Project Document
Easy Link Keeper

Jort van Gorkum - 6142834
Niek Geijtenbeek - 6214096
Tijmen van der Spijk - 6241514
Madio Seck - 5921902
Vince van Noort - 6187021

28-11-2017

Table of contents

voorbeeld: http://www.cs.uu.nl/docs/vakken/b1pica/Materiaal/vbprojectdocument.pdf

Introduction

We will design and create an application, on behalf of Canon, which will visualise and (possibly) evaluate the business relationships between Canon and another company. In order to achieve this we will be using AdonisJs as our framework for our webapplication. Alongside the application we will prepare a presentation, a website to promote our web-application and deliver several other documents.

The reason why we are making a web-application, because it is more accessible for users with different devices in comparison to other applications. This means that more people will be able to use the application and with greater ease.

Purpose:
The purpose of this document is to give a planning for the entire development process. This planning is not definitive and therefore subject to change. This document also describes the system: the analysis of functionalities as well as the system design.

Planning

Goal

The result of our project should be a programme which will visualise and evaluate the business relationship between Canon and another company. With this we will mainly focus on building up the relationship and not maintaining it. The primary goal of our project is to make an application that clearly visualises the relationship with a company, which can be used to close a deal. As secondary goal we would also like to add that the program could advice on a relationship based on the information given by the users.

Development Method

AdonisJS will be the main framework for our program. This framework will make it much easier to build a web-application. Also authentication will be implemented by AdonisJS, so we can easily use it in our program.

Milestones

1.	Temporary project document	
2.	Temporary interface	
3.	Setting up basic database	
4.	Finished Project document	6-12-2017
5.	Temporary website	Week 50
6.	intermediate product	Week 50
7.	Intermediate presentation	20-12-2017
8.	Finished Interface	
9.	Finish Must haves	
10.	Finish Should have	
11.	Final Report with manual	23-01-2018
12.	Finish app	24-01-2018
13.	Final Presentation	24-01-2018
14.	Demonstration	31-01-2018
15.	Submit Projectfile	31-01-2018

Division of tasks

Project paper:
- Introduction: Madio, Jort & Tijmen
- Planning: Madio & Tijmen
- Analysis: Niek & Jort
- Design: Vince

Analysis

Requirements

The requirements our web application needs:

- Taking care of the backhand of the web application.
- A server to host the web application.
- A database to store the data for the web application.
- Unit testing functionality.
- Version Control (GIT)
- Sandbox environment for testing branches.
- Node.JS / other framework(s)

MoSCoW

- **Must haves:**
  - Having a graphical interface designed for easy access.
  - Adding credential authentication.
  - A database to store information related to the application.
  - A website to access our web-application.
  - Basic functionality, being able to overview and edit related data.
  - Creating a warning system voor conflicting data.
  - Being able to add a company.
    - Basic information related to goals of the company.
    - Basic information related to the end deal.
    - GROW information.
  - Being able to add a person of interest.
    - Their title, role, needs, influence and other information.
    - Their stand on your proposal.
    - Both personal and business goals.

- **Should haves:**
  - Give vital advice based on individual relationships.
    - Example: missing information.
  - Granting a score based on an individual relationships.
  - Being able to define relationships in different tiers.
    - Based on priority for example.
  - Being able to give your own priority to relationships.
  - Being able to add social media information to a person of interest.
    - LinkedIN, Facebook and other related social media platforms.
  - Being able to add multiple proposals instead of one and giving feedback.
    - More detailed information added to proposals.

- **Could haves:**
    - Creating an overview of all companies in a searchable manner.
        - Search for keywords such as names or companies.
    - Giving feedback bases on relations compared to others.
        - Time based for example.
    - Animated input/output, trying to create a less linear way of input.
        - Example: priority input based on pressure dials.
        - Animated graphs showing overview over relation(s)
    - Adding chat functionality.
    - Adding notes to a proposal, keyword evaluation.
        - Extra information vital to a proposal.
    - Adding encryption not only on exchange but also stored information.

