Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

Bachelor's Thesis

Title Title Title Title Title

This is a subtitle. It subs titles. Revisiting titles from the ancient Greek.

Torben Reetz

3reetz@informatik.uni-hamburg.de
BSc Software Systems Development
Matr.-Nr. 6524064
Semester 07

Primary Supervisor: Prof. Dr. Walid Maalej
Secondary Supervisor: Prof. Dr. Eva Bittner

Date: 03.2017

Abstract

Project driven organisations have to face the problem of constantly needing to put teams together based on the members' skills, experience and preferences. In many businesses, there is no sophisticated source of information about those data which makes finding the right person with a specific ability even more complicated. A central tool keeping track of all employees' competencies and important metadata such as soft skills would come in handy. As part of this thesis, a prototype of a web application fulfilling sinnerschrader's individual needs regarding will be contrived and partly implemented. The process of analysing those specific requirements with respect to the end users' needs and designing a technical architecture capable of fulfilling them will be documented. Furthermore, the backend part of said application will be implemented using state-of-the-art web technologies. It's internal structure will be explained and substantiated. The frontend components will also be implemented, but this will happen independently from the backend and thus cannot be part of this thesis.

Contents

1 SinnerSchrader

The SinnerSchrader Group is a full service web agency based in Hamburg aggregating the subcompanies SinnerSchrader Deutschland, SinnerSchrader Content, SinnerSchrader Commerce and SinnerSchrader Swipe. The broad spectrum of expertise,including, but not limited to, digital communication strategies, visual and interaction design, technical architecture, full stack development, editorial services, content production, E-Commerce, mobile app development, hosting, and maintenance, allows SinnerSchrader to serve all needs regarding their customer's digital transformation. The combination of all said competencies under one single roof reduces organisational friction between the discipline-specific teams because they all share the same vision of the big picture they are creating. This does not only lead to faster development cycles, but also to a more coherent and unified product.

1.0.1 Project-Driven Business

As a web agency, it is clear that SinnerSchrader has to operate in a project-driven way. This means there no continuous stream of recurring work repeating constantly, but many different projects for different clients, each one dealing with varying challenges and questions. From a technical point of view, the diversity of know-how needed for each project is extremely huge since every application uses its own dedicated stack of technologies. As a consequence, the developers' skill sets are based on the combination of projects they have worked on and their general field of interest. This results in one problem: Managers frequently have to put teams together based on the members' skills with respect to the individual requirements of the project.
This thesis will cover the creation of a tool helping managers with that problem by providing a centralized source of data about each employee's professional abilities.

2 Functional Objectives and Exisiting Solutions

2.1 Requirements

TODO: wo kommen die her, kann man da was machen?

2.1.1 Functional Requirements

- TODO: usable to all
- User Profiles
  Anyone can see another user's profile consisting of basic information about the user such as Name, Location, E-Mail and personal skills. Personal skills are composed of a name, a knowledge level and a will level, both on a scale from one to four.
- Provide/Edit skills
  Users can add new skills from a pool of known skills to their own profile. Already added skills can be edited and removed from the profile.
- Search
  A search function can be used to find people who have added one or more specific skills to their profile. When searching for multiple skills, only persons matching all of them will will be displayed.
  - Ranking
    By default, the search results order should be defined by a score aggregating the individual employee's skill level, will level and grade of spezialization in the searched skills. Employees willing to enhance their knowledge about a searched skill should be preferred to others having the same knowledge but a lower will.
  - Sorting
    The user should be able to sort the search results not only by said score, but also solely by knowledge and will level.
- Management of known skills
  New skills can be added to the set of known skills in the application. Existing skills can be edited and removed. Users personal skills are automatically updated when

a skill has been edited so that the integrity of the user profiles is maintained at all times.

2.1.2 Non Functional Requirements

TODO: Klären und formulieren

- Desktop/Devices
- Browsers
- Scalability
- Load/Response Times

2.2 Commercial Solutions

2.2.1 Skills Base

Skills Base[1] offers the required features, but also includes a large number of functionality SinnerSchrader does not need and ist not willing to use. This includes assessments, the categorization of skills and a role model for advanced access rights configuration. The search function does not provide searching for multiple skills. Furthermore, the sorting of results found cannot be customized. A central point of the application are dashboards displaying information about the most popular skills in the organisation and long term statistics.

