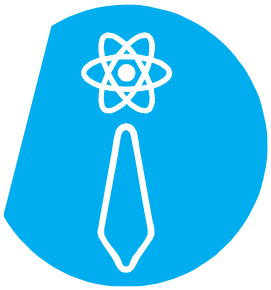


Staffr

Software Requirements Specification

for Project Staffr

Prepared by
Kryštof Sýkora, Marek Szeles
ČVUT FEL SIT,
Enterprise architectures

Version 1.0
1. 10. 2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Marek Szeles	12. 7. 2017	Document initialization	0.1
Marek Szeles	26. 7. 2017	First chapters written	0.2
Kryštof Sýkora	10. 8. 2017	Domain Model v1	0.3
Marek Szeles	12. 8. 2017	First half of chapters done	0.4
Marek Szeles	16. 8. 2017	Use cases updated	0.5
Marek Szeles	20. 8. 2017	Graphics update	0.6
Kryštof Sýkora	24. 9. 2017	Domain Model v2	0.7
Marek Szeles	29. 9. 2017	Revison	0.8
Marek Szeles	1. 10. 2017	Doc finalization	1.0
Kryštof Sýkora	19.11.2017	Changes to functionality and domain Model based on last consultation.	1.1
Marek Szeles	26.11.2017	Update following discussion at CP1	1.2

1. Introduction

1.1 Purpose

The product of this project will be a Java EE and ReactJS based Maven compilable program that allows for staff administration to a company representative with appropriate rights.

Within the EAR project scope, we are aiming to create a system for staff administration for businesses. The primary use is for managers to have a clear overview of available staff they could assign positions on ongoing projects.

The project output is especially intended for larger international corporations in need of sourcing its projects from broad selection of available staff on hand, with various areas of expertise.

1.2 Intended Audience and Reading Suggestions

This document is intended for anyone interested to learn about the concepts and architecture used in the Staffr project.

2. Overall Description

2.1 Product Perspective

This is a standalone product, completely independent in its basic functions. If its planned advanced functions are to be implemented, some use of external APIs is to be expected.
Also besides the basic functions, others are more demonstrative and will not necessarily be implemented for the EAR cp2 checkpoint.

2.2 Product Functions

2.2.1 Basic functions to be implemented within project scope

- User roles
  - There are to be three levels of user power within the system:
    - Admin – Has all the administrative rights to the system – i.e. to create and edit users, administer their contracts and administer the login credentials of other users
    - Project leader – Has the rights only to search for staff to assign to a project
    - Standard User – Same as project leader, but can only view own profile
- Staff search through filtering
  - The staff (standard users) will have several attributes (specialties/expertise) through which they can be searched. The fuctions we will actualy implement will however be limited and of a more demonstrative nature.
  - The use cases we wish to implement are: search for a people with expirience on specific projects, people currently ithout proejct and employes with given skills.
- Basic user administration
  - The admin user is going to be able to edit details about other users, such as personal details, location and areas of expertise (resumé).

2.2.2 More advanced functions NOT to be implemented within project scope

- Project pages
  - Editable pages for individual projects with participants, user roles and goals listed.
- KPI tracking
  - Key performance indicators and deadline tracking for projects

2.3 User Classes and Characteristics

Apart from the state before logging in, there are only three expected user classes defined within the Staffr project – user, project leader and admin. Both have very similar functions, only overall rights are different. The roles are not overlapping and thus are exclusive.

2.3.1 Everyone (before logging in)

Anyone accessing the server/side program through a browser will be presented a simple form to log into the system using a username and password.

2.3.2 Standard User

After logging in, a user can review his account, review the history of the account, and manage own personal details.

