

# 個人ポートフォリオ

KOH YEON WOO  
(コ・ヨンウ)



## 技術力



2018年5～7月

SMART Cloud IT-Master  
CUP課程

Java

修了

2018年9月  
～2019年7月

SMART Cloud IT-Master

Pure Java、  
Web Component等

修了

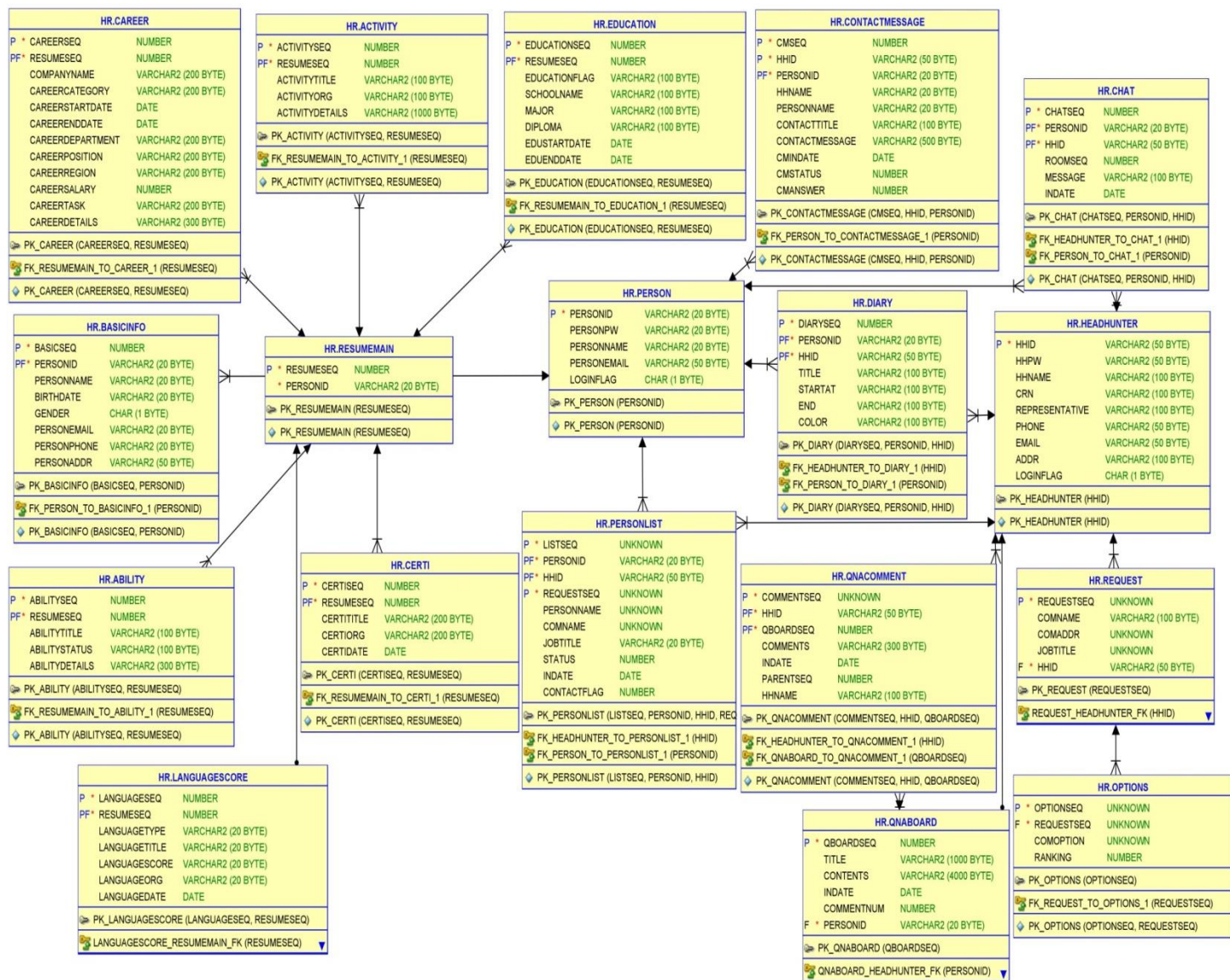
勉強している開発環境、IT言語 :



