Affidavit

Ich, Jan Monschke, geboren am 12.03.1987, versichere, diese Bachelorarbeit selbstständig und lediglich unter Benutzung der angegebenen Quellen und Hilfsmittel verfasst zu haben.

Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Düsseldorf, den

Jan Monschke

Table of Contents

Salon

1. Overview

Salon is a web-based system that allows its users to create pages and to upload images onto these pages. On a first sight this functionality may not look very innovative since there are millions of services on the Internet that allow the user to upload images. But the main improvement that Salon offers that other services don't offer is that registered users are able to fully control the way their images are presented to the visitors of their pages. All images are placed on a canvas and can freely be dragged around by the user to create innovative and unique arrangements. Also the canvas itself can be moved to focus a certain point of a page. Another feature is that images can link to other pages so that users can create associations between pages or even associations between users.

2. The idea

Dipl. Inf. Sebastian Deutsch and Dipl. Des. Stefan Landrock developed the basic idea behind Salon when they were given the chance to take over university courses at HFG in Offenbach. Together with their students they built a working prototype of their idea so they could use it for their courses and especially for their presentations. When other universities heard about Salon they were asked if they could host a system for their students too. But Salon was not built to be deployable for other universities and so they had the idea to completely rewrite and to extend the features of Salon so that it could easily be set up for other universities.

3. Implementation

3.1 Technology

The backend of Salon is implemented in Ruby on Rails (short Rails), a web framework written in Ruby[1] and modeled after the MVC software pattern[2] that allows to quickly create solid web applications without having to care about low-level problems like session-handling or database access. The underlying database is MongoDB[3], a document-oriented database system that was chosen because of its flexibility (document-oriented databases are schema free[4]) and its very good integration into Rails.

Salon does not make use of the frontend tools of Rails because the frontend is designed to work as a Single Page Web App (SPWA) and therefore all Rails frontend tools have been replaced with tools that are written in JavaScript so that they could get executed in the browser (see SPWA#intro).

The communication between the frontend and the backend is realized with a REST[5] interface and all data is being sent in the JSON[6] format, a format that is very easy to use in both JavaScript (frontend) and Ruby (backend).

3.2 Data Model

[ADD schema here]

The underlying data structure of Salon is rather simple. There are users that are used for authentication and have basic properties like a username and a password. Pages are associated to users in a one-to-many relationship, which means that users can have as many pages as they want and each page belongs to only one user. Pages have properties like a title, a cover image and a publish state. Each page has a list of assets that are also associated in a one-to-many relationship so that each asset can be associated to one page. Asset is the parent class for image and it stores properties like a title, a link-to location and a position on the canvas.

The reason for deriving image from asset is to allow other assets like for example texts in the future (see salon#quovadis) and to provide all derivations with the needed

---

[1] http://rubyonrails.org/
[2] http://betterexplained.com/articles/intermediate-rails-understanding-models-views-and-controllers/
[3] http://www.mongodb.org/
[4] http://en.wikipedia.org/wiki/Document-oriented_database
[5] http://de.wikipedia.org/wiki/Representational_State_Transfer
[6] http://en.wikipedia.org/wiki/JavaScript_Object_Notation

properties to have a valid asset. The image then only needs to save special properties like the image files and its display sizes.
All assets have a list of tags that are associated in a many-to-many relationship which means that a tag can get belong to many assets and assets are able to reference many tags. Tags are used in the search (see salon#pages#search).