Skill Matrix documentation



# Introduction

In early November we have been told that we were going to start a new project for our education in Application Development. Our teacher, Sander, had a project of a friend of his for a company called Agilent Technologies.

Sander explained us his thoughts of the project so far and prepared us for what we could expect of it. Our employer will be Stephaan van Heerwaarde. Stephaan is the head of Agilent Technologies in the Netherlands.

Agilent Technologies is a company that provides analytical instruments, software, services and consumables for the entire laboratory workflow. Agilent focuses its products and services on six markets: food, environmental and forensics, pharmaceutical, diagnostics, chemical and energy and research.

The task is to create an application which can replace the current “Skill Matrix” Agilent Technologies is using. Agilent Technologies has engineers who have to go to different locations for installing equipment or perform maintenance on their installed equipment. The engineers all have different skills, this implies that an engineer is only allowed to work with a certain piece of equipment if the engineer is trained for this product. The company keeps track of who is trained for what machines in the current “Skill Matrix”. This application is mostly used by their planners to know which engineer to send where.

The problem with the current “Skill Matrix” is that it’s outdated. The program has been created around 10-15 years ago and a lot of features that are currently in it aren’t being used anymore. We are going to create a new application to replace the current “Skill Matrix” and add the features they would like to have built in it.

# Meetings with Agilent Technologies

## Introduction

While working on this project we had different meetings with Agilent to discuss our progress and answer their questions. To remember what we were discussing we wrote down everything in notes. Most of the time we wrote down simple keywords and sentence’s. Down here are the extended notes we took of the meetings.

## 2-11-2015: Meeting at Agilent

Agilent is a company that supplies medical equipment and maintains the equipment. E.g. appliances temperature, gas, food measuring values. Stephaan is the boss of Agilent in the Netherlands.

Agilent has planners who can plan in the engineers to maintain certain equipment and / or get it replaced. This is now done by a web planner which stands on an internal server. This server is in Brussels and is approx. 15 years old. In this web planner is a skill matrix. This is a list of the products and the engineers who are trained for this. Each engineer has therefore its own products for which he was trained. Engineers can get there through skills training. Engineers and products can also be added.

Agilent wants an updated version of their current skill matrix.

The program should be very user friendly so that anyone without a technical background will understand it. They want to see an image of the selected device, and who are trained to handle this device. This also goes for the engineers. Where you see a picture of the engineer, with the level, other information and which engineer is trained.

This is called the front-end of the program.

They also want to see an admin page where you can modify all the data. This is called the back end. Here you can adjust certain data of the engineer, products and admins.

Between each meeting designs are made based on the information given

## 9-11-2015: Conference call 1

In this meeting we received information on the front of the page.

Certain aspects that need to be added, for example, more information for the engineer.

This includes a drop point code, telephone number, e-mail, a personal number, a VCA number, SCC achieved date, registration of car, car make, model and colour, street, street number and zip code, city / town, country of origin spoken languages.

They also wanted buttons to organize the engineers on land, name etc.

The back-end should also be edited. We were told that the products had a category and subcategory. We also had a separate button when creating new products for "Cross lab"

## 13-11-2015: Conference call 2

We have shown the design of the front end and until so far Stephaan (our contact / boss of Agilent NL) was very satisfied. He also did have a few things to add.

Stephaan wants flags next to the engineers, so that you can see where they come from.

This is in the list of available engineers, and not in the information box. We have also discussed that there should be notes on mouse-over in the list of available engineers if the person is sick. There also needs to be a difference between SFE and ASP. Only thing that we need to take note of is that ASP is listed. The rest of the engineers are SFE.

For the rest we talked more about a data dump from their database so we know what their structure looks like. If we get that, we can easily test things.

At the next meeting we are not only showing the custom front-end design, but also the first back-end design.

## 16-11-2015: Conference call 3

In this meeting we planned to present the back-end design and discuss it. But because the head of planning in Belgium also was at the call, we had to explain a lot over again. As a result we did not get the time to discuss the back end.

We discussed some minor adjustments to the front end.

We wanted to have a date system in the comment, where it will show a date when it begins and ends. This comment then disappeared as the deadline had expired.

Also there was another name to the engineer: FFS.

We had an idea to add automatic copy buttons to data in the data-box of the engineers. There were also minor adjustments of the display of that information.

In the next meeting we are really going to discuss the back-end design.

## 20-11-2015: Conference call 4

In this meeting we discussed the backend design.

In the backend, we had mentioned three different menus. Engineers, Products and Users. It was proposed to add an extra menu called Categories. Here you can choose which categories are there and which have subcategories. We also discussed certain agreements about the categories. Like the way it was written.

We discussed that different rights come with the different users. Ex. An admin has read / write permissions. There also was the idea to put an engineer at idle if it is not available or something. This means he is no longer shown on the front end, but he will still be in the database.

In this meeting Stephaan also approved the designs so we could begin coding.

# Requirements

## What are we going to build?

We are going to build an application which is going to replace the current Skill Matrix Agilent Technologies is using. The goal is to create an application which can be used to see which engineers are trained for which products. The application is mostly going to be used by the planners, so it has to be easy to use so you can find the desired engineer for the desired product as quick as possible.

The new Skill Matrix is going to contain a search bar which lets you search for an engineer or a product in the same field. Once you press in a key or a letter the results will automatically be filtered.

If you click on a machine you will see all the engineers trained for product. The details of the machine will be shown next to the list of products. Once you press an engineer which is trained for the system, you will see all the engineers information.