- **Won't haves:**
    - Creating functionality for signing in.
    - Custom encryption between user and server (we'll used HTTPS).
    - Connecting to a database, we'll use a framework for that.
    - General functionality that's already implemented by Node.JS
    - A plan to colonize Mars (Elon Musk is already doing that for us).

Design

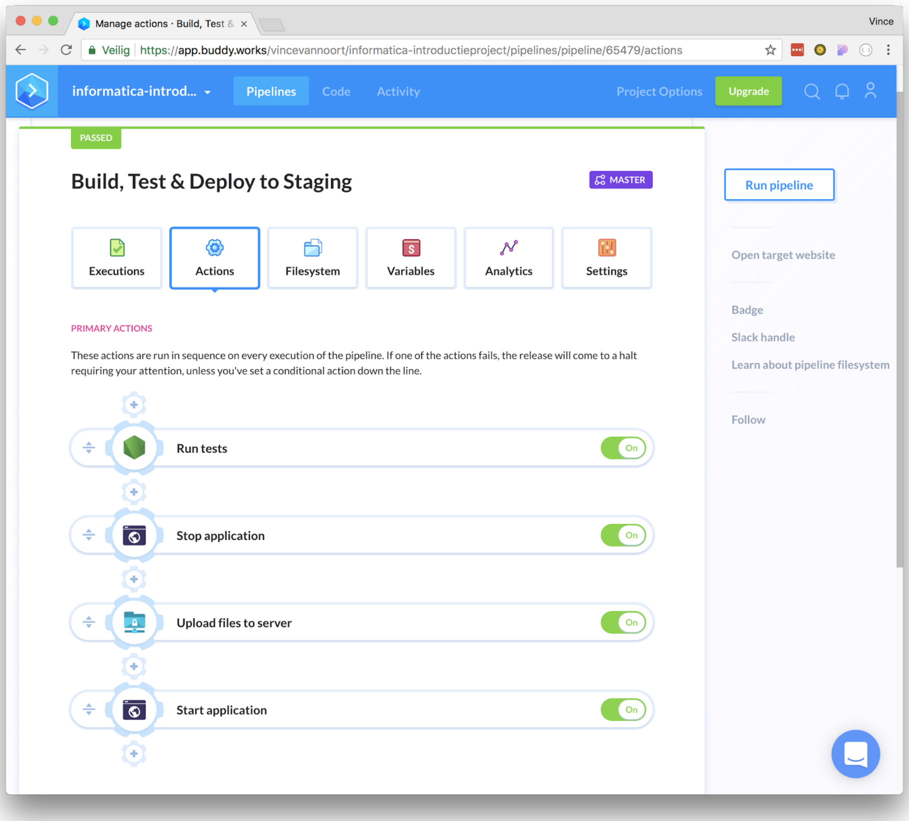
Technical design

Platform

We have chosen to build the project as a web application. The reason behind this is that a web application is platform independent, is easily reachable from anywhere and is easy to update. The application is divided in a backend and a frontend. The backend will be build in AdonisJS, a node-js framework based on the MVC model. The frontend will be build in Vue.js, a progressive framework for building user interfaces.

Development environment

To make developing together less tedious we use a version control system called GitHub. Github makes it easy for everyone to work locally and to merge changes together. Every feature will be developed in a separate branch and eventually is merged together when de feature is ready. Besides a version control system the app will be developed using a continuous integration and build system called Buddy. Buddy will allow us to continuously test and build the application in a sandbox environment. When a commit is tested and build successfully the application is also deployed to a digitalocean server via ssh commands and sftp file transfer. Because Buddy deploys to the server, no one from the team has to be bothered with deploying to the server because Buddy already takes care of this rather tedious task.



