



Onderwijsgroep Professionele Opleidingen  
Handelswetenschappen en bedrijfskunde

# Opdracht 2 & 3

*Team 2Tl2.3 - OneHouseStand*

**Fabian Denoodt**  
**Glenn van Wezel**  
**Luis Sapon Lucas**  
**Bram Van den Abbeele**  
Tweede bachelor - Toegepaste Informatica

**Software Engineering 4**  
Hans Vandenbogaerde

---

Academiejaar 2018-2019

# 1 Inhoudsopgave

2	Opdracht 2 – Deel 1 .....	3
2.1	Een volledige CRUD in spring .....	3
2.2	Backlog refinement .....	11
2.2.1	User stories en tasks .....	11
2.2.2	Ordenen tasks .....	11
2.2.3	JIRA + Kanbanbord .....	12
2.3	Git repo .....	14
2.4	Sprint planning .....	16
2.5	Daily standups: .....	16
2.5.1	Sprint 1 .....	16
2.5.2	Sprint 2 .....	19
2.5.3	Sprint 3 .....	21
2.5.4	Sprint 4 .....	23
2.5.5	Sprint 5 .....	25
2.5.6	Sprint 6 .....	28
2.5.7	Sprint 7 .....	29
2.6	Prototype in Spring Boot – zwarte user stories .....	32
2.7	Validatie en Authenticatie .....	40
2.8	Rest, Rode en Groene user stories..... <b>Fout! Bladwijzer niet gedefinieerd.</b>	
2.8.1	[provide] een OneHouseStand-ruimte via Start-Up-circus te huur aanbieden .....	45
2.8.2	[consume] de matten in de verhuurde ruimtes kunnen bestellen en laten herstellen door Verimpex .....	45
2.8.3	[consume] Start-Up-circus-rumtes via OneHouseStand kunnen verhuren .....	46
2.9	Cucumber test.....	48
2.10	Microservices architectuur .....	50
2.11	Webfront en REST backend .....	51

## 2 Opdracht 2 – Deel 1

### 2.1 Een volledige CRUD in spring

In ons project hebben we CRUD verschillend keren toegepast bv:

Een pand kunnen:

- Toevoegen,
- Verwijderen
- Updaten
- Bekijken

Bekijken:

```
<h1>Lijst van panden</h1>
<a th:href="@{/klant/pandToevoegen}">Nieuw pand</a>
<ul style="list-style-type: none;">
  <li th:each="pand : ${lijstPanden}">
    <a th:href="@{/pand(id=${pand.getId()})}" contenteditable="false">
      <form>
        <span th:text="${pand.getId()}" contenteditable="false">Loading...</span> |
        <span th:text="${pand.getStatus()}" contenteditable="false">Loading...</span> |
        <span th:text="${pand.getPrijsPerUur()}" contenteditable="false">Loading...</span> Euro/uur
      </form>
    </a>
  </li>
</ul>
</ul>
```

Op de aanbod pagina worden alle panden weergegeven. Hier wordt enkele info gegeven van het pand, en wanneer er een pand geselecteerd wordt, navigeert men naar de details van dat pand.

Hier kan je de backend zien van deze pagina. Doormiddel van th:each wordt elk element weergegeven, en kan wordt er naar

de detail pagina gegaan door de id van het pand in de url mee te geven.

```
th:href="@{/pand(id=${pand.getId()})}"
```

via `@GetMapping("/aanbod")`, zal wanneer er naar `localhost:xxx/aanbod` genavigeerd wordt, alle panden weergegeven worden en de html pagina "aanbod.html" weergegeven worden.

```
@Controller
public class HomeController {

    @Autowired
    private OhsService ohsService;

    @GetMapping("/aanbod")
    public String aanbod(Model model) {
        Iterable<Pand> panden = ohsService.geefAlleGoedgekeurdePanden();
        model.addAttribute("lijstPanden", panden);
        return "aanbod";
    }
}
```

```
@Controller
public class HomeController {

    @Autowired
    private OhsService ohsService;

    @GetMapping("/pand")
    public String pandDetailForm(Model model, @RequestParam("id") long id) {
        Pand pand = ohsService.geefPand((int) id);

        model.addAttribute("pand", pand);
        return "pand";
    }
}
```