It’s also possible to click on an engineer. Once you press the engineer you will see all the products the clicked engineer is trained for.

## The requirements

HTML & CSS  
We are going to create a web application. The styling and frontend will be done in HTML and CSS. For the styling we are going to use Bootstrap. For more information about the styling see the chapter “Style definition”.

### Python

The programming language we are going to use for the backend and most of the application is Python. To create the application we are going to use Django, which is a framework for Python.

### Database

We are also going to need a database to store all our data as shown below in the ‘Data Definition’. We are probably going to use a MySQL database which is relational.

### Server

To host the application we are going to need a server. Agilent Technologies will find a provider to host the server. The application will be used at multiple locations so it has to be reachable from a different location.

# Style definition

## This describes the basic style of the application.

## Colours

* #0085D5 – Primary. (Hyperlinks, buttons, accents)
* #DDDDDD – Borders.
* #333333 – Text.
* #FFFFFF – Background.
* #F5F5F5 – Alternative Background

## Whitespace

Large space: 30px. This space is used between blocks.  
Medium space: 20px. Is used between the border of a block and the content.  
Small space: 10px. Is used between the content of certain elements.

Rounded corners  
Input-elements and buttons: 4px for all corners.  
Other elements: None.

## Typography

### All text

Line Height: 1.428  
Letter type: Open Sans (regular 400)

### Standard text

Letter size: 14px

### Headlines

Letter size header 1: 28px  
Letter size header 2: 22px  
Letter size header 3: 16px

## Icons

We will use the icons of Font Awesome.

# Data definition

For the users we will use the build in system of Django (django.contrib.auth). Django has the following models for this system: User, Group, Permission, ContentType.

## Engineers

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Type** |
| Employee number | id | Integer  Primary key |
| First name | first\_name | Varchar (100) |
| Last name | last\_name | Varchar (100) |
| Droppoint | droppoint | Char (20) |
| Phone | phone | Char (11) |
| Email | email | Varchar (255) |
| VCA number | vca\_number | Char (19) |
| VCA date | vca\_date | Date |
| Car type | car\_type | Varchar (100) |
| Car color | car\_color | Varchar (100) |
| License plate | license\_plate | Varchar (10) |
| Street | street | Varchar (100) |
| Zip code | zip\_code | Varchar (10) |
| City | city | Varchar (100) |
| Country | country\_id | TinyInt |
| Is active | is\_active | Boolean |

## Skills

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Data type (length)** |
|  | id | Integer  Primary key |
| Engineer | engineer\_id | Integer  Index (engineer\_id, product\_id) |
| Product | product\_id | Integer  Index |
| Skill level | level | TinyInt |

## Products

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Data type (length)** |
|  | id | Integer  Primary Key |
| Category | category\_id | Integer |
| Model | model | Varchar (100) |
| Type | type | Varchar (100) |
| Crosslab | is\_crosslab | Boolean |
| Is active | is\_active | Boolean |

## Categories

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Data type (length)** |
|  | id | Integer  Primary key |
| Short name | short\_name | Char (2) |
| Full name | name | Varchar (100) |
| Parent category | parent\_id | Nullable Integer  Index |

## Notes

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Data type (length)** |
|  | id | Integer  Primary key |
| Engineer | engineer\_id | Integer |
| Content | content | Varchar (255) |
| Shows from this date | visible\_from | Nullable date |
| Shows until this date | visible\_until | Nullable date |
| Created at | created\_at | DateTime |

## Countries

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Data type (length)** |
|  | id | Integer  Primary key |
| ISO Code | Code | Char (2) / Char (3) |
| Full name | name | Varchar (100) |

## Languages

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Data type (length)** |
|  | id | Integer  Primary key |
| ISO Code | Code | Char (2) / Char (3) |
| Full name | name | Varchar (100) |

## Engineer languages

|  |  |  |
| --- | --- | --- |
| **Name** | **Column** | **Data type (length)** |
|  | id | Integer  Primary key |
| Language | language\_id | Integer |
| Engineer | engineer\_id | Integer |

# Feature list front-end

## Front-Page

1. **Search bar**

*What is it?*

A Search bar

*What is its purpose?*

Looking for engineers and / or products

*How does it work?*

* Seeks the name of the engineer and the machine simultaneously
* Results will all appear as you type.
* Can filter the results by country and more.



1. **Product list**

*What is it?*

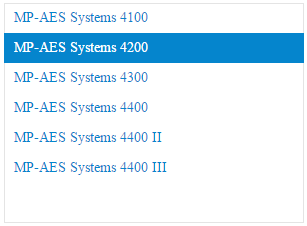
A list of names

*What is its purpose?*

The display of products that is sought out for.

*How does it work?*

* List of products that are sought after, which depends on what you type in the search bar.
* On a mouse-over of a name, change the colour of the selection to make it clear that you have selected this name to blue.
* There will be a picture of the product to the right of the product list if the product has been clicked on. In the engineer list you will also be able to see what skills the engineers have.



1. **Product information**

*What is it?*

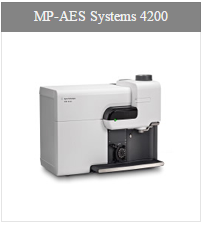
A field of information

*What is its purpose?*

Displaying information about the product you just selected

*How does it work?*

The information of the selected product will be shown. This includes the name and the image of that product.