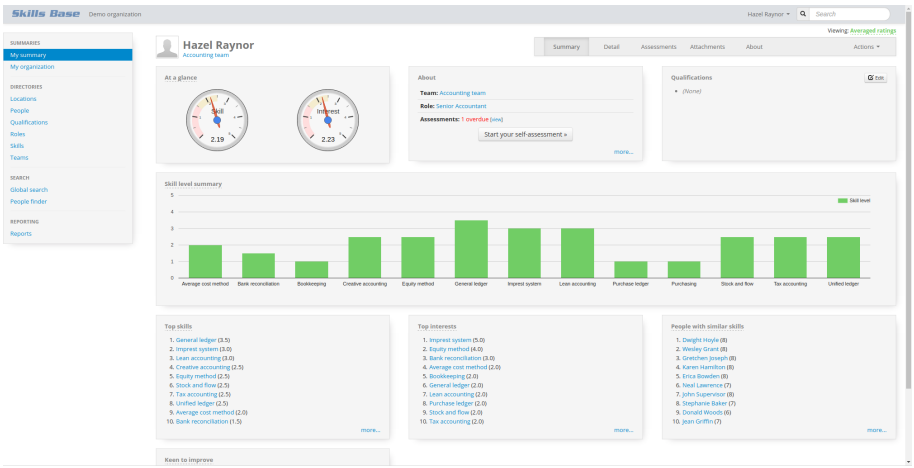

Figure 2.1: SkillsBase Dashboard

[1] http://http://www.skills-base.com/

2.2.2 Talent Management (engage!)

Talent Management[2] is a module for Infoniqa's management software engage!. It offers advanced features for managers such as a powerful search function controlled via a special query language. It also includes data about the employees' salaries, feedback protocols and certificates. It can only be used in combination with engage!, an complete human resources management solution including features like time tracking, e-learning, applicant management and payroll accounting.

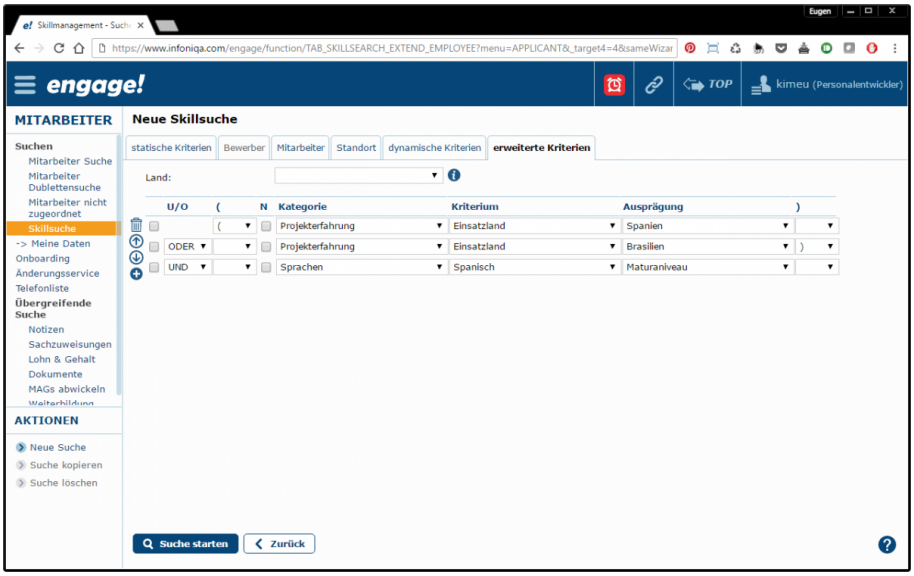

Figure 2.2: Talent Management Search

2.2.3 SkillsDB Pro

SkillsDB Pro[3] is an application designed to serve as a database in an organization providing an overview about every person's own skills and and trainings only to themselves and their manager. The search function is capable of searching for multiple skills combined with different logical operators which enables users to enter very sophisticated queries. Not only can users provide information about their skills, but managers can also do this with the limitation that no employee can see their manager's rating about themselves. Furthermore, only managers can search for persons. Taking into consideration that SinnerSchrader needs a tool to enable everyone to find someone with a specific skillset, this is a serious disadvantage. SkillsDB also offers features SinnerSchrader does not intent to use including, but not limited to the automatic generation of project reports based on plan succession and demands for assessments.