2.3.3 Project leader

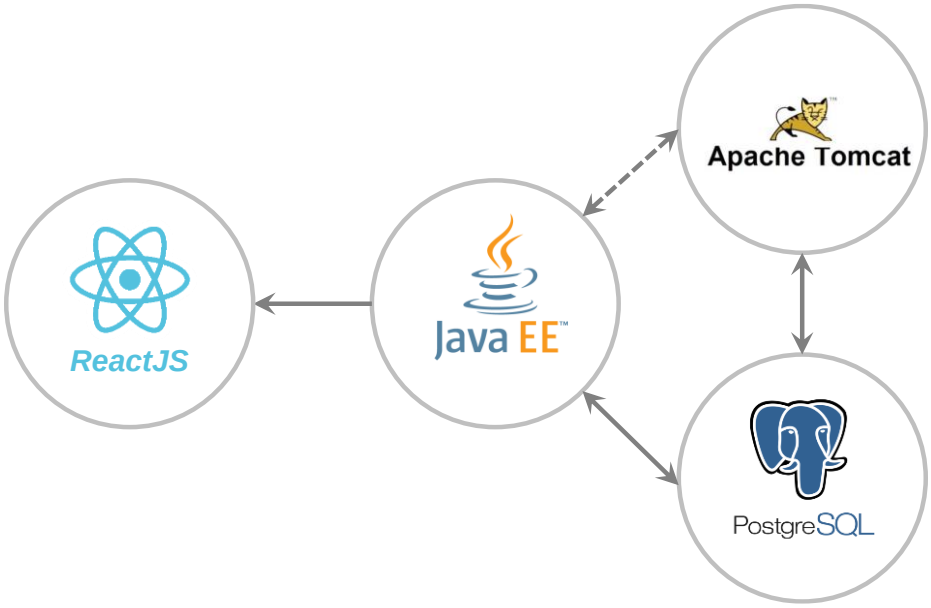
Has the rights only to search for staff to assign to a project. Can also edit project pages.

2.3.4 Admin

An admin has all the rights a project leader has, but also has all the administrative rights to the system – i.e. to create and edit users, administer their contracts and administer the login credentials of other users.

2.4 Operating Environment

The backend source code of the program is assembled mainly using Java Enterprise Edition. It is deployed on a Tomcat server environment, using a PostgreSQL database to store inputted information. The frontend is created using the ReactJS library.



2.5 Design and Implementation Constraints

If some of the advanced functions are to be implemented, the development is likely to be limited by external APIs – for example, if using an external API to recognize text, or when connecting to an external API handling access to different systems (mailbox handlers, other enterprise applications, etc.)

2.6 User Documentation

Overview and development documentation will be provided, along with a simple user manual in the form of pdf files.

3. External Interface Requirements

3.1 User Interfaces

The GUI is going to be design with minimalism and simplicity in mind in order to allow for intuitive user flow of work. The layout is to be clearly separated into functional and passive parts, with labels describing all buttons and functions.

Here is a draft of how the interface may look like:

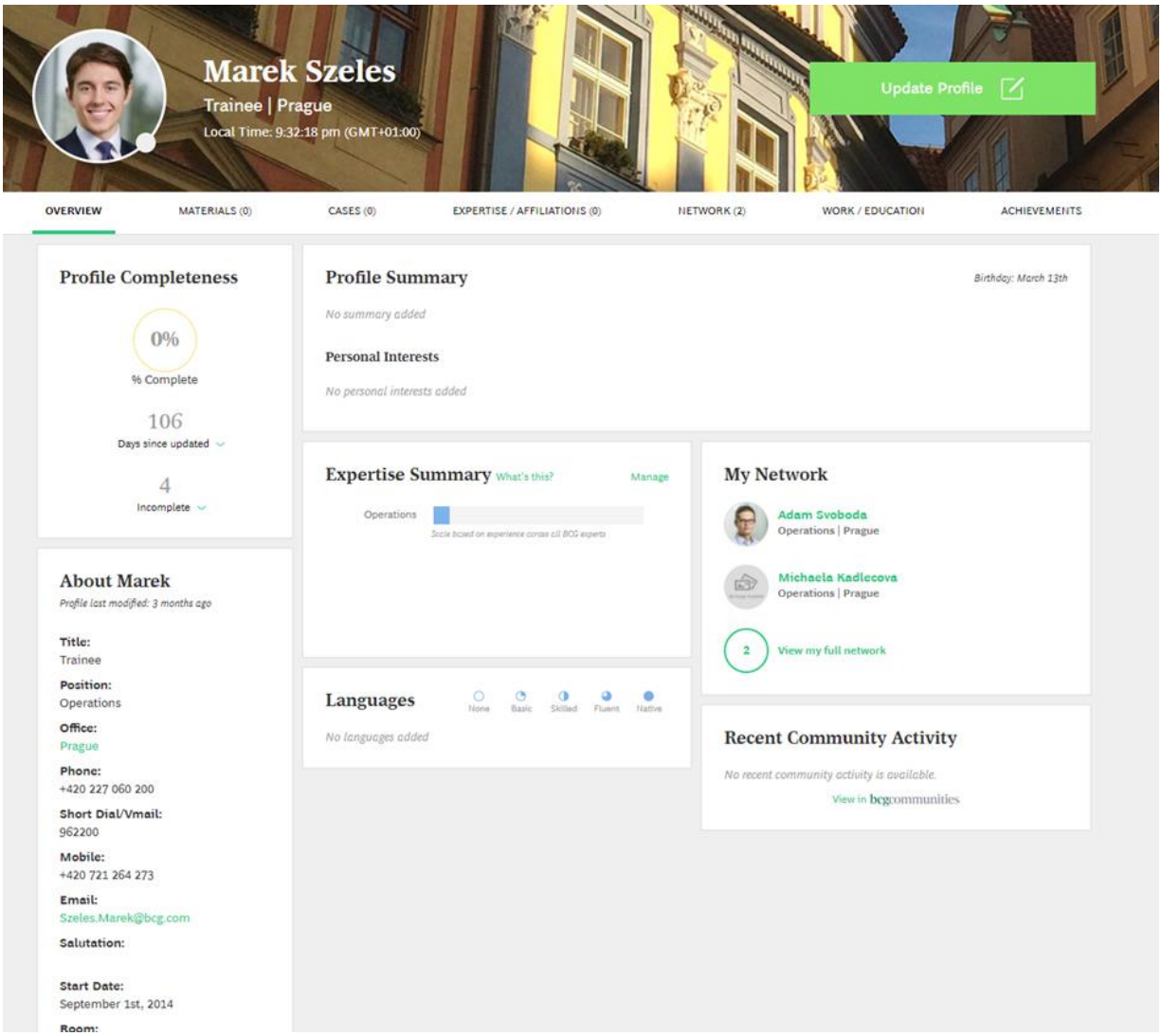


Figure 1: User screen

3.2 Software Interfaces

The program in its basic functions is fully integrated with itself and needs no access to outside sources. If and when the advanced functions are implemented, they will by definition need interfaces to access outside systems, however these systems and the nature of such transactions are to be determined when it comes to the relevant phase of development.

3.3 Communications Interfaces

Communication between the server and the user is going to take place through a web browser, using the HTTP protocol. All other communication is translated directly into code and objects within the program itself.

4. System Features – Use Cases

This section describes the features to be implemented in the project and features planned for future development.

4.1 Basic features to be implemented within CP2

4.1.1 All users

Login

Description and Priority

The user will be able to login using a combination of username and password

High priority

Stimulus/Response Sequences

The user selects login > Fills out their username and password > System checks if an entry in database equals to input > Login complete

Functional Requirements

The software is required to not mix up requests.

Show user overview

Description and Priority

C**R**UD read operation.

The user will be able to view his account overview

Low priority

Stimulus/Response Sequences

The user selects overview > User accounts overview is shown

Functional Requirements

The software is required to show all information stored relevant to a particular employee

Edit own personal information

Description and Priority

CR**U**D update operation.

The user will be able to change information on their account overview, including assigning skills to their profile.

Low priority

Stimulus/Response Sequences

The user selects overview > User accounts overview is shown > Selects edit > Fills out new information > Selects update > Page is reloaded with new information

Functional Requirements

The software is required to show all information stored relevant to particular employee

4.1.2 Project Leader only

Search for user according to criteria

Description and Priority

The project leader can search users using filtering based on project involvement, skill, etc.