1. **Engineer list**

*What is it?*

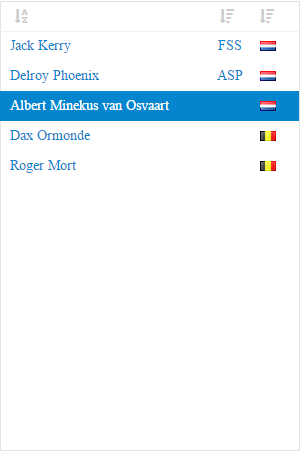
A list of names

*What is its purpose?*

To display the engineers that are searched for

*How does it work?*

* A list of engineers which will reduce the more you type in the search bar.
* The colour of the space in which you clicked will change to blue to show that you clicked on that name.
* There is also a pop-up notification on mouse over of a name. This shows the status of this engineer (Sick, Vacation, etc.)
* Next to the engineers name is also a title which include ASP and FFS. In addition, the flag of their country of origin is also shown.



1. **Engineer Information**

*What is it?*

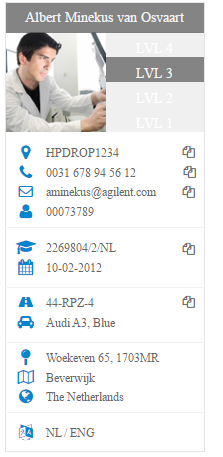
A list of names

*What is its purpose??*

Displaying information about the engineer you just selected

*How does it work?*

If an engineer is selected there will be an info block with the following information:

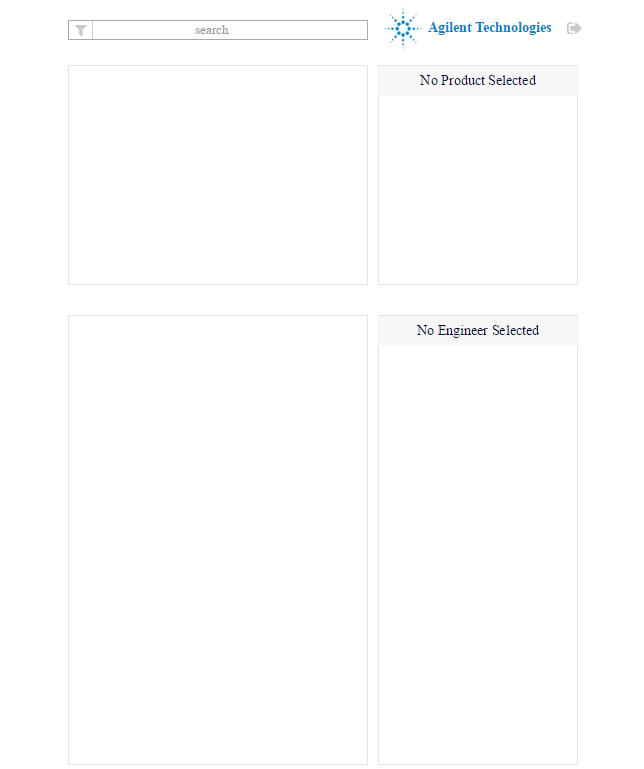
* The name
* the level on the current selected product
* a picture
* a drop point
* a telephone number
* an e-mail address
* the employee ID
* the SCC
* the date when the VCA was achieved
* the license plate number of the used car
* the car brand
* his/her street
* his/her zip code
* his/her street number
* the city
* the country
* the spoken languages

Also, if an engineer is selected, the devices he/she is trained for will show up in the “Products Field”.

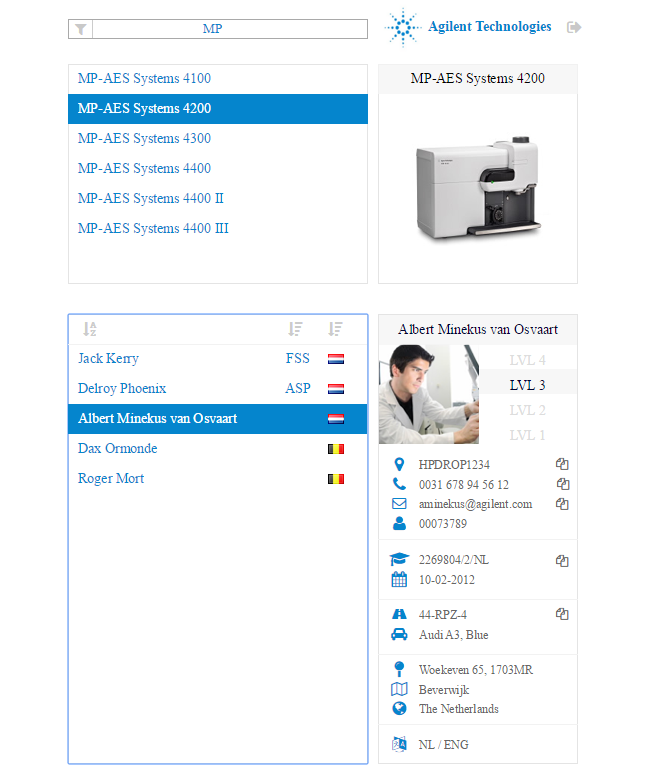
\*Front end 1 is if you didn’t select anything

\*Front end 2 is if you selected everything.

