Leopold-Franzens-Universitt Innsbruck

Institut fr Informatik
Datenbanken und Informationssysteme

Development of an OpenSource Feedbackcommuncationplatform

Bachelor-Arbeit

Martin Karrer

betreut von
Wolfgang Gassler und Eva Zangerle

Innsbruck, April 11, 2016

Abstract

Within this work, the development process of an open-source feedback collecting tool with the ability to communicate with the feedback giving person and the maintainer. Most feedback applications only offer a one-way communication from the feedback giving person - to the feedback requester. The approach to develop this system, with the ability to communicate or discuss a feedback, which is extendable, scale-able and especially easy to maintain and install is described in detail. All used techniques and patterns used in this project to implement the single-page-application will be described in this thesis. It tires to show the usage of a functional language in everyday life and all advantages and disadvantages which occurred in this project. The presented application is a minimal but extendable and scale-able open-source platform for requesting and discussing feedback.

Vorwort

Contents

Chapter 1

Introduction

Ever thought about getting honest feedback? This does not work if it is not possible to collect this feedback anonymous. To achieve this, the system has to accept feedback with user information and anonymous without. The application presented in this work can be used to get this information and the feature to discuss.
It is common to use an e-mail mailbox or some online discussion boards to collect feedback. Most of them does not scale for the user. An example: Holding a talk and want to know how it was for the audience of more than 400 people? Send an e-Mail with the feedback does not really scale, because if you have a feedback friendly audience you will have 400 messages in your inbox. Managing Feedback in a discussion board is a huge hassle to find and manage those feedback, without thinking about the ability to discuss a specific feedback.

Chapter 2

Requirement Specification

In the following, a description of the software requirement of Constructeev is given. This chapter describes mainly the requirements, while Chapter 5 gives a deep insight into the development of the application.

2.1 General Requirments

The Application needs to be accessible from all kind of devices, started by mobile-phones, tablets, notebooks and desktop workstations. Web technologies, like Javascript, CSS and HTML ensures that the user only needs to have a HTML5 compatible Browser, which is pre-installed on all of the devices listed above. All feedback relevant data need to be stored permanently on a storage medium and should be accessible at any time. It is important to keep the data in a structured way and it should be possible to change the database adapter and migrate data vom one to another database system without any hassle.

The Application needs to be spitted into two segments in a so called client- and server-application. Both applications should run on as many platforms as possible without any constraints to productivity and scalability.

2.1.1 Server Application

The server application should be scalable and runnable on almost any operating system and platform. Furthermore it has to provided a uniform interface to the client application, which will be in this case a JSON-based REST-like API. The datastorgae has to be a common SQL-Database like mysql or Postgresql. It should also be possible to exchange the database at installation time.

2.1.2 Client Application

Same as the server application, the client application needs to run also on almost any platform but targets end-user hardware with an display and a HTML5 compatible Browser. AngularJS is used to build a single-page-application with a model view controller pattern on client side. Asynchrone API calls will communicate with the application server.

2.2 Server System Requirements

The server system has easy to be set up and also easy to maintain. The pure application resulting from this project is also running as a public service available to anyone. So scalability is also an important key point. A channel with more than 16.000 feedback should be as responsive and useable as one with only 20 feedback. Not only the amount of data should scale, also the number of requests should be easy to handle by adding more and more servers.

2.3 Channel Components

To create a easy to use application it is important to be minimalism. A channels task is to hold a specific quantity of feedbacks for a specific topic set by the admin of the channel. There is always one admin per channel. Users which need more than one channel will get seperate login hashes for each channel. This makes sharing a company channel or community channel easy.

2.4 Feedback Components

2.5 Scalability

2.6 Administrator View

Chapter 3

Related Work

Arsnova

Chapter 4

Used Technologies

4.1 Elixir

4.1.1 Phoenix

4.1.2 Cowboy

4.1.3 ETS

4.1.4 Ecto

4.2 PostgresSQL

4.2.1 Schema

4.3 AngularJS

4.4 Semantic-UI

4.5 Server-Platform

Chapter 5

Implementation

Chapter 6

Usability and Responsive Design

Chapter 7

Performance and Scalability

Chapter 8

Conclusion

Appendix

A.1 Subsection Appendix