# 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

Affidavit 1

Table of Contents 2

Salon 2

1. Overview 2

2. The idea 2

3. Implementation 3

3.1 Technology 3

3.2 Data Model 3

3.3 Navigation 4

3.4 Menu 5

3.5 Pages 5

3.5.1 Index 5

3.5.2 Registration and Account 6

3.5.3 User Overview 6

3.5.3 Page Overview 7

3.5.4 Page Edit Form 7

3.3 Drag and Drop 8

3.4 Quo vadis Salon? 8

3.5 Evaluation 8

# 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]](#footnote-1) and modeled after the MVC software pattern[[2]](#footnote-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]](#footnote-3), a document-oriented database system that was chosen because of its flexibility (document-oriented databases are schema free[[4]](#footnote-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]](#footnote-5) interface and all data is being sent in the JSON[[6]](#footnote-6) format, a format that is very easy to use in both JavaScript (frontend) and Ruby (backend).

### 3.2 Data Model

[ADD schema image 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 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 belong to many assets and assets are able to reference many tags. Tags are used in the search (see salon#pages#search).

### 3.3 Navigation

[ADD screenshot of navi here (initial and completely expanded version)]

The navigation in Salon is designed in a breadcrumb-like style[[7]](#footnote-7). When first visiting the website, the user only sees a caption saying "Salon" which should tell him that he is currently in the most top level of the website. The second element in the navigation is the search bar. Then when the user goes to the overview of a user, the caption "Salon" is replaced by the name of the user that this page belongs to. Removing the "Salon"-caption should emphasize that Salon is about the users and about the work they want to present and that it is not about the platform itself. Normally the first element in a breadcrumb navigation brings the user back to the index page but this is not a scenario that applies to Salon because the index page, intentionally, doesn't offer more features than any other page in Salon but the list of recently created and edited pages. Searching for images is possible from any page through the search field that also resides in the navigation (more on search in [x.y]). If a user wants to go back to the index page he can do this via the menu in the top right corner (more on the Menu in [x.y]) at any time.

Another element, the name of the current page, is added to the navigation when the user navigates to a page of a user, and another one, the position of the current image in this set of images, is added when the user navigates to a specific image of a page. The breadcrumb navigation helps the user to keep track of certain information like the owner of the current page and the page an image belongs to. In that way these relations don't need to be displayed on every image or every page, which leads to a cleaner and lighter interface. Also the navigation helps a user to quickly jump back to a user's overview without having to manually navigate there with the back button.

### 3.4 Menu

The menu is consistently placed in the top right corner of each page and is by default not expanded so that it doesn't unnecessarily take away screen space. To expand the menu the user simply needs to hover over it with the mouse or tap it on the screen (on touch-based devices).

There are two states for the menu: a) The user is logged in; b) The user is not logged in.

[add screenshot of menu when user is not logged in]

When the user is not logged in the menu will have the caption "Sign in/up" which stands for the two most important options that are displayed in the expanded menu. The first point in the menu (see Fig. XX) will lead the user to the sign-in form and the second point to the sign-up form. The third point will lead to the about page, that explains the concept of Salon, and the fourth point will lead the user back to the index page.

[add screenshot of menu when user is logged in]

The caption for the menu when a user is logged is its username. The first entry in the expanded menu now leads the user to his own overview page, which allows the user to quickly jump there from any other page. The second entry will lead the user to the page index (more on that here x.x) and the third one will lead the user to the account page where he can edit details of his account.

The fourth entry is highlighted and it is an interactive entry because when this entry gets clicked it will reveal a simple form that lets the user create a new page right from the menu [show screenshot with the states of the form]. After submitting the form the user will get forwarded to the newly created page. By placing the form inside the menu there is no need to create an own page just for adding a new page and the user is able to create pages no matter on what page he is currently on. He only needs to be logged in.

The fifth entry triggers a log-out and a redirect back to the index where also the last entry is leading.

### 3.5 Pages

#### 3.5.1 Index

The index page has, as well as other elements in Salon, two states that depend on the login state of the user. If the user is not logged in the index page displays a text that invites the user to register an account at Salon and a link to the about page so that new users quickly get an idea about what Salon is and how they can use it.

If the user is logged in, the text on the index page welcomes the user and a list of recently created and edited pages is shown at the bottom of the page.

The index page does not have much functionality since the discovery of pages and images is realized with the search field that is located in the navigation (more on search here [x.x]).

#### 3.5.2 Registration and Account

#### 3.5.3 User Overview

[add screenshot of a simple user overview]

In the user overview all, published and not hidden, pages of a user are displayed on a canvas. Since pages can have a cover image, on this page only the cover images are shown. If a page does not have a cover image, a default picture will get displayed instead. The user is able to arrange all images just by dragging them around (more on DnD here [x.x]). The positions are being saved to the server so that this page will look the same for all visitors and just as the user wants it to look like. Visitors themselves can also drag the images around and create a new layout but the position will not get saved to the server since only the owner has the right to decide how his pages look like.