Front end 1



Front end 2



# Feature list backend

This is an overview of all the features in the backend of our project for Agilent Technologies. In this feature list you will find all the features with the corresponding images. The features are split up in four different sections. These contain the following categories and their sub-categories:

* Engineers
* Products (with 7 sub-categories)
* Categories
* Users

## Engineers

**1. Navigation**

*What is it?*

A button which links to a different page and where the current page is highlighted.

*What is its purpose?*

Navigating a user to a different page in the backend of the application.

*How does it work?*

You press on the desired page you would like to visit with the left mouse button. After you click it the application will take you to this page.

**2. Search**

*What is it?*

An input field in which you can search for an engineer, either by only typing a few letters or the engineer’s full name.

*What is its purpose?*

To look for an engineer which you would like to edit or view the details of.

*How does it work?*

You press the field and type in a part of the name, or the full name of the desired engineer. The search bar searches without reloading the page. You can search for first and / or last name of an engineer. By default all the engineers will be listed.

**3. Add new engineer**

*What is it?*

A button on the top right side of the page which lets you add a new engineer.

*What is its purpose?*

Creating a new engineer.

*How does it work?*

You press on the button with the left mouse button. After you press the button a new form opens in a panel on the right side of the page. In this panel you can add a new engineer.

**4. Engineer**

*What is it?*

All engineers currently in the database will be shown here. If you click on the name of an engineer his information will show up in a panel on the right side of the page. If you hover over the name of an engineer you will see the current note of the corresponding engineer.

*What is its purpose?*

Listing all the engineers.

On click: Showing all the information of the engineer.

On hover: Shows the current note of the engineer

*How does it work?*

When the page is loaded al the engineers will be listed here. When you click on an engineer all his information will be shown in a panel on the right. When you hover over an engineer his current note will be seen in a small block. **(see 08)**

**5. Note**

*What is it?*

A button which opens a panel on the right side of the page where you can edit the current note of the engineer.

*What is its purpose?*

Editing the current note of the corresponding engineer.

*How does it work?*

When you click on the button the note panel will be opened on the right side of the screen. You can edit the current note here in a form. **(see 09)**

**6. Add / remove skill**

*What is it?*

A button next to the name of the engineer. When the button is clicked with the left mouse button it opens a panel with a form on the right side of the screen.

*What is its purpose?*

Editing or removing skills on an engineer.

*How does it work?*

When the button is clicked with the left mouse button it opens a panel with a form on the right side of the screen. In this panel you can add and / or remove skills on the engineer object. **(see 10)**

**7. Edit**

*What is it?*

A button which you can click to edit the details of the selected engineer.

*What is its purpose?*

To edit the details of the selected engineer.

*How does it work?*

You press the edit button next to the engineer with the left mouse button. Once you pressed the button a new panel opens on the right side of the screen with a form in it which lets you edit the details of the engineer. **(see 11)**.

**8. Input ‘Add new Engineer’**

*What is it?*

A form which lets you add a new engineer to the database.

*What is its purpose?*

Adding a new engineer to the application.

*How does it work?*

After you press the “Add new engineer” button with the left mouse button, a new form appears in a panel on the right side of the screen. In this form you can fill in all the information for the desired engineer and add the new engineer after this.

**8.1 First name:** the first name of an engineer.

**8.2 Last name:** the last name of an engineer.

**8.3 Email:** the email address of the engineer.

**8.4 Engineer ID:** the ID of the engineer.

**8.5 Phone number:** the phone number of the engineer.

**8.6 Spoken languages:** the spoken languages of the engineer.

**8.7 Address:** the country, city, postal code, street name and drop point of the engineer.

**8.8 Car:** the type, license plate and colour of the engineer’s car.

**8.9 VCA:** the code of the safety certificate and the date on which the engineer got the certificate.

**8.10 Save:** saves all the options noted in **8.1 till 8.9** to the engineer.

**9. Input ‘Note’**

*What is it?*

A form which lets you add or edit the engineers current note.

*What is its purpose?*

To add or edit the engineers current note.

*How does it work?*

After you press the Note button **(see 4)** the form opens on a panel on the right side of the page. In this form you can fill in all the details that are needed to create or edit the note

**9.1 Delete note:** Deletes the current note

**9.2 Current from:** Start date from current note

**9.3 Current till:** End date of current note

**9.4 Current note:** The note which is currently active

**9.5 New note:** Add a new note to the engineer

**9.6 New date from:** The start date of the new note

**9.7 New date till:** The end date of the new note

**9.8 New note info:** The information about the new note

**9.9** **Save:** Saves the new note **(9.5 till 9.8)**

**10. Input ‘Skills’**

*What is it?*

A form / field which lets you add or remove a skill and its level on an engineer.

*What is its purpose?*

Adding or removing skills and its level to an existing engineer.

*How does it work?*

After you press the ‘Skills’ button with the left mouse button **(see 5)** a new form opens in a panel on the right side of the page. In this panel you can add new skills to the engineer. It’s also possible to remove skills from the engineer and to edit the engineers current level on a skill.

**10.1 New skill:** You can fill in the new skill you want to add in this form. There will be a search bar to search for a skill, and a dropdown menu which lets you select the skill from this menu. You can also add the engineers current level on this skill.

**10.2 Remove skill / Edit skill:** You can press the arrow buttons next to the skill to edit the engineers current level on this skill. It’s also possible to remove the skill by pressing the small red cross next to the skill.

**10.3 Save:** Save the information you filled in in the fields in **10.1 and 10.2**.

**11. Input ‘Edit Engineer’**

*What is it?*

A form to edit an engineer.