## チームプロジェクト

❖ プロジェクト名	ヘッドハンティング
❖ プロジェクト期間	2019. 3. 4 ~ 2019. 4. 19
❖ 開発環境・技術	JAVA, ORACLE DB, MYBATIS, APACHE TOMCAT HTML, CSS, JAVASCRIPT, JQUERY, JSP, SPRING, AWS
❖ 担当作業	<ul style="list-style-type: none"><li>✓ プロジェクト工程管理（リーダー）</li><li>✓ アイディア提案</li><li>✓ DB設計</li><li>✓ Login、掲示板、オファー機能、関心人材管理、コンタクト人材登録、履歴書登録、修正のバックエンド作業</li></ul>

# チームプロジェクト-DB設計図



✓ 履歴書を管理するため、  
カテゴリー別テーブルを分離

✓ RESUMESEQというシークエンスで  
連結

# チームプロジェクト-担当機能説明

## ✓ ログインページ

ログインする時、  
DBのLOGINFLAG  
カラムを検索して  
移動する画面を区分

HOME LOGIN COMPANY PRICE

### LOGIN

PERSONE MEMBER OR COMPANY MEMBER

www.perfectten.website

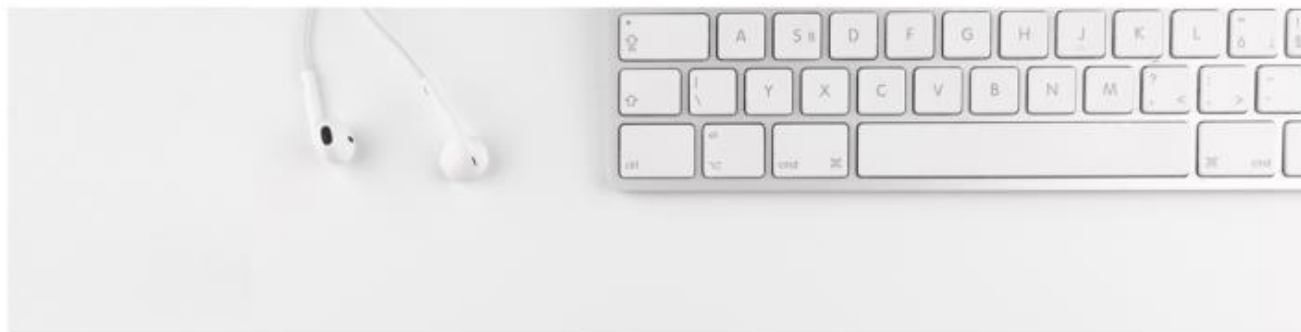
ID User

PW Password

Person Company

Sign Login

会員加入機能



# チームプロジェクト-担当機能説明

## ✓ ヘッドハンターページ

オファー登録機能

コンタクトされた  
人材管理機能

気に入る人材管理機能

The screenshot shows a web interface for a headhunter. At the top, there's a banner with the text "GOOD RELATIONSHIP". Below the banner, there are navigation tabs: "SEARCH PERSON", "REQUEST", "PROGRESS", "FAVORITES", "CALENDAR", and a "LOGOUT" button. The main content area displays a table of job offers. To the right of the table, there's a sidebar with a "구직자 목록" (Job Seeker List) section and a "기업해드윈터 검색" (Company Headhunter Search) section. At the bottom of the sidebar, there are four buttons: "화상채팅" (Video Chat), "계시글가기" (Go to Post), "채팅하기" (Chat), and "메시지전송" (Send Message).

번호	제목	게시자	등록일자
283	이직 관련해서 문의드립니다. [0]	1155	2019-05-01 08:40:12
263	이직을 원합니다. [0]	human	2019-05-01 06:27:23
244	이직에 대해서 문의 드릴께 있습니다. [1]	Edennetic	2019-04-28 07:16:31
243	안녕하세요 이직에 대해서 문의 드릴께 있습니다 [1]	Edennetic	2019-04-28 07:14:13
223	이직을 원합니다 [0]	yw	2019-04-23 00:29:30
222	aa [0]	aaa	2019-04-22 23:56:28
206	안녕하세요 고민이 있습니다.11 [0]	sds	2019-04-22 22:03:08
63	이직을 하고 싶은데 방법을 잘 모르겠습니다. [17]	Tociddi	2012-09-15 00:00:00
61	재 취업이 어렵습니다. [0]	Newbigis	2017-09-12 00:00:00
60	헤드헌팅 관련 문의 드립니다. [0]	Newbigis	2019-01-08 00:00:00

12345next

ヘッドハンターが人材の質問に  
書き込みを書ける機能

# チームプロジェクト-担当機能説明

## ✓人材ページ



A: 履歴書登録

B: 履歴書修正 (削除と登録で修正具現)