[2]http://www.infoniqa.com/hr-software/skill-management
[3]http://www.skillsdbpro.com

2.2.4 Conclusion

None of the analyzed softwares offers all features required, but all of them include various functions SinnerSchrader does not intend to use, which brings undesired complexity into the applications. One of the most critical features, sorting the search results by best match, is not offered by any of the commercial solutions. Furthermore, all those systems differentiate between employees and their supervisors and thus restrain transparency. Instead of a solution for monitoring employees and rating them, we want a tool for everyone to find another person who offers the skills needed to solve a concrete problem.

Pain Point Fitness Scoring

As shown by [Can13], motivation is a vital factor regarding any employee's performance and quality of work. Although motivation is a complex construct of many highly diverse dimensions, the overlap of a person's interests and their duties is a key aspect to it. Assuming that every member of the company has some skills they prefer to employ over others, matching people to tasks that require the exact same abilities they are interested in employing will lead to more motivated employees and thus have a positive impact on the overall productivity of SinnerSchrader. Consequentely, when searching for persons having specific skills, the application should not only take into acount the employees' skills but only their preferences in order not to find the most skilled, but the best fitting one. Unfortunally, none of the examined applications does provide a way to aggregate both, skills and preferences, into a single score indicating the overall grade suitability of a person relative to the searched skills.

3 Concept

The application should be accessible to all employees of SinnerSchrader. Due to the heterogeneity of the people's computer setups running Windows, macOS and Linux, creating a native application supported by everyone's system is a rather complicated task. A web application using standard technologies does not only solve this problem, but can also be used from mobile devices such as smart phones and tablets. Furthermore, there is no need to manually install and update the software so that it can be assumed that all users use the latest version of the application. This is not only a positive factor regarding the overall usability of the system, but also assures bugs and security issues are eliminated the moment a fixed version of the software is deployed. All those advantages compared to native clients and the fact that SinnerSchrader's expertise lies in the development of web applications lead to the decision, that such an application would be the appropriate solution.

3.1 Visual Concept & Wireframes

The application should be as simple as possible and usable for everyone, in order to provide an efficient and fast tool. Thus, it will be designed as a single page application based around a people search that provides a way to input the skills needed and returns all persons offering said skills. After entering a search, the user can select any of the found colleagues and view their personal profile showing extended information like contact details, more skills the user did not search for, and the employee's location. This profile will also include links to directly contact the inspected person via Email or Google Hangouts[1]. Unlike the considered commercial solutions, this tool will not include features like creating statistics, assessments, applicant management, or any dashboard other than the basic search view. Furthermore, there will not be any different roles with different access rights for employees and their managers, since this application is meant to be a tool enhancing collaboration, not supervision.