Build, Test & Deploy to Staging

PRIMARY ACTIONS

These actions are run in sequence on every execution of the pipeline. If one of the actions fails, the release will come to a halt requiring your attention, unless you've set a conditional action down the line.

Run tests — On

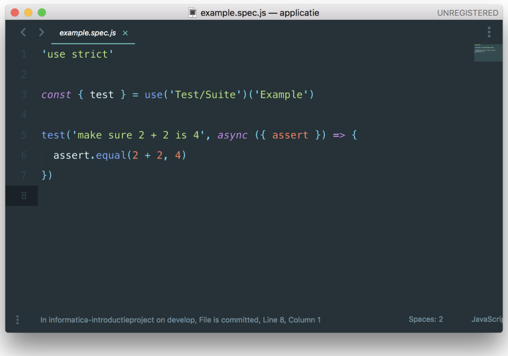
Stop application — On

Upload files to server — On

Start application — On

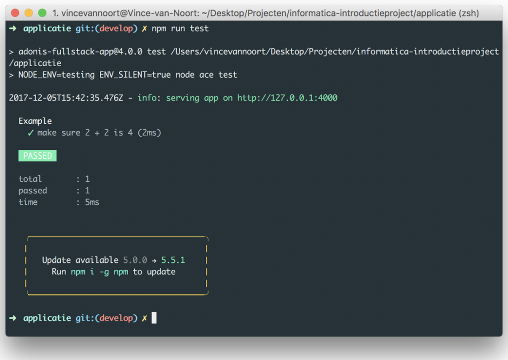
Testing

AdonisJS comes provided with a testing framework called VOW (http://vowsjs.org/). The framework is built for test driven development. The test cases will be written based on the requirements set by Canon and our own group. The 2 images beneath describe how test are written (left) and how test are run (right).



```
'use strict'

const { test } = use('Test/Suite')('Example')

test('make sure 2 + 2 is 4', async ({ assert }) => {
  assert.equal(2 + 2, 4)
})
```

example.spec.js — applicatie
UNREGISTERED
example.spec.js
In informatica-introductieproject on develop, File is committed, Line 8, Column 1
Spaces: 2
JavaScri

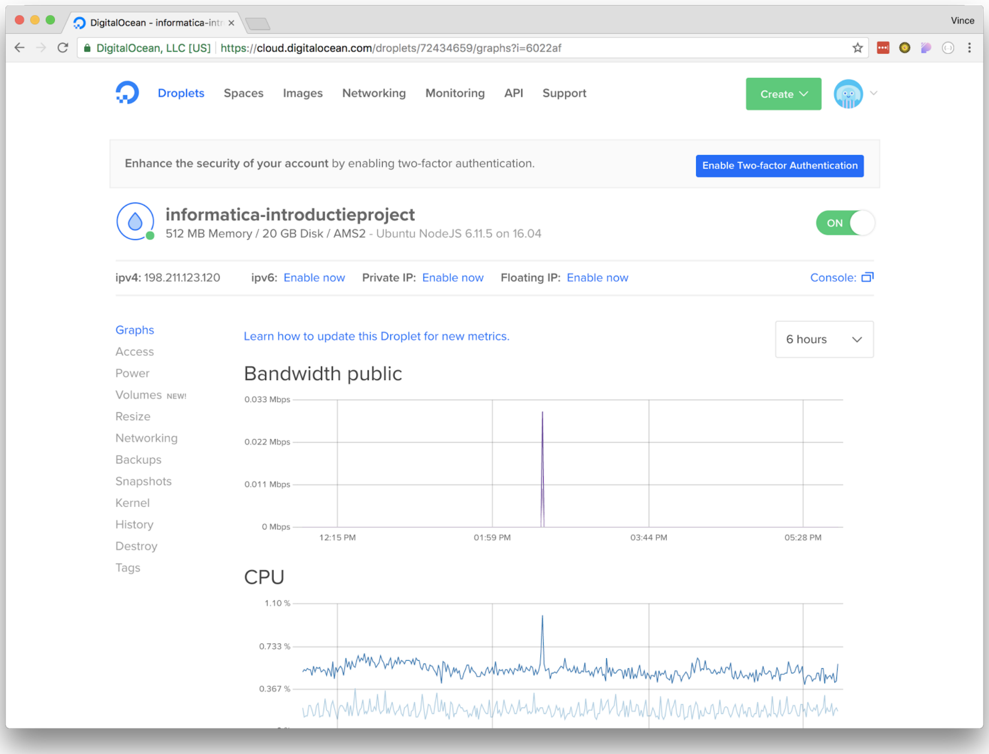


```
➜ applicatie git:(develop) ✗ npm run test

> adonis-fullstack-app@4.0.0 test /Users/vincevannoort/Desktop/Projecten/informatica-introductieproject
/applicatie
> NODE_ENV=testing ENV_SILENT=true node ace test

2017-12-05T15:42:35.476Z - info: serving app on http://127.0.0.1:4000

  Example
    ✓ make sure 2 + 2 is 4 (2ms)

  PASSED

  total       : 1
  passed      : 1
  time        : 5ms


   ┌────────────────────────────────────┐
   │                                    │
   │   Update available 5.0.0 → 5.5.1   │
   │     Run npm i -g npm to update     │
   │                                    │
   └────────────────────────────────────┘

➜ applicatie git:(develop) ✗
```

