 1.5;  
  background-clip: padding-box;  
  border-radius: 0.25rem;  
  transition: border-color 0.15s ease-in-out, box-shadow 0.15s ease-in-out;  
  color:white;  
  background-color:black;  
  border: 1px solid #292929;  
}
```

시연

- <http://70.12.115.166:8000/oive>

Retrospect!

지윤



욕심 많은 저때문에ㅠㅠ 덩달아 고생하신 소연언니랑 영기씨 수고많으셨어요
ㅎㅎ

영기

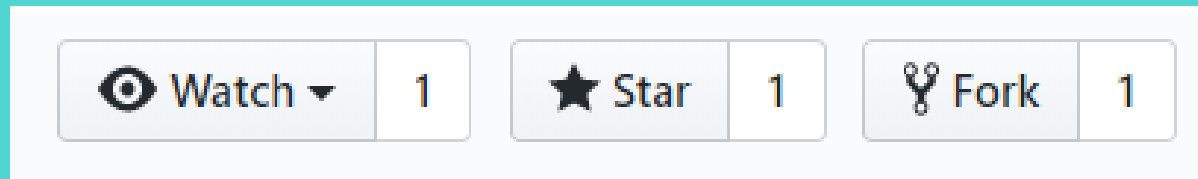
처음에는 저는 저희 조 아이디어를 게시판을 이용해서 할 수 있는 간단한 미니 프로젝트라고 생각을 했습니다. 하지만 생각보다 간단하지 않았고, 생각한 거 이상으로 어려워서 조원들에게 많은 도움을 받았습니다. 그래도 도움을 받으며 오류를 찾고 과정을 진행해보니 저한테는 도움이 많이 되었던 시간이었습니다. 소연누나, 지윤씨 모두 다 고생하셨습니다. ㅎㅎ

소연

코로나바이러스로 인한 휴강!! 혼자 개발환경을 갖추고 작업 하라!! 나의 문제해결 능력의 민낯과 마주하는 고통스러운 시간이었어요. 원격회의를 함께하고 인강과 포스팅을 공유해준 동료 감사해요. 학원을 떠나 홀로서기를 하는데 큰 자양분이 될거라 생각해요.

부족함을 많이 느꼈고, 포기하고 싶은 순간들도 있었습니다. 그러나 “Done better than perfect!” 함께해준 팀원들에게 감사인사 드려요~!

Thank you!



<https://github.com/hajiyooooon/oive>



```
.click = function(){  
    message.send("Good Job!");  
}
```