*What is its purpose?*

Editing an existing engineer.

*How does it work?*

After you press the ‘edit’ button **(see 6)** a form opens in a panel on the right side of the page. All the current information of the engineer will be noted here and you can edit the details you want to edit. Afterwards you can press save to save your changes to the database.

**11.1 First name:** the first name of an engineer.

**11.2 Last name:** the last name of an engineer.

**11.3 Email:** the email address of the engineer.

**11.4 Engineer ID:** the ID of the engineer.

**11.5 Phone number:** the phone number of the engineer

**11.6 Spoken languages:** the spoken languages of the engineer

**11.7 Address:** the country, city, postal code, street name and drop point of the engineer.

**11.8 Car:** the type, license plate and colour of the engineer’s car

**11.9 VCA:** the code of the safety certificate and the date on which the engineer got the certificate.

**11.10 Is active:** a checkbox to check if the engineer is active or not. If the engineer is not active he won’t be found if you search for him.

**11.11 Save:** saves all the information noted in **11.1 till 11.10** in the database.

## Products page

**1. Navigation**

*What is it?*

A button which links to a different page and where the current page is highlighted.

*What is its purpose?*

Navigating a user to a different page in the backend of the application.

*How does it work?*

You press on the desired page you would like to visit with the left mouse button. After you click it the application will take you to this page.

**2. Search**

*What is it?*

An input / search field which lets you search for a machine either by typing its full name or part of the name.

*What is its purpose?*

Searching for a certain product / machine in the database.

*How does it work?*

By default all the products currently in the database will be listed. If you want to search for a machine press the search field and type in either the full name, or part of the name of the machine in the field. The listed machines will reload and look for the searched product. It’s possible to search for category, sub-category, product name or any of these combined.

**3. Add new product**

*What is it?*

A button on the right top side of the products page which you can click to add a new product.

*What is its purpose?*

Adding a new product (machine).

*How does it work?*

You press the ‘Add new product’ button with the left mouse button. Once you clicked it, it will open a new form in a panel on the right side of the page. In this input field you’ll find all the required fields to add a new product **(see 6)**.

**4. Show products and its details (machine)**

*What is it?*

By default all the products in the database will be listed on the products page. The listed products in the database can be clicked to show the details of the desired product.

*What is its purpose?*

Showing the details of a product in the database

*How does it work?*

Press on the name of a product in the listed items with the left mouse button to show its details and information in a panel on the right side of the screen.

**5. Edit**

*What is it?*

A button on the right side of the page (next to the product) which opens an edit form in a panel on the right side of the page.

*What is its purpose?*

Editing the desired product.

*How does it work?*

After you searched for a product **(see 2)**  and you found the desired product. After you searched for the right product, press the edit button next to it to open a form in a panel on the right side of the page which shows all the current information of the product **(see 6)**.

**6. form ‘Add new product’**

*Wat is het?*

A form with different input fields for adding a new product.

*What is its purpose?*

Adding a new product to the database

*How does it work?*

After you press the ‘Add new product’ button **(see 3)** a form will open in a panel on the right side of the page. In this form you can fill in all the information needed to add the new product.

**6.1 Category:** in this input field you select the category of the product by pressing the dropdown menu with the left mouse button. This will open a dropdown menu which will list all the categories currently in the database. You select one of the categories by pressing them with left mouse button. These categories can be added and edited in the ‘Categories’ page **(see 1)**. **This field is required!**

**6.2** **Subcategory:** in this input field you select the subcategory of the product by pressing the dropdown menu with the left mouse button. All the subcategories of the selected category in **6.1** will be listed in the dropdown. You can select a subcategory by opening the dropdown menu and clicking the desired subcategory with the left mouse button. These categories and subcategories can be added and edited in the ‘Categories’ page **(see 1)**. **This field is required!**

**6.3 (product)Name:** in this input field you can enter the name of the product you want to add. Take note that these product names should not contain the name of the category and / or the subcategory! Make sure the product doesn’t already exist in the database since its name has to be unique. **This field is required!**

**6.4 (product) photo:** you can add a photo to the new product by pressing the squared area in the top of the panel. You can select an image from the computer which you’d like to add to the product. This image will then be written to the database.

**6.5 Add engineers:** in this part you can add engineers who are trained for the new product, with their desired skill level.This field works as a search area where you can search for engineers which are currently in the database. You can type either part of the engineers name or its full name. After you select the desired engineer you can edit its skill level by pressing either the up or down arrow icon next to the name. After this you can add the engineer by pressing the green checkmark next to the engineers name. The engineer is now added to the new product. *(note for programming: the list of engineers that should be added to the product, has to be saved in an array first, because the product does not yet exist at that moment)*

**6.6 Save:** save the product with the information filled in in **6.1 till 6.5**. The product will be added to the database after you press the save button and can be found or edited from the products list.

**7. form ‘Edit product’**

*What is it?*

A form which lets you edit a products current information.

*What is its purpose?*

Editing a product in the database.

*How does it work?*

After you press the ‘Edit’ button with the left mouse button **(see 5)** the ‘Edit product’ form will open in a panel on the right side of the page. In this form you can fill in all the information needed to edit the existing product in the database.