Wanneer een pand geselecteerd wordt, kan je de details ervan bekijken in de aanbod pagina

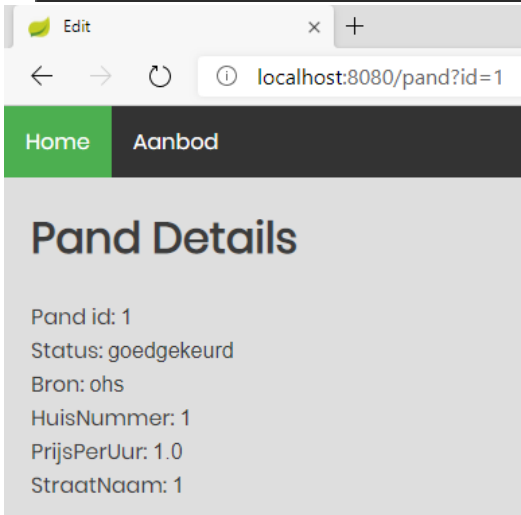
```
@Controller
public class HomeController {

    @Autowired
    private OhsService ohsService;

    @GetMapping("/aanbod")
    public String aanbod(Model model) {
        Iterable<Pand> panden = ohsService.geefAlleGoedgekeurdePanden();
        model.addAttribute("lijstPanden", panden);
        return "aanbod";
    }
}
```

Een pand kan worden opgehaald uit de databank adhv de id.

@RequestParam haalt de id op.

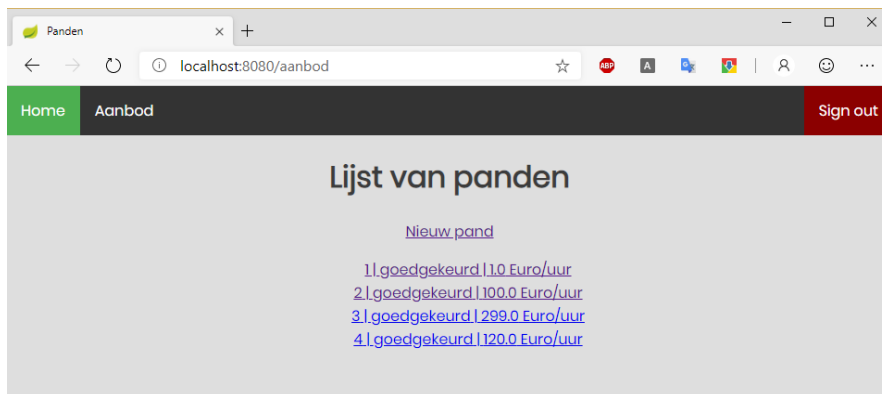


```
<div class="container">
<form method="post" th:action="@{/klant/pand(id=${pand.id})}" th:object=${pand}>
  <label for="id">Pand id:</label>
  <input id="id" class="readonly" type="text" th:field="*{id}" readonly="readonly"/>
  <br>
  <label for="status">Status:</label>
  <input id="status" class="readonly" type="text" th:field="*{status}" readonly="readonly"/>
  <br>
  <label for="bron">Bron:</label>
  <input id="bron" class="readonly" type="text" th:field="*{bron}" readonly="readonly"/>
  <br>
  <label for="huisnummer">HuisNummer:</label>
  <input id="huisnummer" readonly="readonly" class="readonly" type="text" th:field="*{huisnummer}">
  <br>
  <label for="prijsperuur">PrijsPerUur:</label>
  <input id="prijsperuur" readonly="readonly" class="readonly" type="text" th:field="*{prijsPerUur}">
  <br>
  <label for="straatnaam">StraatNaam:</label>
  <input id="straatnaam" readonly="readonly" class="readonly" type="text" th:field="*{straatNaam}">
  <br>
</div>
```

