

Final Report Document

for Project Cashr

Prepared by   
Yana Koval, Marek Szeles  
ČVUT FEL SIT,   
Enterprise architectures

Version 1.0  
9. 1. 2016

# Summary

## Introduction

With Cashr, we aimed to produce a Java EE based Maven compilable program that allows its registered users to analyse their personal finances.

## Development report

Even though we had an ambitious plan for our program, due to several time and other limitations of the team, the finished product is very limited in both scope and function. We have encountered many obstacles during development, specifically since our experience was very limited and we weren’t very effective at pooling resources and didn’t know where to look for external sources.

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 “repository tool” concept, so we built our program on a similar base.

## Functioning additions

### 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/>

### Neural Network Search

For contextual search and categorical analysis, a very simple neural network was used that was developed through a tutorial shortly before the main project.

The Network can be feeded a dictionary of keywords and categorise them to be income or expense, for example. It is optimized to have a wide application for a two-set array of input and a bool output test cases.

Source and inspiration from here: <https://www.udemy.com/neural-networks-from-scratch-in-java/>

## 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, user manuals for it and multiple documents reporting on the development and purpose of the program.

# Installation

## Development Environment Setup

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

* JDK 8
* NodeJS v6 or later
* 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 cashr\_db running at localhost:8084, and credentials cashr/cashr.

## 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.