**7.1 Category:** in this input field you select the category of the product by pressing the dropdown menu with the left mouse button. This will open a dropdown menu which will list all the categories currently in the database. You select one of the categories by pressing them with left mouse button. These categories can be added and edited in the ‘Categories’ page **(see 1)**. **This field is required!**

**7.2** **Subcategory:** in this input field you select the subcategory of the product by pressing the dropdown menu with the left mouse button. All the subcategories of the selected category in **6.1** will be listed in the dropdown. You can select a subcategory by opening the dropdown menu and clicking the desired subcategory with the left mouse button. These categories and subcategories can be added and edited in the ‘Categories’ page **(see 1)**. **This field is required!**

**7.3 (product)Name:** in this input field you can enter the name of the product you want to add. Take note that these product names should not contain the name of the category and / or the subcategory! Make sure the product doesn’t already exist in the database since its name has to be unique. **This field is required!**

**7.4 (product) photo:** you can add a new photo to the existing product by pressing the squared area in the top of the panel. You can select an image from the computer which you’d like to add to the product. This image will then be written to the database.

**7.5 Add engineers:** in this part you can add engineers who are trained for the new product, with their desired skill level.This field works as a search area where you can search for engineers which are currently in the database. You can type either part of the engineers name or its full name. After you select the desired engineer you can edit its skill level by pressing either the up or down arrow icon next to the name. After this you can add the engineer by pressing the green checkmark next to the engineers name. The engineer is now added to the new product.

**7.6 Edit / remove engineer:** below the ‘Add engineers’ field you can edit the skill levels of the engineers who are currently trained for the product. It’s also possible to remove an engineer by pressing the red cross next to the name of an engineer.

**7.7 Is active?:** a checkbox to check if the product is active or not. If the product is not active it won’t be found if you search for it.

**7.8 Save:** save the existing product with the new information filled in in **7.1 till 7.7**. The product will be replaced in the database with the old product after you press the save button and can be found or edited from the products list.

## Categories

**1. Navigation**

*What is it?*

A button which links to a different page and where the current page is highlighted.

*What is its purpose?*

Navigating a user to a different page in the backend of the application.

*How does it work?*

You press on the desired page you would like to visit with the left mouse button. After you click it the application will take you to this page.

**2. Search**

*What is it?*

An input field which lets you search for a category in the database.

*What is its purpose?*

Searching for a category in the database.

*How does it work?*

You press the input field / search bar with the left mouse button. After you press it you can fill in either the full name or part of the name of a category and it will automatically list the categories matching with the input. By default all the categories currently in the database will be listed.

**3. Add new category**

*What is it?*

A button which lets you add a new category.

*What is its purpose?*

Adding a new category to the database.

*How does it work?*

You press the ‘Add new category’ button in the top right with the left mouse button. After you press the button a new form will be opened in a panel on the right side of the screen **(see 6)**.

**4. Add/Remove subcategory**

*What is it?*

A button which lets you add a new subcategory to the corresponding category.

*What is its purpose?*

Adding or removing a subcategory on the selected category.

*How does it work?*

You press the ‘Add / Remove subcategory’ button next to the desired category with the left mouse button. After you press this a new form will open in a panel on the right side of the screen **(see 7)**.

**5. Current subcategories**

*Wat is het?*

A button which shows all the current subcategories of the category.

*What is its purpose?*

Showing the current subcategories of the desired category.

*How does it work?*

You press the name of one listed categories with the left mouse button. After this the category’s subcategories will be listed below its name in the listed categories.

**6. form ‘Add new category’**

*What is it?*

A form which lets you add a new category.

*What is its purpose?*

Adding a new category to the database.

*How does it work?*

After you press the ‘Add new category’ button **(see 3)** on the top right of the screen, this form will be opened in a panel on the right side of the screen.

**6.1 Category name:** the name of the category which you’d like to add.

**6.2 Add subcategory:** add a subcategory to the new category.

**6.3 Save:** save the information which you filled in in **6.1 and 6.2**.

**7. form ‘Add / remove subcategory’**

*What is it?*

A form which lets you edit a category which is currently in the database.

*What is its purpose?*

Editing the information of a category currently in the database or to add or remove sub categories.

*How does it work?*

After you press ‘Add / remove subcategory’ **(see 3)** a form will be opened in a panel on the right side of the screen.

**7.1 Category name:** the name of the category.

**7.2 Add subcategory:** add a new subcategory to the category by typing its name and pressing the green checkmark next to the input field.

**7.3 Remove subcategory:** Remove a subcategory by pressing the red cross next to the name of a subcategory.

**7.4 Save:** Save all the information filled in in **7.1 till 7.3** to the database.

## Users

**1. Navigation**

*What is it?*

A button which links to a different page and where the current page is highlighted.

*What is its purpose?*

Navigating a user to a different page in the backend of the application.

*How does it work?*

You press on the desired page you would like to visit with the left mouse button. After you click it the application will take you to this page.

**2. Search**

*What is it?*

An input field which lets you search for a user in the database.

*What is its purpose?*

Searching for a user in the database.

*How does it work?*

You press the input field / search bar with the left mouse button. After you press it you can fill in either the full name or part of the name of a user and it will automatically list the users matching with the input. By default all the user currently in the database will be listed.

**3. Add new User**

*What is it?*

A button which lets you add a new user to the database.

*What is its purpose?*

Adding a new user.

*How does it work?*

You press the ‘Add new user’ button in the top right of the screen with the left mouse button. After you press the button a form will open in a panel on the right side of the screen. **(see 5)**

**4. Edit user**

*What is it?*

A button next to the name of a user which lets you edit the desired user.

*What is its purpose?*

Editing a user.