Pagina van ons pand pagina waar we de details van de gekozen pand getoond wordt

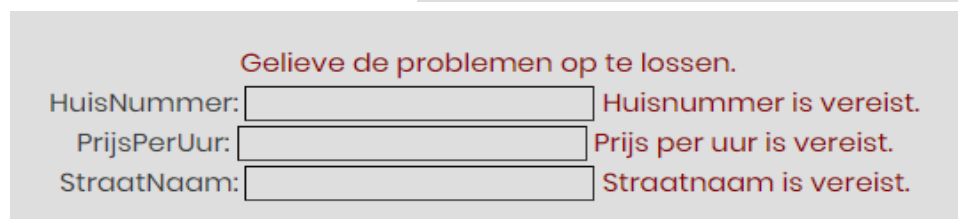
## Toevoegen

Als gebruiker kunnen we in ons aanbod pagina een nieuw pand toevoegen



Pagina van pandToevoegen:

Een pand kan toegevoegd worden, dit geeft een successboodschap wanneer alle gegevens ingevuld zijn en correct zijn. Zo niet wordt er een foutmelding weergegeven van het veld dat niet goed is.



Dit werd in ons domain pand ingesteld

Velden van klasse "Pand" die gevalideerd worden:

```
@NotBlank(message = "Huisnummer is vereist.")
private String huisnummer;

@Range(min = 0L, message = "Prijs per uur moet een positief getal zijn.")
@NotNull(message = "Prijs per uur is vereist.")
private Double prijsPerUur;

private String status;

@NotBlank(message = "Straatnaam is vereist.")
private String straatNaam;
```

```
@GetMapping("/pandToevoegen")
public String toevoegenPandForm(Model model) {
    model.addAttribute("pand", new Pand());
    return "pandToevoegen";
}
```

Met GetMapping komen we tot ons pandToevoegen pagina en geven we een pand object door

Hier kan je zien hoe we met een html "span" de error ophalen en die weergeven wanneer nodig.

```
<form method="post" th:action="@{/klant/pandToevoegen}" th:object=${pand}>
  <div th:if=${#fields.hasErrors()}>
    <span class="validationError">Gelieve de problemen op te lossen.</span>
  </div>

  <label for="huisnummer">Huisnummer:</label>
  <input id="huisnummer" type="text" th:field=${#fields['huisnummer']}>
  <span class="validationError" th:if=${#fields.hasErrors('huisnummer')} th:errors=${#fields['huisnummer'].errors}>
  </span>

  <br>

  <label for="prijsPerUur">PrijsPerUur:</label>
  <input id="prijsPerUur" type="number" th:field=${#fields['prijsPerUur']}>
  <span class="validationError" th:if=${#fields.hasErrors('prijsPerUur')} th:errors=${#fields['prijsPerUur'].errors}>
  </span>

  <br>

  <label for="straatNaam">StraatNaam:</label>
  <input id="straatNaam" type="text" th:field=${#fields['straatNaam']}>
  <span class="validationError" th:if=${#fields.hasErrors('straatNaam')} th:errors=${#fields['straatNaam'].errors}>
  </span>

  <br>
  <input type="submit" class="submit" name="submit" value="Toevoegen">
</form>
```

## Pand toevoegen

We hebben je aanvraag ontvangen, gelieve enkele dagen te wachten zodat wij je pand kunnen verwerken.

HuisNummer:   
PrijsPerUur:   
StraatNaam:

Toevoegen

Wanneer op “toevoegen” geklikt wordt, zal de controller reageren op een post van “pandToevoegen”, hier worden de velden gevalideerd en wordt een nieuw pand eventueel toegevoegd aan de DB. Hierboven zie je het resultaat van het nieuwe pand dat toegevoegd is.

```
@PostMapping(path = "/pandToevoegen")
public String processToevoegenPand(@Valid Pand pand, Errors errors, Model model) {
    if (errors.hasErrors())
        return "pandToevoegen";

    User user = userDetailsService.getUser();
    ohsService.addPand(pand, user);

    model.addAttribute("success", true);
    model.addAttribute("pand", new Pand());
    return "pandToevoegen";
}
```

wanneer (success == true), wordt er op het startscherm een success bericht weergegeven van het toevoegen van pand.

Home Aanbod

### Lijst van panden

1 | goedgekeurd | 1.0 Euro/uur [Update](#)  
2 | goedgekeurd | 100.0 Euro/uur [Update](#)  
3 | goedgekeurd | 299.0 Euro/uur [Update](#)  
4 | goedgekeurd | 120.0 Euro/uur [Update](#)  
5 | ongekeurd | 1.0 Euro/uur [Update](#)

Het pand is toegevoegd, en zichtbaar in de lijst.

