

CUSTOMER ACCOUNT MANAGEMENT

A Web Application Project for the Media Programming Course

by Anja Kammer in collaboration with Asiel Novia Tjahjana

Assignment 1

https://github.com/tjahjana/customer-account-management/tree/anja_branch

1. Project vision

An application that supports the customer care department of a company to manage all contracts and customers. They can search for the customer to get information about the billing details of contracts and products the customer pays for. Furthermore it is possible to book new features for the products and list them as a part of the contract fee.

Not yet part of this project is the providing of a product database to choose from, while maintaining a customer's contract. All contract, product and feature details must be created from scratch and are not reusable for other customers.

2. Product Backlog

- the user can create and edit customers, contracts, products and features
- the user can list all contract details of a customer
- the user can search for customers, contracts, products and features
- the user can mark a contract as billed, paid and canceled
- the user can search for contracts which are overdue in billing and payment
- the user can add, edit and delete other users

3. Simple CRUD operations story

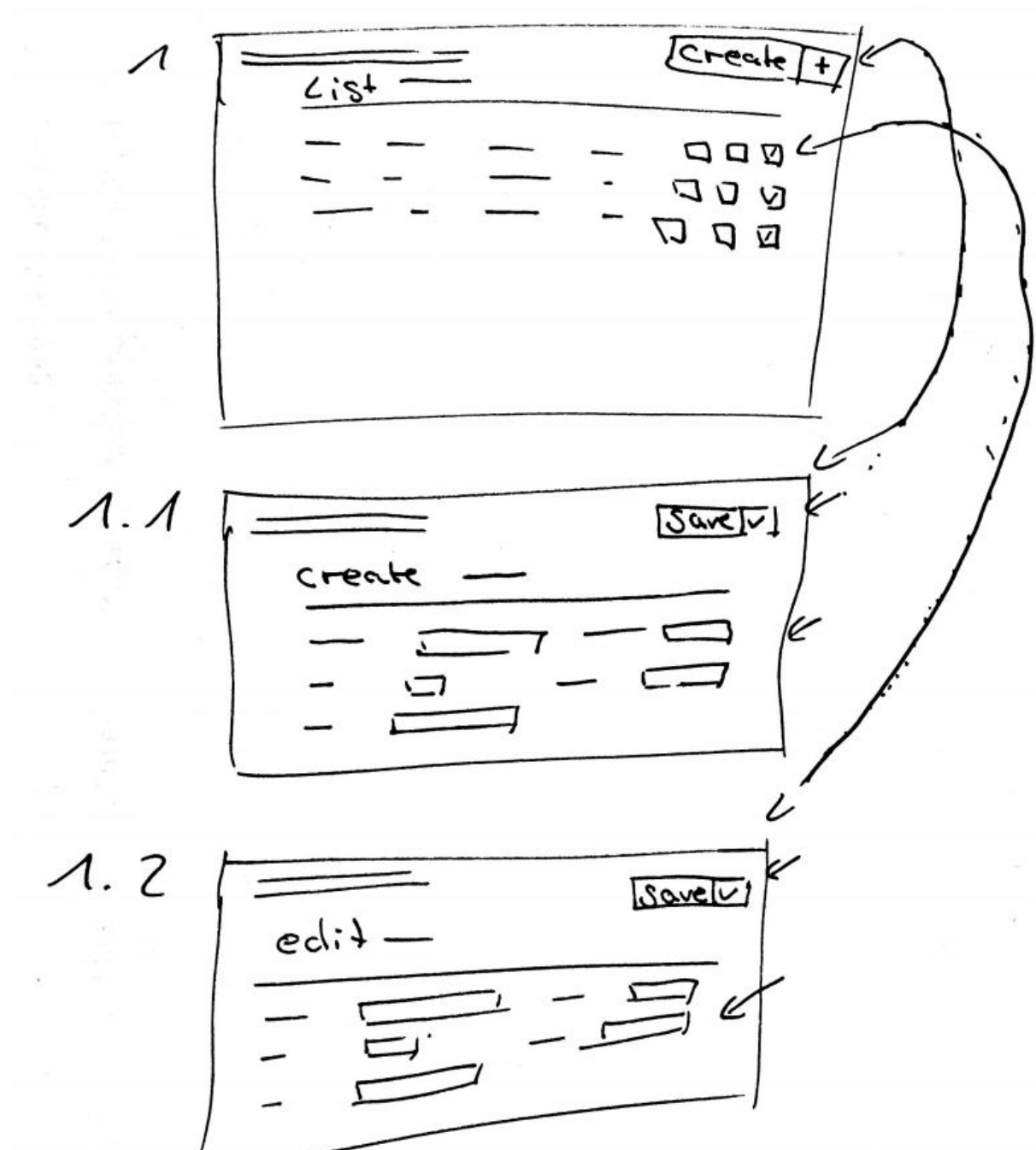
The user can create and edit customers, contracts, products and features.