*How does it work?*

You press the ‘Edit’ button next to the name of the listed user with the left mouse button. After this a form will open in a panel on the right side of the screen. In this form you can edit the information of the user. **(see 6)**

**5. form ‘Add new User’**

*What is it?*

A form with all the required information to add a new user.

*What is its purpose?*

Adding a new user to the database.

*How does it work?*

After you press the ‘Add new user’ button this form will open in a panel on the right side of the screen. **(see 3)**

* 1. **E-mail:** this is the e-mail address of the user. The user will use a combination of this e-mail address and the password to log in to the application.
  2. **Password:** this is the password of the new user. The user will use a combination of the e-mail address and this password to log in to the application.
  3. **Group:** In this field you have to select the new users permissions with a radio button. One of the options has to be selected. The options are:
     1. **Admin:** a user with admin permissions can edit / add Engineers, Products, Skills, Categories and Users.
     2. **Planner:** a user with these permissions can only edit engineers.
     3. **User:** a regular user has read only permissions on everything.
     4. **Save:** This will save the new user to the database with the information from **5.1 till 5.3**.

# Planning

## Sprint 1

### Week 1

**2-11-2015**

In the first week of our project, we planned on going to Agilent. We would go there to receive information about the project from Stephaan. They showed us what they already have, and what they wanted to improve on. After this, we started discussing ideas and we set up the mockups for the front- and the backend in Axure Pro. The designs would be presented at the next meeting, which took place 9-11-2015.

### Week 2

**9-11-2015**

After the meeting, where we discussed if the designs met their criteria, we discussed about the changes that needed to be made. In this week we made new and improved designs. We also discussed our ideas with Stephaan to make make sure we had the same idea that Agilent had for the Skill Matrix application. These designs would be shown to Stephaan at the next meeting, which took place 16-11-2015

### Week 3

**16-11-2015**

Per usual, we planned a meeting. Because we couldn’t show what we wanted, we only planned minor changes this week. We also planned a new meeting, which took place 20-11-2015.

After the second meeting that week, we finally got a green light for the front end designs. There were still some minor issues in the mockup, and the mockup of the backend was not completely done, but this meant that we could start making our templates.

## Sprint 2

Week 4 **23-11-2015**

Because we could begin working on the project, we planned that we would prepare the tools necessary to create the application. This includes: Github, Python, Django and more.

Since Python was a new programming language for most of us, we started learning working in Python on CodecAdemy. We also refined the mock-ups that we created and started working on our feature list.

Tom started working on setting up the database and the normalisation for this.

### Week 5

**30-11-2015**

We were still preparing before we would code. Erik and Davy were doing a Python code academy tutorial. And Jeffrey started a Django tutorial. We were also discussing a lot about what we should do and use. We still weren’t sure if we were going to use Python or PHP. The feature list was also still in progress this week.

### Week 6

**7-12-2015**

We were working on our HTML templates for the backend and frontend this week. Erik started working on the template for the backend and Davy and Jeffrey were working on the template for the frontend. Meanwhile Tom was working on the style definition for the application we were going to make.

We were also still working on our mockups. We had an agreement with Agilent over the overall graphic design of the application but were still working on creating some features in our mockups.

### Week 7

**14-12-2015**

This was our last week before the Christmas vacation and we had our last meeting about the mockups planned for the end of this week. In this meeting we wanted to get an agreement with Agilent so we could start working on the application.

Erik, Jeffrey and Davy worked on making the final version of the mockups, while they were also still learning to program in Python and work with Django.

Tom started programming on the API in this week. He also set up the project on GitHub so we all could start working on it. On the end of the day we got an agreement with Agilent after the meeting so the next sprint started. We could start working on our application.

## Sprint 3

Week 8 **04-01-2016**

Tom continued working on the API for the application. Erik, Jeffrey and Davy where still busy with documentation and Davy was still working on learning Python. Meanwhile Jeffrey continued learning more about Django and he started working on the login page of the application. Since Erik had no experience on working with Javascript / JQuery he also started learning this CodecAdemy course.

We also finished our HTML templates for the front- and backend with the required CSS files. We put Tom’s style definition of a few weeks ago in the custom CSS of the templates.

### Week 9

**11-01-2016**

While Davy was still busy with the python lessons, Tom started finishing up the API of the application. With this API we could get data from the database with JSON requests, so this meant we could start working on some functions of our application.

We started with getting our HTML templates in our Django application and started discussing about which features we should build first. We decided we could start with working on listing the items from the database in the backend, since we could get items from the API with JSON requests now.

We started working on these features the week after this.

### Week 10

**18-01-2016**

This week Erik started working on listing all the engineers on the engineers page in the backend. Jeffrey started working on the login feature for the application. Tom was making some changes to the API and was also working on creating a form plugin. This plugin will later be used to get the data from an engineer or product in our HTML template. Davy was still working on learning Python and Django.

### Week 11

**25-01-2016**

This week Erik started working on listing all the items on the product and the categories page. We also build the subcategory feature. We also decided we were going to use DataTables for listing these items. This plugin will make things a lot easier for us since it comes with a lot of nice features. Tom was still working on the API and started working on showing the details of an engineer once you clicked it in the listed items. Jeffrey was still working on the login feature of the application. Davy was working on a country filter for the front- and backend.

