

# Project Task

NOTE!! It is NOT necessary that all the following functionality is implemented. On the contrary it is important to choose what functionality to code. The requirement is to code just as much as is needed to implement all architectural solutions, AND NOT MORE.

## 1 Background

A company is recruiting staff for the coming season. To facilitate the recruitment process, they benefit from a web based recruitment tool. They hope to attract 15,000 applications during a two week period.

The company has for a long time used a recruitment system based on CGI scripts and a relational database. The system is shaky and some bugs have still not been fixed. As the company that built the original version no longer exists, consultants have been hired to fix bugs which have resulted in significant maintenance costs.

Over the years, many new requirements have emerged, which would bring major time savings for the company. Introducing new functionality into an already buggy system does however not attract the IT manager.

You have been asked to build a new, robust, scalable and well-documented system that can easily be expanded with new functionality. Since time is short (it always is) before this year's recruitment, the company is however satisfied if the new system provides the same functionality as the existing system, although an upgraded version is the goal.

## 2 The Recruitment System

The system distinguishes between two types of users, applicants and recruiters. An applicant applies for a job offered by the company while a recruiter manages applications. The system is divided into two parts: The registration of job applications and the administration of applications.

## **2.1 Registration of Applications**

An applicant is anyone who is interested in a job provided by the company. A web browser is used to register an application.

### **2.1.1 Current Functionality**

A user can register a job application that consists of the following components:

- Personal information - name, surname, date of birth and email address.
- Competence Profile - experience in different areas of expertise.
- Availability - periods of time the applicant can work.

### **2.1.2 Future Functionality**

The following functionality is desired in a coming version of the system:

- The online application form is available in several languages.
- It is possible to generate a PDF version of the application form.

## **2.2 Administration of job applications**

Recruiters administering job applications are employees of the company. A web browser is used to administer the applications. A user name and a password are required to access the service.

### **2.2.1 Current Functionality**

The following functionality is implemented in the current version of the system:

- The tool can list applications ordered according to different attributes.
- A job application may be rejected or accepted.

### **2.2.2 Future Functionality**

The following functionality is desired in a coming version of the system:

- A job application can be presented as a PDF document instead of an HTML page.
- Other internal systems can easily access information of the applications without directly connecting to the database.

- The system itself is able to select the job applications that are of interest. The decision making component is changeable by configuration. Different components can be combined.

### 3 Existing Database Model

The current system uses the following database script to generate tables.

```
CREATE TABLE role (  
  role_id BIGINT PRIMARY KEY,  
  name VARCHAR(255)  
);  
  
CREATE TABLE person (  
  person_id BIGINT PRIMARY KEY,  
  name VARCHAR(255),  
  surname VARCHAR(255),  
  ssn VARCHAR(255),  
  email VARCHAR(255),  
  password VARCHAR(255),  
  role_id BIGINT REFERENCES role,  
  username VARCHAR(255)  
);  
  
CREATE TABLE availability (  
  availability_id BIGINT PRIMARY KEY,  
  person_id BIGINT REFERENCES person,  
  from_date DATE,  
  to_date DATE  
);  
  
CREATE TABLE competence (  
  competence_id BIGINT PRIMARY KEY,  
  name VARCHAR(255)  
);  
  
CREATE TABLE competence_profile (  
  competence_profile_id BIGINT PRIMARY KEY,  
  person_id BIGINT REFERENCES person,  
  competence_id BIGINT REFERENCES competence,  
  years_of_experience NUMERIC(4,2)  
);
```

Unfortunately there is no documentation of the model, and the CGI script that operates on the database led only to confusion (and headache). It's up to you whether you want to use these tables, or parts thereof, in the new system. The important thing is that the data in the current database is not lost but instead transferred to the new system. You can run the following script to generate data that is used to simulate the data in the current system.

```
INSERT INTO role (role_id, name) VALUES (1, 'recruit');

INSERT INTO role (role_id, name) VALUES (2, 'applicant');

INSERT INTO person (person_id, name, surname, username, password, role_id)
VALUES (1, 'Greta', 'Borg', 'borg', 'wl9nk23a', 1);

INSERT INTO person (person_id, name, surname, ssn, email, role_id)
VALUES (2, 'Per', 'Strand', '19671212-1211', 'per@strand.kth.se', 2);

INSERT INTO availability (availability_id, person_id, from_date, to_date)
VALUES (1, 2, '2014-02-23', '2014-05-25');

INSERT INTO availability (availability_id, person_id, from_date, to_date)
VALUES (2, 2, '2014-07-10', '2014-08-10');

INSERT INTO competence (competence_id, name)
VALUES (1, 'Korvgrillning');

INSERT INTO competence (competence_id, name)
VALUES (2, 'Karuselldrift');

INSERT INTO competence_profile (competence_profile_id, person_id,
competence_id, years_of_experience)
VALUES (1, 2, 1, 3.5);

INSERT INTO competence_profile (competence_profile_id, person_id,
competence_id, years_of_experience)
VALUES (2, 2, 2, 2.0);
```