TODO: Put UML/BPMN here

[1] https://support.google.com/hangouts/answer/2944865

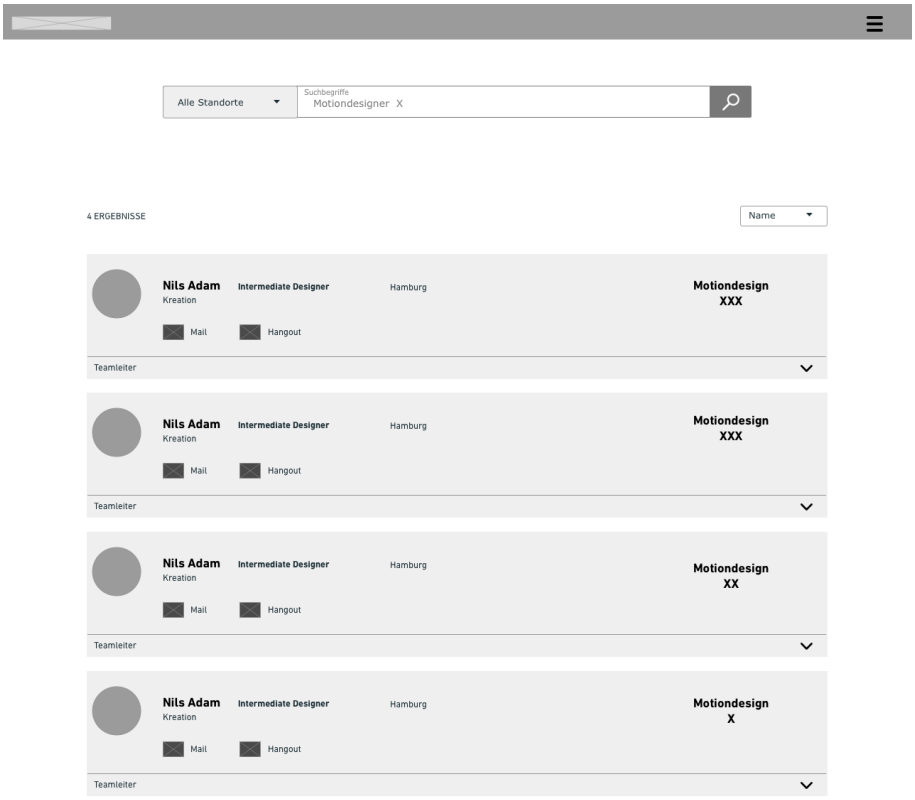

Figure 3.1: Wireframe

3.2 Scoring Algorithm

The application should (TODO wording, rein vom Konjunktiv her) sort all found persons by their fitness into the searched skill set. This implies there has to be an algorithm that can assign a score describing said fitness to every person found.

3.2.1 Requirements

According to [JHS07], an algorithm that matches persons to positions based on their skills has to meet more demands than solely the functional ones. They define the specific requirements such an algorithm assigning naval personnel to positions on a ship as follows:

- Easy to implement and maintain
- Fast to execute, so as not to become a computational bottleneck
- Takes into account factors: rating, pay grade and NECs[2] and future taxonomies characterizing required knowledge, skills and abilities
- Basic eligibility attained if achieving a specified score level Factors going into a skill match typically include:
  - Rating(s) or job type
  - Pay grade(s) or pay bands
  - NEC(s) or future taxonomies characterizing required knowledge, skills, and abilities

[JHS07, P. 14]

These qualities include factors very specific to the US Navy and thus will have to be evaluated and traslated into SinnerSchraders field of operation, but general requirements such an algorithm has to meet can be deducted: It may not be too complex as employees should be able to understand the system they are rated by, it should take into account different groups of factors and must be easy to adjust in order to keep the system maintainable.

3.2.2 Factors to Include

An Estimation of a concrete person's fitness into a position described by the searched skill set needs not only to take into account the matching of offered and required skills, but also the employees motivation to apply said skills derived from their preferences and their personal specialities and expertise. Latter can be described as the skill and will levels that stick out of their overall capabilities. So the important factors to be included in the algorithm are:

[2] Navy Enlisted Classifications

- Average level of knowledge regarding the searched skills.
- Average level of will regarding the searched skills.
- Specialization in the searched skills, including:
  - Specialization in knowledge about the searched skills.
  - Preference of the searched skills over others.

3.2.3 Proposed Fitness Score

Skill and will levels are described as integer values on a scale form zero to three. This scale, called V, can be expressed as

$$V = \{x \in \mathrm{N}_0^+ \mid 0 \leq x \leq 3\}$$

All existing skills are accumulated in the set S. The employee's skills are represented by E which is a subset of S. The search items are defined as Q.

$$S = \{java, ruby, ...\}$$
$$E = \{x \in S \mid \text{employee has skill x}\}$$
$$Q = \{x \in S \mid \text{user searches for skill x}\}$$

The functions v_s and v_w assign a level of skill/will to each value in E.

$$v_s : E \mapsto V$$
$$v_w : E \mapsto V$$

The averages of the employees' skill/will values of the searched skills are defined as a_s and a_w. The variables s_s and s_w describe the employee's specialization in the searched items defined as the difference of the average skill/will level of the searched items and the average level of all items. A person with maximal interest and knowledge in all searched items and an infinite number of skills with the minimal value of zero would have the greatest specialization possible and thus get assigned a value of one.