### Week 12

**01-02-2016**

Erik started working on the search function in the backend this week. Since this was a fairly easy task, he started working on features for the frontend. He started with listing the engineers and the products. Davy continued working on the search filter for countries. Tom was still making changes to the API and working on the form plugin so we could show and edit the details of an Engineer / Product / Category or User.

### Week 13

**08-02-2016**

This week we started working on the frontend of the application. Erik finished the listing of the engineers and the products in the frontend last week. Davy and Erik started working on getting Davy’s country filter to work on the frontend. Meanwhile Erik also started working on displaying the details of an engineer or a product on the frontend of the application.

Tom and Jeffrey started finishing the login feature of the application. They implemented it in the Django application so we finally had a login screen when starting up the application.

Meanwhile Tom was also working on showing the details of a database item and editing this item, and the first version of the form plugin was ready.

We had a meeting planned for Friday, but the meeting was cancelled, so we were showing Sander what we have.

### Week 14

**15-02-2016**

We discussed with Sander that the 26st of February would be the deadline of the project. This meant that we had two weeks left to finish the first version of the application. We also discussed that we would find a provider who could run our application. In the end we decided to go with WebFaction.

Tom finished the show details function in the backend and editing the database object in the backend was also working. He started working on the users and their permissions. Erik started working on adding a selected class to a table row and implementing a refresh button.

We finished up the engineer detail panel in the frontend and most big features of the application were done at this moment.

### Week 15

**22-02-2016**

Davy was going to find a way to copy a sentence in the engineer info panel by just clicking on a button. Erik decided to split up the search bar in the front end in an engineer and a product field. Erik also made the skill level of the engineer on the selected product show up in de details panel.

The biggest feature of the frontend worked at this moment. When you click an engineer you get to see which products the engineer is trained for and when you click a product you can see which engineers are trained for this. This week we also moved all static files.

Most of all we were fixing minor and major bugs to have a presentable version of the application by Friday. Even going as far as working till 12 o’clock in the evening. This because we would show Stephaan the “final” version.

## Sprint 4

### Week 16

**07-03-2016**

After the presentation and a vacation, we had some minor bugs that needed fixing.

For example: Changing the level display from a number to stars. Or fixing the filter button on the front-end. While working in the application, Agilent also found some bugs which we fixed this week.

Tom was also rounding up the note function, which Erik implemented in the front end. When an engineer has a note it now shows up in the engineer detail panel.

We also added FSS to the skill level field.

### Week 17

**14-03-2016**

This week we would be fixing all the bugs and be get the final version of the application online. The more we tested the application, the more bugs we found. We also added labels to the backend input fields and the right icons at all places.

Unfortunately, because there were too many bugs to fix in one week, we needed one more week. We got a solid working application working at the end of this week though. We also needed to complete some documentation.

### Week 18

**20-03-2016**

As said before, this was the last week. In this week we planned on fixing all the bugs and finishing the documentation.

We also had to finish some checkboxes which didn’t save the right value, and we made our code a little cleaner with some more comments. This is the last week and once our new hosting package is ready the final application will be uploaded.

|  |  |  |  |
| --- | --- | --- | --- |
| **When** | **What** | **Who** | **Hours** |
|  |  |  |  |
| **Sprint 1** | Discussions/Meetings | All | 60 |
|  | Mock-ups | All | 153 |
|  | Documentation | All | 90 |
|  |  |  |  |
| **Sprint 2** | Refining Mock-ups | Erik, Jeffrey | 68 |
|  | Discussions/Meetings | All | 86 |
|  | Studying Python | Erik, Davy, Jeffrey | 66 |
|  | Studying Django | All | 100 |
|  |  |  |  |
| **Sprint 3** | Programming API | Tom, Jeffrey | 64 |
|  | Html/Css Template Frontend | Davy, Jeffrey | 25 |
|  | Html/Css Template Backend | Erik | 55 |
|  | Frontend Functions | Davy, Erik, Jeffrey | 200 |
|  |  |  |  |
|  | Login | Jeffrey, Tom | 20 |
|  | Search Function | Erik | 40 |
|  | Refresh Search | Erik | 15 |
|  | Country Filter | Davy, Erik | 45 |
|  | Display Products/Engineers | Erik | 40 |
|  | Engineer Note | Erik | 25 |
|  | Engineer Level | Davy, Erik | 10 |
|  | Admin/Logout | Davy | 5 |
|  |  |  |  |
|  | Backend Functions | Erik, Davy, Tom | 226 |
|  |  |  |  |
|  | Display Items | Erik | 28 |
|  | List subcategories | Erik | 5 |
|  | Search Function | Erik | 6 |
|  | Country Filter | Davy ,Erik | 9 |
|  | API | Tom | 81 |
|  | Add Engineer | Tom | 25 |
|  | Edit Engineer | Tom | 25 |
|  | Edit Skills Engineer | Tom | 5 |
|  | Edit Note Engineer | Tom | 5 |
|  | Add Product | Tom | 5 |
|  | Edit Product | Tom | 5 |
|  | Add Category | Tom | 5 |
|  | Edit Category | Tom | 5 |
|  | Add User | Tom | 5 |
|  | Edit User | Tom | 5 |
|  | SkillMatrix/Logout | Davy | 5 |
|  |  |  |  |
|  | Discussions/Meetings | All | 36 |
|  | Documentation | Davy, Jeffrey, Erik | 24 |
|  |  |  |  |
| **Sprint 4** | Bugfixing | All | 208 |
|  | Discussions | All | 36 |
|  | Documentation | Davy, Jeffrey | 44 |