CUSTOMER ACCOUNT MANAGEMENT

A Web Application Project for the Media Programming Course

by Anja Kammer in collaboration with Asiel Novia Tjahjana

4. Screens involved



CUSTOMER ACCOUNT MANAGEMENT

A Web Application Project for the Media Programming Course

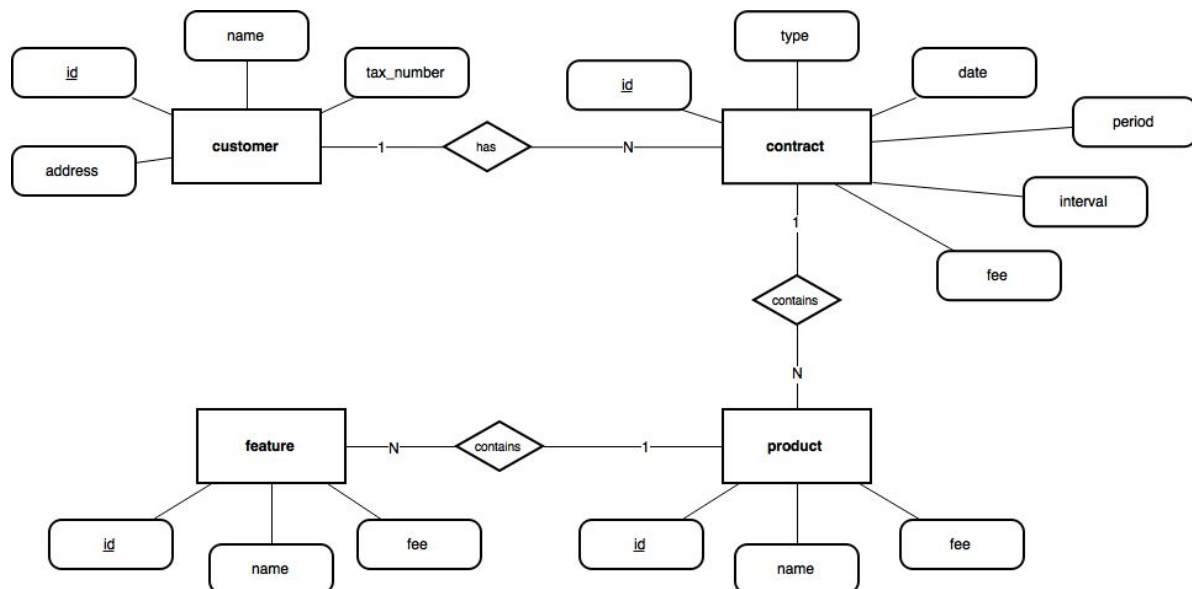
by Anja Kammer in collaboration with Asiel Novia Tjahjana

5. Data model

1 customer has N contracts

1 contract contains N products

1 product contains N features



6. Rails application

To create a rails application it was needed to install rails. Therefore the windows railsinstaller¹ was in use. The Application was created by running the following command:

```
> rails new customer-account-management
```

After that, it was not possible to run the rails server by using the command:

```
> bin/rails server
```

There were errors shown, that the uglifier app was not able to download for the installation process. So the Gemfile was checked. The uglifier app was not the only gem which were not able to get installed. Therefore it was needed to install the affected gems manually by running:

```
> gem install gem-name -v '11.3.0' -r --source http://rubygems.org/
```

After installing all required gems, it was possible to run the rails server.

¹ railsinstaller.org

CUSTOMER ACCOUNT MANAGEMENT

A Web Application Project for the Media Programming Course

by Anja Kammer in collaboration with Asiel Novia Tjahjana

7. model/view/controller

Scaffolding

To generate the models, views and controller of the entities customer, contract, product and feature, the scaffolding was used. The following commands create all required files:

```
> rails generate scaffold customer name:string address:string  
tax_number:string
```

```
> rails generate scaffold contract type:string date:date period:string  
interval:string fee:integer
```

```
> rails generate scaffold product name:string fee:integer
```

```
> rails generate scaffold feature name:string fee:integer
```

After that, the database is getting set up by running:

```
> rake db:migrate
```

This command creates a SQLite3 database with all required tables which were defined by its models in db/migration and the database scheme from the file db/schema.rb.

Rails server

It was not possible to enter the application through the server, because there was no root page set. For now, it is sufficient to provide the customer index page as default welcome page. This resource helped me out with this issue:

http://docs.railsbridge.org/intro-to-rails/setting_the_default_page

I added the following root page to the config/routes.rb:

```
root 'customers#index'
```

ExecS::ProgramError

Furthermore, I needed to add a new gem to the Gemfile after an ExecJS::ProgramError.

```
gem 'coffee-script-source', '1.8.0'
```

Then it was needed to run bundler to install this gem:

```
> bundle update coffee-script-source
```

CUSTOMER ACCOUNT MANAGEMENT

A Web Application Project for the Media Programming Course

by Anja Kammer in collaboration with Asiel Novia Tjahjana

1 initial List View

Listing Customers

Name	Address	Tax number
------	---------	------------

[New Customer](#)

2 Editing View

New Customer

Name

Address

Tax number

[Back](#)

3 Confirmation View

Customer was successfully created.

Name: BestCustomer

Address: BestAddress 1

Tax number: 123456_best

[Edit](#) | [Back](#)

4 Updated List View

Listing Customers

Name	Address	Tax number	
BestCustomer	BestAddress 1	123456_best	Show Edit Destroy

[New Customer](#)

CUSTOMER ACCOUNT MANAGEMENT

A Web Application Project for the Media Programming Course

by Anja Kammer in collaboration with Asiel Novia Tjahjana

8. html templates

To edit the customer's address, a text field is required, which provides more space to display an address. Therefore a small change was made in the form template of the customer's view. The text_field was changed into an text_area. This small change affects all views which contain the customer form.

```
<div class="field">
  <%= f.label :address %><br>
  <%= f.text_area :address %>
</div>
```

9. User story that requires functionality

- the user can add, edit and delete other users

10. Creating a new controller action and route by hand

A new table is required called 'user' which hold the user's email address and a password field. the following files were needed to create:

- db/migrate/20161017170836_create_users.rb
- views/users/*
- models/user.rb
- controllers/users_controller.rb

The config/routes.rb file needed one addition:

```
resources :users
```

The last step was to run the migration script by using:

```
> rake db:migrate
```

Listing Users

Email

post@anjakammer.de [Show](#) [Edit](#) [Destroy](#)

[New user](#)

5 hours