1. vincevannoort@Vince-van-Noort: ~/Desktop/Projecten/informatica-introductieproject/applicatie (zsh)

Backend

As mentioned before we've chosen for AdonisJs to develop our backend. AdonisJs will connect to a SQLite database for easy development. If the application tends to get bigger the switch to MySQL will be seamless due to AdonisJs having a transparent database interface. SQLite allows us to develop locally without creating a MySQL server. AdonisJs is responsible for our data structures, authentication, backend-validation and API.



Enhance the security of your account by enabling two-factor authentication.

informatica-introductieproject
512 MB Memory / 20 GB Disk / AMS2 - Ubuntu NodeJS 6.11.5 on 16.04

ipv4: 198.211.123.120 ipv6: Enable now Private IP: Enable now Floating IP: Enable now Console:

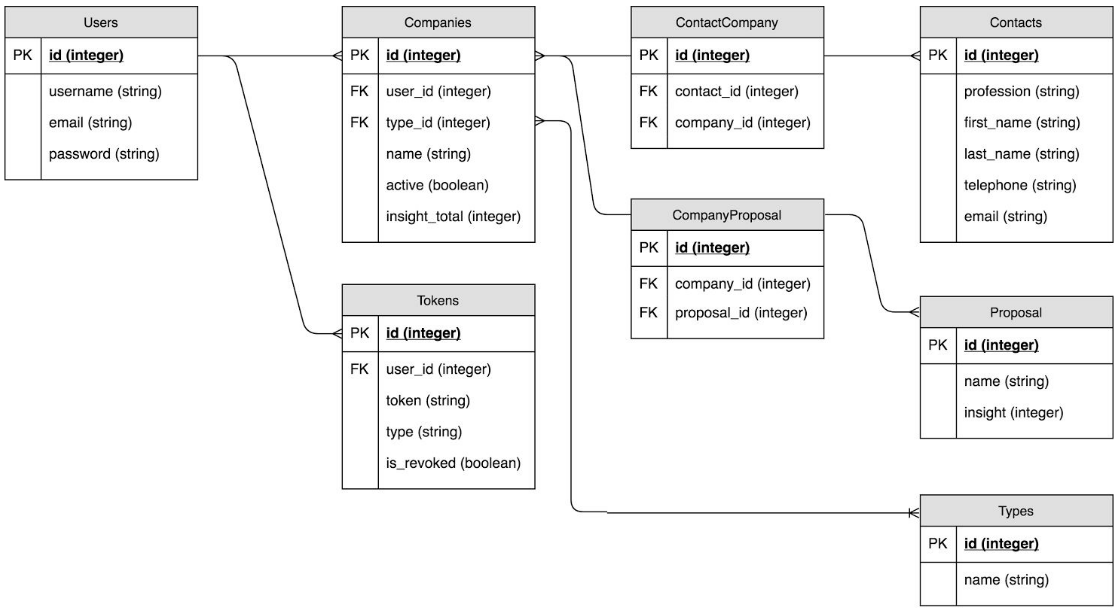
Learn how to update this Droplet for new metrics.