Medium priority

Stimulus/Response Sequences

The user selects search employees > Fills out criteria > Selects search > Relevant users overview is shown

Functional Requirements

The software is required to show all employees in organization relevant to the search

Creation of project pages

Description and Priority

CRUD create operation.
The project leaders will be able to create projects they are leading and staff assigned to them, including KPI tracking, etc.

Medium priority

Stimulus/Response Sequences

The project leader selects create new project > selects assign staff > system periodically refreshes project staff capacity status > at end of project, PL inserts outcome evaluation

Functional Requirements

The software is required to repeatedly and correctly refresh connection to project pages

4.1.3 Admin only

Register new user

Description and Priority

CRUD create operation.
The administrator will be able to register a new user using a username and a password of his choosing

High priority

Stimulus/Response Sequences

The admin selects create new user > Fills out their username and password, which he inputs twice for confirmation > Registration complete

Functional Requirements

The software is required to assign unique Ids to all new users, check for database consistency (no duplicate usernames) and handle security (hash passwords)

Remove user

Description and Priority

CRU**D** delete operation.
The admin will be able to remove users (employees) from the system

High priority

Stimulus/Response Sequences

The admin selects specific user > Fills out information about removing (if effect is immediate, or time-bound, associated data handling, etc.) > confirms > User removed

Functional Requirements

The software is required to keep track of all existing dependent classes on user and handle them according to their input. For example, the removal of an employee shouldn't mean a removal of a project they were part of, or the office they are from, however, it would mean the removal of the address associated with them, assuming they are the only ones living there.

4.2 Advanced features NOT to be implemented within project scope

4.2.1 All users

Login using Google or Facebook

Description and Priority

The user will be able to login using Google or Facebook for login credentials,
Low priority

Stimulus/Response Sequences

The user selects login > Login using Google/Facebook > System checks if user is registered > login complete

Functional Requirements

The software is required to handle authorization communication with external servers.

Scanning CVs and other documents

Description and Priority

The system will be able to fetch relevant information from a digital copy of a CV, even a photo
Low priority

Stimulus/Response Sequences

The user selects scan CV > uploads picture > system fetches information from data and pre-fills form > asks user to confirm > new entry created

Functional Requirements

The software is required to efficiently and correctly fetch data from different digital sources

Connect to MS Outlook/other widely used systems

Description and Priority

The system will connect to other present enterprise systems used by the organization
Low priority

Stimulus/Response Sequences

The user selects other systems > system to connect > computer monitors relevant data and informs user of development

Functional Requirements

The software is required to efficiently and correctly fetch data from different servers

4.2.2 Project Leader only

Deletion of project pages

Description and Priority

CRU**D** delete operation.
The project leaders will be able to delete projects they are leading.
Medium priority

Stimulus/Response Sequences

The project leader selects project > selects delete > fills out explanation form > sets parameters of delete > confirms

Functional Requirements

The software is required to handle dependencies of project – delete individual relationships coming from employees, however noth deleting these entities themselves.

4.2.3 Admin only

Creation of a user using Google or Facebook

Description and Priority

The admin will be able to register users using Google or Facebook for login credentials.
Low priority

Stimulus/Response Sequences

The user selects connect > Connect using Google/Facebook > Grants permission to Staffr to fetch email > Registration complete

Functional Requirements

The software is required to handle authorization communication with external servers.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The program is expected to handle all logged users with no noticeable delays. To assure this, sufficient hardware is needed for possible service overload.

5.2 Security Requirements

Due to the nature of the stored data, the database and user login data is going to be secured using different methods, such as hashing passwords and passing data through secure connections, to which a separate security layer will be committed.

5.3 Software Quality Attributes

The resulting software should be flexible and adaptable to different uses – it should be simple and intuitive to add new categories or tags for example. Also, if, in the future, a new world currency comes to exist which is going to be widely used, its retrospective implementation into the project should be simple as well.

5.4 Business Rules

While all users can see other user's profiles, only project leaders can assign employees to projects and only admins can create new users.

6. Other Requirements

The project is required to have at least five database tables, have at least one M:N relation and use one dependency.

Appendix A: Glossary

Java EE – Java Enterprise Edition

GUI – Graphical User Interface

PL – Project Leader

Appendix B: Analysis Models

Domain model