$$a_s = \left(\sum_{x \in E \cap Q} v_s(x) \right) \cdot \frac{1}{|E \cap Q|}$$
$$a_w = \left(\sum_{x \in E \cap Q} v_w(x) \right) \cdot \frac{1}{|E \cap Q|}$$

$$s_s = \frac{|V| + a_s - \left((\sum_{x \in E} v_s(x)) \cdot \frac{1}{|E|} \right)}{2\,|V|}$$
$$s_w = \frac{|V| + a_s - \left((\sum_{x \in E} v_w(x)) \cdot \frac{1}{|E|} \right)}{2\,|V|}$$

The resulting fitness score f is a weighted mean of the introduced facors. The weights $w_{as}, w_{aw}, w_{ss}, w_{sw}$ sum up to one.[3]

$$f = \left(\frac{w_{as} \cdot a_s}{|V|} + \frac{w_{aw} \cdot a_w}{|V|} + w_{ss} \cdot s_s + w_{sw} \cdot s_w \right) \cdot \frac{1}{4}$$
$$w_{as} + w_{aw} + w_{ss} + w_{sw} = 1$$

3.2.4 Example Caculation

Given are three example employees, Alice, Bob and Charlie, with the same three skills each. (Notation: skill level/will level)

Person	Java	Ruby	C++
Alice	2/1	2/2	3/3
Bob	2/3	0/3	0/1
Charlie	3/3	2/1	1/2

Applying the algorithm with $w_{as} = w_{aw} = w_{ss} = w_{sw} = 0.25$ to a search for the skills Java and Ruby results in the following values[4]:

Person	a_s	a_w	s_s	s_w	f
Alice	2	1.5	0.44	0.42	0.127
Bob	1	3	0.56	0.61	0.156
Charlie	2.5	2	0.58	0.5	0.161

Ranking the employees only by the average value of skill regarding the two searched items would result in Charlie being preferred to Alice and Alice being preferred to Bob. Sorting them using the proposed fitness score, however, would also designate Charlie as the most fitting person. Interestingly, Bob would be preferred to Alice, since his higher motivational levels compensate his lack of skill in Ruby.

3.3 Recommendation Engine

Recommender systems 'are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of [...]

[3] Mathematically, this is not neccessary, but it results in much more human readable values.
[4] Values have been rounded off to two significant digits.

information'[IFO15] that are commonly used to recommend an item to the user based on their previous interactions with other items. For example, recommender systems are used to predict products a customer might want to buy based on the ones they already bought in order to present those items more prominentely than articles the customer is unlikely to buy. In the context of the skill management application, a recommender system will be used to predict skills a user might want to search for based on the ones they already entered and a the knowledge which skills are more often searched for together. When entering an item into the search bar, the user will be offered to select items the recommender system predicts they might want to combine with the already entered ones.

3.3.1 Markov Chains

Markov chains are a relatively simple tool for predicting future states of systems based on the current state. In fact, makrov chains rely on the fact, that the next state of the system is only dependent on the current state which is called the 'markov property'. In the context of the skill management application, we assume this to be true because only two states will be examined: the current state is represented by the set of all items entered in the search field; the future state is skill set of the current search plus one more item. The basic concept is to store all possible states of the system and the respective probability of switching between from any state to any other one. Knowing the current state one can easily deduct the most probable future state. When a state transistion occurs, the outgoing probabilities of the origin states can be adjusted accordingly in order to factor the transition into the prospective projections.

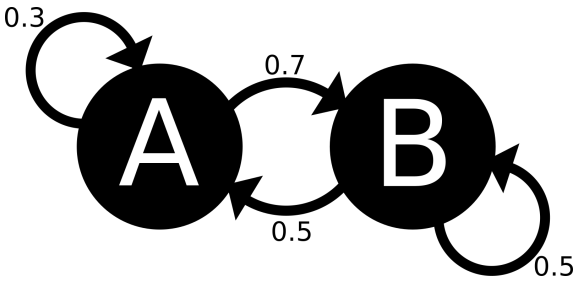

Figure 3.2: A simple markov chain displayed as graph. The states are represented as vertecies. All possible transitions between states are denoted as edges. The edge weights define the probability of the transition relative to all other outgoing edges of a node.