C: 헤드헌터からのめメッセージを見せるページ



# チームプロジェクト-Point機能

## ✓ オファー登録具現

REQUEST

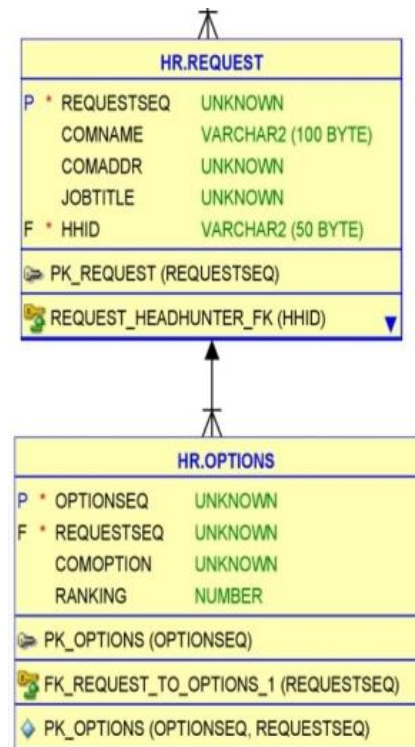
기업오ffer등록

기업 이름

삼성 sds	서버 관리자	서울시 송파구 잠실1동	경력자
사원 이상	포직 600 이상	조건을 입력하세요.	조건을 입력하세요.

Close Register

OPTIONS



DB를分離してオファー의追加項目を五つまで登録する。



# チームプロジェクト-Point機能

## ✓履歴書登録具現

com.sample.test.VO.resume

- > Ability.java
- > Activity.java
- > BasicInfo.java
- > Career.java
- > Certificate.java
- > Education.java
- > LanguageScore.java
- > TotalResume.java

①履歴書具現のため、VO (DTO) をカテゴリー別で分離



②一つのVO (DTO) に履歴書登録に必要な全てのデータを入れて「List」タイプで入力する。

③「for」文と「toString」を使ってDBに入力

```
public class TotalResume {  
  
    private int resumeSeq;  
    private String personId;  
  
    private int basicSeq;  
    private String personName;  
    private String birthDate;  
    private String gender;  
    private String personEmail;  
    private String personPhone;  
    private String personAddr;  
  
    // Ability  
    private int abilitySeq;  
  
    private List<String> abilityTitle;  
    private List<String> abilityStatus;  
    private List<String> abilityDetails;  
  
    private ArrayList<Ability> abilityList;  
  
    // Activity  
    private int activitySeq;  
  
    private List<String> activityTitle;  
    private List<String> activityOrg;  
    private List<String> activityDetails;  
  
    private ArrayList<Activity> activityList;  

```

```
List<String> abilityNum=totalResume.getAbilityTitle();  
  
if (abilityNum!=null) {  
  
    ArrayList<Ability> abList=new ArrayList<Ability>();  
    totalResume.setAbilityList(abList);  
  
    for(int i=0;i<abilityNum.size();i++) {  
  
        String title=totalResume.getAbilityTitle().get(i);  
        String status=totalResume.getAbilityStatus().get(i);  
        String detail=totalResume.getAbilityDetails().get(i);  
  
        Ability ability=new Ability();  
  
        ability.setResumeSeq(resumeSeq);  
        ability.setAbilityTitle(title);  
        ability.setAbilityStatus(status);  
        ability.setAbilityDetails(detail);  
        totalResume.getAbilityList().add(ability);  
  
    } // ability setting for  
  
    System.out.println("for 문 이후 totalResume : "+totalResume);  
  
    ArrayList<Ability> abilityList = new ArrayList<Ability>();  
    abilityList=totalResume.getAbilityList();  
  
    boolean flag = false;  
  
    for (int i = 0; i < abilityList.size(); i++) {  
        int abilityResult=dao.insertAbility(abilityList.get(i));  
  
        if(abilityResult==1) flag=true;  
        else flag=false;  
    }  
  
    System.out.println("최종 결과 : "+flag);  
  
} else {  
    System.out.println("ability 속성 없음");  
} //abilityNum if else  
  
<insert id="insertAbility" parameterType="Ability">  
    INSERT INTO ABILITY(  
  
        ABILITYSEQ  
        ,RESUMESEQ  
        ,ABILITYTITLE  
        ,ABILITYSTATUS  
        ,ABILITYDETAILS  
  
    )VALUES(  
  
        ABILITYSEQ.NEXTVAL  
        ,#{resumeSeq}  
        ,#{abilityTitle}  
        ,#{abilityStatus}  
        ,#{abilityDetails}  
  
    )  
</insert>
```



ありがとうございました。