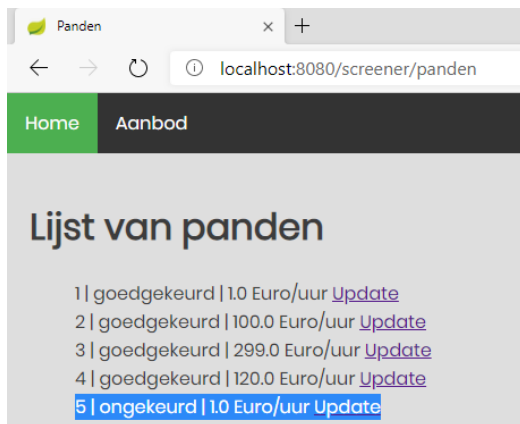
## Cucumber test:

Hier wordt gecheckt of het toevoegen van een pand werkt, en dat dat nieuwe pand zichtbaar is in de lijst. Opmerking: deze is verouderd, en hier wordt er wat verder in de paper meer in detail op gegaan.

```
1  Feature: Add House
2
3  It should be possible to add a house to the list
4  Just after adding the house,
5  the list is shown and contains the newly added house
6
7  Scenario: Add one house
8    Given I am on the page where I can add a new house
9    When I enter "800" in the prijsPerUur field
10   And I enter "Boulevardlaan" in the straatNaam field
11   And I enter "60" in the huisNummer field
12   And I press on the Submit button
13   Then I should see a list containing "ongekeurd", "800"
14   And I close the browser
```

## Een screener kan een pand deleten

Als we op update drukken kunnen komen we op ons edit pagina terecht



Op deze pagina kunnen we de juiste pand met zijn details

Home Aanbod

## Edit pand

Pand id: 1  
 Status: goedgekeurd  
 Bron: ohs  
 HuisNummer:   
 PrijsPerUur:   
 StraatNaam:

```

<!--Delete-->
<form method="post" th:action="@{/screener/deletePand(id=${id})}" th:object="${pand}">
  <input
    type="submit"
    name="Verwijder pand"
    class="submit"
    value="Verwijder pand"/>
</form>

<!--Delete-->
<form method="post" th:action="@{/screener/deletePand(id=${id})}" th:object="${pand}">
  <input
    type="submit"
    name="Verwijder pand"
    class="submit"
    value="Verwijder pand"/>
</form>

```

Op deze pagina

kunnen we de gekozen pand verwijderen, als we op verwijderen pand drukken dan wordt de @PostMapping /deletepand opgeroepen en daarna komen we op de panden pagina terecht.

```

@PostMapping("/deletePand")
public String deletePand(@RequestParam("id") long id, Model model) {
    ohsService.deletePand(id);
    Iterable<Pand> panden = ohsService.geefAllePanden();
    model.addAttribute("lijstPanden", panden);
    return "panden";
}

```



```
<h1>Edit pand</h1>

<div class="container">
  <form method="post" th:action="@{/screener/editPand(id=${id})}" th:object="${pand}">
    <div th:if="${#fields.hasErrors()}>
      <span class="validationError">Gelieve de problemen op te lossen.</span>
    </div>
    <label for="pandId">Pand id:</label>
    <input id="pandId" class="readonly" type="text" th:field="**{id}" readonly="readonly"/>
    <br>
    <label for="status">Status:</label>
    <input id="status" class="readonly" type="text" th:field="**{status}" readonly="readonly"/>
    <br>
    <label for="bron">Bron:</label>
    <input id="bron" class="readonly" type="text" th:field="**{bron}" readonly="readonly"/>
    <br>
    <label for="huisNumber">HuisNumber:</label>
    <input id="huisNumber" type="text" th:field="**{huisNumber}">
    <span class="validationError" th:if="${#fields.hasErrors('huisNumber')}" th:errors="**{huisNumber}">huisnummer err
    <br>
    <label for="prijsPerUur">PrijsPerUur:</label>
    <input id="prijsPerUur" type="number" th:field="**{prijsPerUur}">
    <span class="validationError" th:if="${#fields.hasErrors('prijsPerUur')}" th:errors="**{prijsPerUur}">prijsPerUur
    <br>
    <label for="straatNaam">StraatNaam:</label>
    <input id="straatNaam" type="text" th:field="**{straatNaam}">
  </form>
</div>
```