3.3.2 Proposed Recommendation Algorithm

In order to get the best results, the recommender system should save all combinations of items that have previously been searched for together and then recommend the next one based on this exact starting point. On the other downside, this would require the system to save a large number of origin states and thus consume memory. Furthermore, it would

result in a lot of different origin states that contian very few possible future states due to being to specific to a single search. Another way to implement such a recommendation system would be to only inspect the last entered search item and ignoring all other ones. Given n known skill items, all probabilities for the future state could be saved in one single $n \times n$ matrix saving only n^2 values. This solution would disregard the whole context of the search item. A usable tradeoff would be to store all known items together with a list of all other items the former hast been searched together with and the count of times this has happened before. In order to suggest the next skill to search, the recommendation algorithm has to merge the lists of simiar items of every searched item in order to sum up the search counts. The resulting list contains skills that users have searched before in combination with the given ones, their counts have been aggregated. The item having the highest total count in the list will be suggested.

3.3.3 Pseudo-Code

A possible pseudo-implementation of the propsed algorithm could look like this:

```
var knownSkills = [
  {
    name: "java",
    similar: [
      {
        name: "php",
        count: 3
      }, {
        name: "ruby",
        count: 2
      }
    ]
  }, {
    name: "php",
    similar: [
      {
        name: "java",
        count: 5
      }, {
        name: "ruby",
        count: 2
      }
    ]
    name: "ruby",
    similar: [
      {
        name: "java",
        count: 0
      }, {
        name: "php",
        count: 5
      }
    ]
  }
]
```

```
function suggest(searched) {
  var accumulated = {};

  for (s in searched) {
    for (t in knownSkills.getByName(t).getSimilar()) {
      if (accumulated.getByName(t) exists) {
        accumulated.getByName(t).count += t.count
      } else {
        accumulated.getByName(t) = t.clone()
      }
    }
  }

  return accumulated.getHighestCount();
}
```

4 Implementation

4.1 Application Structure

4.1.1 MongoDB

BSON

Data Structure

Queries

4.1.2 LDAP

4.1.3 Reverse Proxy

4.1.4 API

4.2 Infrastructure

4.2.1 Maven Build

Frontend Build

4.2.2 Gitlab CI

4.2.3 Deployment

4.3 Backend

4.3.1 Spring Boot MVC

4.3.2 Spring Data

4.3.3 Spring LDAP

4.3.4 Swagger

4.3.5 Testing

4.4 License

5 Evaluation

5.1 Implementation vs. Algorithm

Tut die Anwendung das, was definiert wurde

5.2 Algorithm vs. Flow Team

Tut die anwndung das, was das Flow team für richtig hält

5.3 Conclusion

War das jetzt was?

Bibliography

[Can13] CANÓS-DARÓS, Lourdes: An algorithm to identify the most motivated employees. In: *Management Decision* 51 (2013), Nr. 4, 813-823. http://dx.doi.org/10.1108/00251741311326581. – DOI 10.1108/00251741311326581

[IFO15] ISINKAYE, F.O. ; FOLAJIMI, Y.O. ; OJOKOH, B.A.: Recommendation systems: Principles, methods and evaluation. In: *Egyptian Informatics Journal* 16 (2015), Nr. 3, 261 - 273. http://dx.doi.org/http://dx.doi.org/10.1016/j.eij.2015.06.005. – DOI http://dx.doi.org/10.1016/j.eij.2015.06.005. – ISSN 1110–8665

[JHS07] JANET H. SPOONAMORE, Ricky D. H. H. James Simien S. H. James Simien: *Person-to-Position Matching*. 2007

List of Figures

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und mich anderer als der im beigefügten Verzeichnis angegebenen Hilfsmittel nicht bedient habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe und die eingereichte schriftliche Fassung der auf dem elektronischen Speichermedium entspricht.
Ich bin mit einer Einstellung in den Bestand der Bibliothek des Fachbereiches einverstanden.

Hamburg, den _____ Unterschrift: _____