## 4 Other Requirements

In addition to the above requirements, the company has the following general requirements for any system they are using.

## **4.1 Logging**

All the system's main events are logged.

## **4.2 Browsers**

Any HTML that the company publishes on the Internet must be compatible with the following browsers:

- Internet Explorer 8 and later
- Firefox 20 and later
- Chrome 25 and later

## **4.3 Code Handover**

After the system is finalized the code is handed over to the company. With the help of its documentation the developers at the company's IT department can easily modify the system.

## **4.4 Prototype of the Web Service Client**

The company is looking to add a stand-alone client that communicates via web services. Therefore, you should develop a prototype of such a client, where one or more of the recruiter's use cases can be managed. Note that it must be possible to use the stand-alone client in parallel with the browser client.

# **5 Use Cases**

## **5.1 Register**

### **5.1.1 Actors:**

Applicant

### **5.1.2 Requirements:**

None

### **5.1.3 Scenario:**

1. The applicant enters first name, last name, email address, username and password.
  - (a) If a field is missing, the system returns the same form with an appropriate error message.

2. The system registers the newly created account.
3. The system displays a confirmation page.

## **5.2 Apply**

### **5.2.1 Actors:**

Applicant

### **5.2.2 Requirements:**

The applicant has logged in.

### **5.2.3 Scenario:**

1. The applicant chooses an area of expertise from a list, specifying years of experience in the area.
2. The system records the data and presents the specified areas of expertise.
3. Steps 2-3 are repeated until the applicant is satisfied.
4. The applicant specifies the availability periods.
5. The system records the data and presents the specified periods.
6. Steps 5-6 are repeated until the applicant is satisfied.
7. The system presents all the information the applicant has entered in steps 1-6.
8. The applicant chooses to hand in the job application or to cancel.
9. The system registers the job application.

## **5.3 Login**

### **5.3.1 Actors:**

Recruiter, Applicant

### **5.3.2 Requirements:**

The actor has requested a page requiring authentication without having logged in previously.

### **5.3.3 Scenario:**

1. The actor enters user name and password.
  - (a) If a field is missing, the system returns the same form with an appropriate error message.
  - (b) If the system can not authenticate the actor, it returns a blank form and tells that the login failed.
2. The system shows the originally requested page.

## **5.4 List Applications**

### **5.4.1 Actors:**

Recruiter

### **5.4.2 Requirements:**

The recruiter has logged in.

### **5.4.3 Scenario:**

1. The recruiter chooses to list a number of applications under selected search parameters. The following parameters can be selected: Time period, date of registration, competence, name.
  - (a) If the number of requested applications exceeds 1000, the system ignores the number and instead use the default value for number of applications per view.
2. The system returns a view showing the applications matching the selected search parameters. For each application is shown first name, surname and date of registration.
  - (a) If there are more applications matching the search parameters than are displayed in a single view, a "Show next page" alternative is presented.
  - (b) The recruiter may choose to display the next page (if any).
  - (c) The recruiter may choose to repeat step 1 to change search parameters.

## **5.5 Show Application**

### **5.5.1 Actors:**

Recruiter

### **5.5.2 Requirements:**

The recruiter has logged in and has listed applications.

### **5.5.3 Scenario:**

1. The recruiter chooses an application from the applications listed in the List Applications use case.
2. The system displays a full view of the job application and its status. The status is either accepted or rejected.
  - (a) The recruiter may choose to view the application as a PDF document.
  - (b) The recruiter may choose to change the application's status.
    - i. If the current application is being modified by another user concurrently, the system aborts the update and informs the recruiter why the action was aborted.
  - (c) The recruiter may choose to go back to view the listing all applications.