Ons editPand pagina

**Update:**

En op deze pagina kunnen we ook ons pand updaten. Op ons editPand krijgen we de pand die we wouden, pand id, status en bron kunnen niet gewijzigd worden alleen huisnummer, prijsperuur en straatnaam.

```

<!--Edit-->
<form method="post" th:action="@{/screener/editPand(id=${id})}" th:object="${pand}">
  <div th:if="${#fields.hasErrors()}>
    <span class="validationError">Gelieve de problemen op te lossen.</span>
  </div>

  <label for="pandId">Pand id:</label>
  <input id="pandId" class="readOnly" type="text" th:field="*{id}" readOnly="readOnly"/>

  <br>

  <label for="status">Status:</label>
  <input id="status" class="readOnly" type="text" th:field="*{status}" readOnly="readOnly"/>

  <br>

  <label for="bron">Bron:</label>
  <input id="bron" class="readOnly" type="text" th:field="*{bron}" readOnly="readOnly"/>

  <br>

  <label for="huisNummer">HuisNummer:</label>
  <input id="huisNummer" type="text" th:field="*{huisNummer}">
  <span class="validationError" th:if="${#fields.hasErrors('huisNummer')}" th:errors="*{huisNummer}">huisnummer error</span>

  <br>

  <label for="prijsPerUur">PrijsPerUur:</label>
  <input id="prijsPerUur" type="number" th:field="*{prijsPerUur}">
  <span class="validationError" th:if="${#fields.hasErrors('prijsPerUur')}" th:errors="*{prijsPerUur}">prijsPerUur error</span>

  <br>

  <label for="straatNaam">StraatNaam:</label>
  <input id="straatNaam" type="text" th:field="*{straatNaam}">
  <span class="validationError" th:if="${#fields.hasErrors('straatNaam')}" th:errors="*{straatNaam}">straatNaam error</span>

  <br>

  <!--Update button-->
  <input type="submit" class="submit" name="submit" value="Update">
</form>

```

Als we op update drukken komen we op de editPand terecht, dankzij onze @GetMapping /editPand verkrijgen we de juiste pand.id

```

@Controller
@PostAuthorize("#{model.get('role').user.username == authentication.principal.username}")
@Secured({"SCREENER"})
public class ScreenerController {

    @Autowired
    private OhsService ohsService;

    @GetMapping("/{id}/editPand")
    public String editPandForm(Model model, @RequestParam("id") long id) {

        Pand pand = ohsService.geefPand((int) id);
        Iterable<Mat> matten = pand.getMatten();
        Iterable<Review> reviews = pand.getReviews();

        model.addAttribute("id", pand.getId());
        model.addAttribute("pand", pand);
        model.addAttribute("mat", new Mat());
        model.addAttribute("lijstMatten", matten);
        model.addAttribute("lijstReviews", reviews);

        return "editPand";
    }
}

```

```

@GetMapping("/{id}/editPand")
public String editPandForm(Model model, @RequestParam("id") long id) {

    Pand pand = ohsService.geefPand((int) id);
    Iterable<Mat> matten = pand.getMatten();

    model.addAttribute("id", pand.getId());
    model.addAttribute("pand", pand);
    model.addAttribute("mat", new Mat());
    model.addAttribute("lijstMatten", matten);

    return "editPand";
}

@PostMapping(path = "/editPand")
public String processWijzigenPand(@Valid Pand pand, Errors errors, Model model, @RequestParam("id") long id) {

    if (errors.hasErrors()) {
        model.addAttribute("id", id);
        return "editPand";
    }

    ohsService.editPand(pand, id);
    Iterable<Pand> panden = ohsService.geefAllePanden();
    model.addAttribute("lijstPanden", panden);
    return "panden";
}

```

Uiteindelijk als we op update drukken wordt ons @PostMapping /editPand en daarna worden we teruggebracht naar ons lijst van panden waar ons pand is geupdate.