Frontend

For the frontend side we have chosen Vue.js as our javascript framework for developing the user interface. Vue.js will be responsible for each component and its functions. The framework comes with a frontend router and a webpack loader. The frontend router will allow us to make a single page application (https://github.com/vuejs/vue-router). This delivers a smoother UX to the end user. The webpack loader allows us to develop each components with its markup, styles and functionality in one single file (https://github.com/vuejs/vue-loader).

Model relations

The relation model beneath describes how each entity is connected with each other. These models will be used when building the application in AdonisJS in a MVC manner.

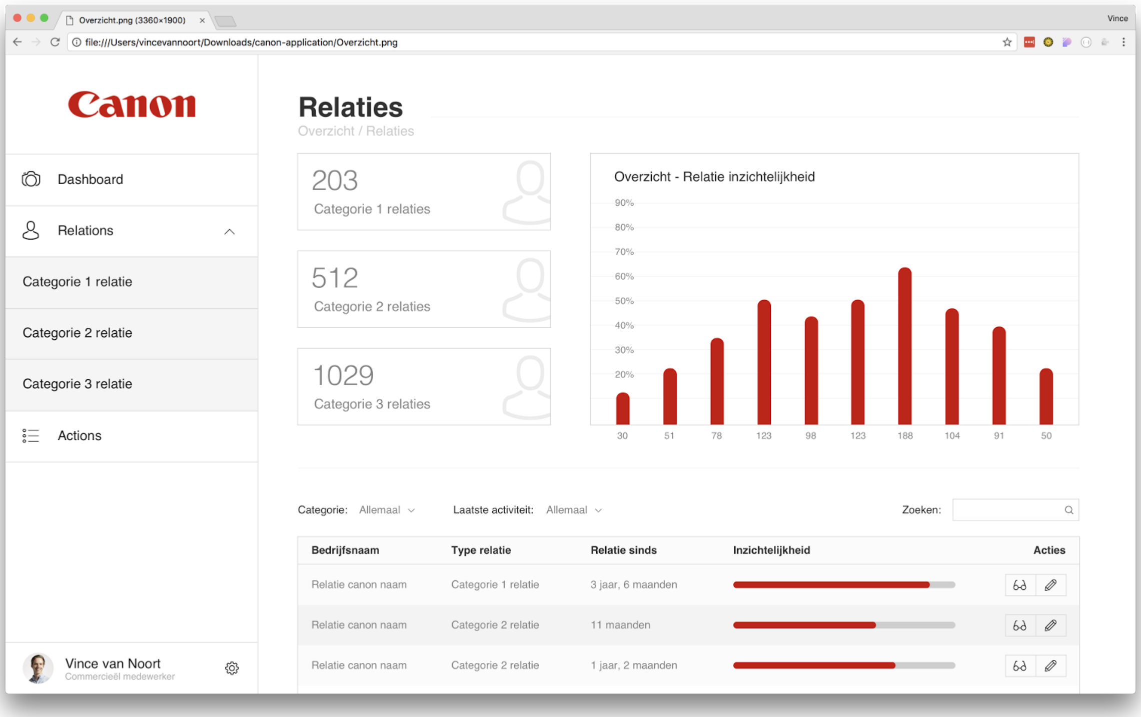


Pages

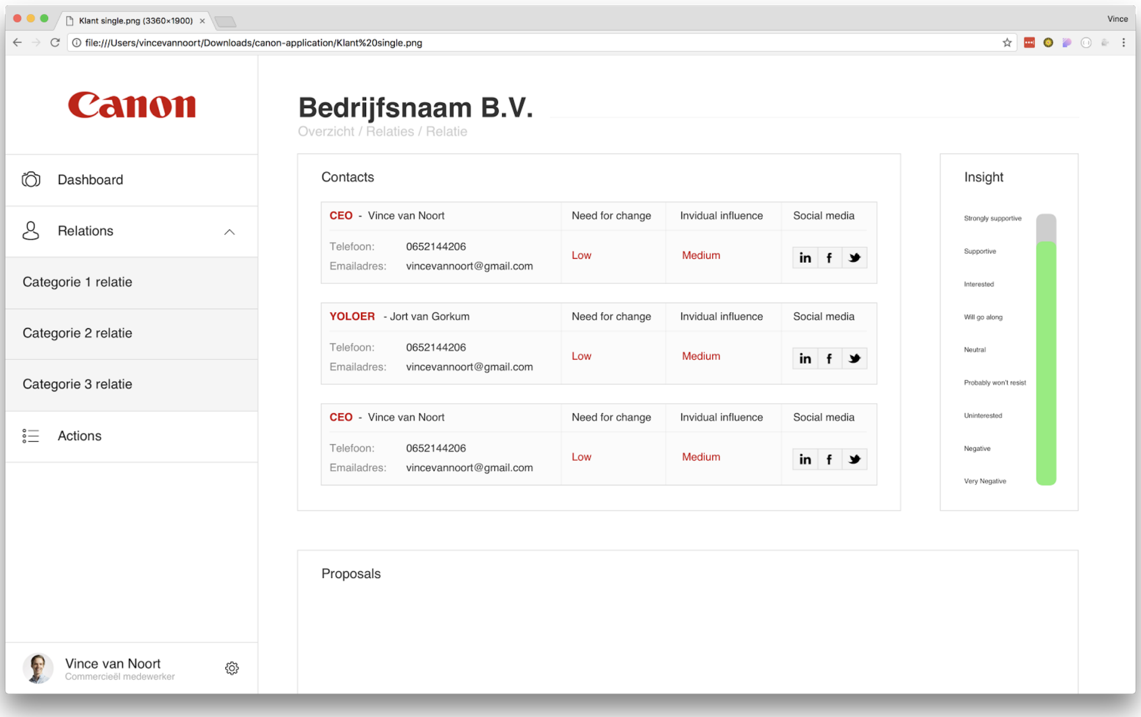
- login - dashboard - insights	- users - user detail - user create/edit	- companies - company detail - company create/edit	- contacts - contact detail - contact create/edit	- proposals - proposal detail - proposal create/edit

Graphic design

The graphic design will be partly based on the colors that canon is already using in their style. The design consists of a main layout with separate components per page. The current design will be the basis for the other not yet designed pages, but should provide a good look and feel for the final product.



Bedrijfsnaam	Type relatie	Relatie sinds	Inzichtelijkheid	Acties
Relatie canon naam	Categorie 1 relatie	3 jaar, 6 maanden		
Relatie canon naam	Categorie 2 relatie	11 maanden		
Relatie canon naam	Categorie 2 relatie	1 jaar, 2 maanden		



	Need for change	Invidual influence	Social media
CEO - Vince van Noort Telefoon: 0652144206 Emailadres: vincevannoort@gmail.com	Low	Medium	in f
YOLOER - Jort van Gorkum Telefoon: 0652144206 Emailadres: vincevannoort@gmail.com	Low	Medium	in f
CEO - Vince van Noort Telefoon: 0652144206 Emailadres: vincevannoort@gmail.com	Low	Medium	in f