When dragging an image the image will get populated to the top of all other images so that users can easily create nice effects with occluding images. All theses changes will all automatically get saved to the server without the need for the user to initiate the save-process.

To highlight the importance of the images and especially their arrangement there are no further information displayed on top of each image. This is also done so that text elements don't clutter or disturb arrangements that contain a lot of images.

[add screenshot of two hover states (logged-in, not logged-in)]

Further information for a page is displayed on top of the images when a user hovers (or taps) over one of the images. The name of the page and the number of assets that are contained in this page will then fade in and the image gets a half-lucent overlay to highlight which image currently is being hovered. There is the need for the half-lucent overlay because when many pages are placed in the same page it is hard to find out which of the images just has been hovered. Clicking one of the images or its captions will navigate the user to the overview of the page.

When the current user is logged in there will also be additional controls displayed on top of each hovered image. First there is the control to set the size of the image that lets the user choose between four different size options. Then there is a link to the edit page of the current page that allows a user to quickly edit the page and there is a link to delete the current page. All delete operations in Salon trigger a prompt before actually deleting an element to prevent accidental deletions.

Furthermore the user is not only able to drag each image around but also the whole page which allows to choose a special "starting" point of the canvas that the visitor sees when he first comes to the page. To drag the whole page the user simply needs to drag the background and all other images will get moved accordingly.

#### 3.5.3 Page Overview

At first sight the page overview looks similar to the user overview. The images can freely get dragged around and the title of each image is displayed when the image is hovered. Logged in users also have the ability to directly edit or delete images with the additional captions here. As in the user overview the user is navigated to the image page when he clicks the image or one of the captions on the image.

[add screenshot of images with special icons]

Besides that, there are subtle changes to some of the images. They have special icons that should indicate that they don't link to the image page but to an external page (see Fig.YY e.g. http://google.com) or to an another page of this user (e.g. test). (More on cross-references of images in Image#edit[])

Also there is another caption right underneath the navigation that allows the user to quickly jump to edit form of this page.

[add screenshot of upload progress]

Another additional feature is the ability to directly upload pictures to the page by simply dragging picture files from the file system onto the page. A progress dialog will open up that shows the user how many files are left to get uploaded and the images will after the upload directly get added to the page so that the user can work with them on the page right away.

#### 3.5.4 Page Edit Form

[add screenshot of edit form]

On this page the user is able to edit several aspects of a page like its title or its description. Changing the title of a page also leads to the creation of a new URL-slug[[8]](#footnote-8) for this page, so that the url and the title of a page always correspond. Underneath the normal form there is a listing of all assets that are associated to this page. When hovering one of these images, new controls to edit and delete the image fade in. Also another option fades in that lets the user set this image as the cover image. When this one gets clicked the image in the normal form automatically changes.

The user is furthermore able to add new images on this page directly by dragging them somewhere onto the page or by opening up the file dialog with the "Add asset" button. The uploaded images will then automatically appear in the asset list.

Other than on the overview pages the user here has to manually save changes with the buttons that are placed directly under the navigation. There also is a button to cancel the edit form that will remove all changes the user has made and will redirect the user to the page. The third button deletes the page.

#### 3.5.5 Image Overview

On this page the current image is shown in the original size as the user uploaded it. The image is centered horizontally and vertically so that the images' center lies on top of the pages' center. Like on the other pages, the image here can also get dragged around which is handy for images that are bigger than the browser screen so users can see the rest of each image by dragging it around. The position of an image is not saved to the server because the main focus on this page should not lie on a specific arrangement but on the image itself.

By pressing the right- or left key, the user can navigate through the rest of the images of the current page to quickly get an overview over all images.

#### 3.5.6 Image Edit Form

### 3.3 Drag and Drop

### 3.4 Quo vadis Salon?

### 3.5 Evaluation

1. http://rubyonrails.org/ [↑](#footnote-ref-1)
2. http://betterexplained.com/articles/intermediate-rails-understanding-models-views-and-controllers/ [↑](#footnote-ref-2)
3. http://www.mongodb.org/ [↑](#footnote-ref-3)
4. http://en.wikipedia.org/wiki/Document-oriented\_database [↑](#footnote-ref-4)
5. http://de.wikipedia.org/wiki/Representational\_State\_Transfer [↑](#footnote-ref-5)
6. http://en.wikipedia.org/wiki/JavaScript\_Object\_Notation [↑](#footnote-ref-6)
7. http://en.wikipedia.org/wiki/Breadcrumb\_(navigation) [↑](#footnote-ref-7)
8. http://en.wikipedia.org/wiki/Slug\_(web\_publishing) [↑](#footnote-ref-8)