## 2.2 Backlog refinement

Tijdens de backlog redefiniment gaan we de userstories beter uitwerken zodat we deze tot in detail begrijpen. Dit is nodig om later een schatting te kunnen maken van hoeveel tijd een userstorie in beslag zal nemen om deze uit te werken, en hoe moeilijk deze is. Dit doen we door de gekregen userstories aan het begin van het semester uit te werken in tasks.

### 2.2.1 User stories en tasks

- Als verhuurder kan ik een pand voor verhuur aan OneHouseStand voorstellen
  - Pand toevoegen
- Als screener kan ik een pand aan OneHouseStand toevoegen
  - Pand keuren
- Als syndicus kan ik een pand van de nodige Verimpex-matten voorzien
  - Toevoegen van een mat aan een pand
- Als huurder kan ik een geschikt pand voor mijn evenement zoeken en vastleggen
  - Maken van een reservatie op een pand
- Als syndicus kan ik een probleem met de matten (laten) doorgeven aan Verimpex
  - Syndicus kan een mat herstellen
  - Syndicus kan een mat markeren als versleten
  - Een lijst van matten is zichtbaar bij de details van pand
- Als huurder kan ik na huur het gehuurde pand reviewen
  - Huurder kan na huur pand reviewen
- Als screener kan ik de informatie over het pand aanpassen op basis van de reviews door de huurder
  - Screener kan informatie aanpassen over het pand op basis van reviews
- [provide] een OneHouseStand-ruimte via Start-Up-circus te huur aanbieden
  - Mock api maken voor start-up circus
  - OneHouseStand-ruimte via Start-Up circus te huur aanbieden
- [consume] de matten in de verhuurde ruimtes kunnen bestellen en laten herstellen door Verimpex
  - Mock api maken voor Verimpex
  - De matten in de verhuurde ruimtes kunnen bestellen en laten herstellen door verimpex
- [consume] Start-Up-circus-ruimtes via OneHouseStand kunnen verhuren
  - Start-up circus ruimtes kunnen verhuren via OHS

### 2.2.2 Ordenen tasks

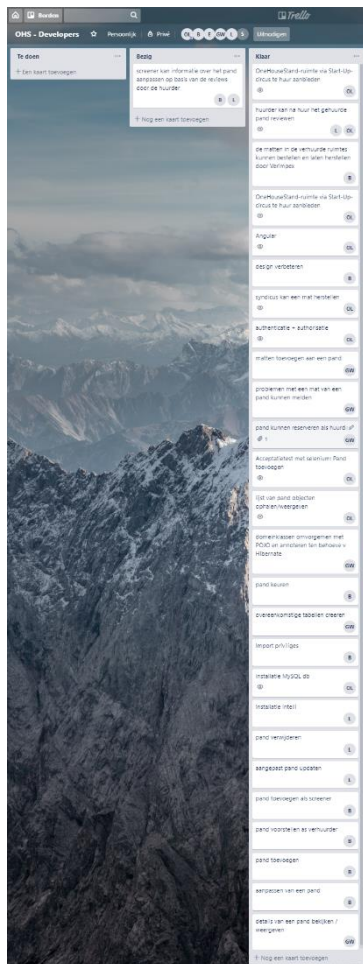
Hier gaan we de taken die we hiernet gemaakt hebben ordenen **op basis van moeilijkheidsgraad**. De moeilijkste taken komen eerst, de makkelijkste komt laatst.

1. De matten in de verhuurde ruimtes kunnen bestellen en laten herstellen door verimpex
2. Start-up circus ruimtes kunnen verhuren via OHS

3. OneHouseStand-ruimte via Start-Up circus te huur aanbieden
4. Screener kan informatie aanpassen over het pand op basis van reviews
5. Mock api maken voor start-up circus
6. Mock api maken voor Verimpex
7. Pand toevoegen
8. Toevoegen van een mat aan een pand
9. Maken van een reservatie op een pand
10. Huurder kan na huur pand reviewen
11. Syndicus kan een mat herstellen
12. Een lijst van matten is zichtbaar bij de details van pand
13. Pand keuren
14. Syndicus kan een mat markeren als versleten

## 2.2.3 JIRA + Kanbanbord

Kanbanbord:



▼ **OHS Sprint 1** 5 issues






0 0 25 ...

