

Final Report Document

for Project Staffr

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

Version 0.3  
30. 12. 2017

# Summary

## Introduction

With Staffr, we aimed to produce a Java EE based Maven compilable program that allows xxx.

## Development report

Xxx

In effect, most of the program is heavily inspired by the provided “reporting tool” and “ear-setup” repositories, with other functions added to the body.

# Used Technologies

For most of the program, we were inspired by the “reporting tool” concept, so we built our program on a similar base wireframe.

## Functioning additions

### ReactJS

xxx

### JavaScript Library “Raphaël”

For graphs and other graphics, the javascript library/vector graphics mapper called “Raphaël” was used, as found here: <http://dmitrybaranovskiy.github.io/raphael/>

## Failed technologies

### Javascript emulation

To save time, we have tried to use the DotVVM (<https://www.dotvvm.com/>) project as a framework to generate javascript, however implementations of this have failed to compile and be compatible with the main project, and so we abandoned the idea.

# Project outputs

The outputs are a Java EE Maven compilable program with pre-defined basic functionality, user manuals for it and multiple documents reporting on the development and purpose of the program.

# Installation and deployment

## Development Environment Setup

The following software needs to be installed on the system for development:

* JDK 8
* NodeJS v6 or later
* ReactJS
* Maven
* Apache Tomcat (or any other application server)

To start developing, first go to src\main\webapp and run npm install. This will download the necessary Node dependencies (they are used by the UI written in ReactJS). You can check that everything is working by running npm test.

## Storage Setup

The application uses a standard relation database. It is preconfigured to a PostgreSQL server named Staffr\_db running at localhost:8084, and credentials Staffr/Staffr.

## Running the Application

To run the application locally, start JS compile watcher by running npm start from app/root/src/main/webapp. The watcher will recompile JS whenever a change is made to the UI code.

Running the application is simple, just build it with maven and deploy the artifact into you application server.

# Prepared Sample User Walkthrough

## Booting the application

## Login

Credentials:

This user is a project leader

## Show own user page

## Edit personal information

## Search for user according to criteria

## Create project page

## Delete project page