31/mrt/19 5:21 PM • 13/apr/19 5:21 PM

 import en privileges	OHS-1	↑	3
 installatie intellij	OHS-2	↓	2
 installatie MySQL db	OHS-3	↓	2
 domeinklassen omvormen tot POJO en annoteren ten behoeve	OHS-4	↑	11
 details van een pand bekijken	OHS-5	↑	7




**OHS Sprint 2** 5 issues

Sprint starten ...

 overeenkomstige tabellen creëren	OHS-6	↑	7
 lijst van pand objecten ophalen	OHS-7	↑	7
 pand toevoegen	OHS-8	↑	4
 pand keuren	OHS-9	↑	5
 aangepast pand updaten	OHS-10	↓	3

**OHS Sprint 3** 3 issues

...

 acceptatie testen schrijven met selenium + testen : panden	OHS-11	↑	8
 paper bijebrengen	OHS-12	↓	1
 design	OHS-13	↑	3

**OHS Sprint 4** 5 issues

...

 authenticatie	OHS-14	↑	4
 Pand toevoegen (nieuwe pagina)	OHS-15	↑	2
 huurder kan na huur pand reviewen	OHS-16	↑	4
 lijst van goedgekeurde panden voor huurder	OHS-17	↑	2
 maken van een reservatie op een pand	OHS-18	↑	4



### OHS Sprint 5 5 issues

 toevoegen van een mat aan een pand	OHS-19	↑	4
 syndicus kan een mat herstellen	OHS-20	↑	3
 syndicus kan een mat markeren als versleten	OHS-22	↑	2
 screener kan informatie aanpassen over het pand op basis v	OHS-21	↑	4
 een lijst van matten is zichtbaar bij de details van een pand	OHS-23	↑	3

### OHS Sprint 6 6 issues

 Mock api maken voor Verimpex	OHS-24	↑	5
 Mock api maken voor Start-Up-Circus	OHS-25	↑	5
 OneHouseStand-ruimte via Start-Up-circus te huur aanbiede	OHS-26	↑	5
 OHS api maken	OHS-27	↑	5
 start-up-circus ruitmes kunnen reserveren via OHS	OHS-28	↑	7
 de matten in de verhuurde ruimtes kunnen bestellen en late	OHS-29	↑	9

### OHS Sprint 7 2 issues

 frontend in angular	OHS-30	↑	7
 Microservices	OHS-31	↑	6

## 2.3 Git repo

GlennvanWezel / OneHouseStandSE4 Private Watch 0 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#)

*No description, website, or topics provided.*

101 commits 5 branches 0 releases 3 contributors

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find File](#) [Clone or download](#)

VandenAbbeeleBram update lijst van reviews Latest commit 77a95ac 25 minutes ago

src	update lijst van reviews	25 minutes ago
.gitignore	initial commit	a month ago
README.md	Initial commit	19 days ago
pom.xml	Mat toevoegen aan een pand werkt nu als men zich inlogt als syndicus...	4 days ago

README.md

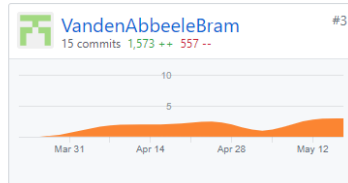
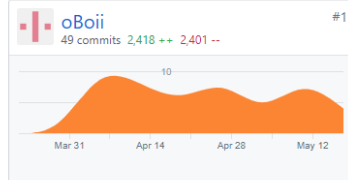
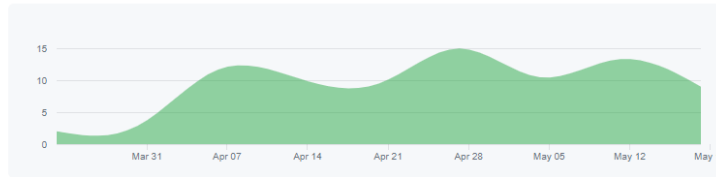
# OneHouseStandSE4

- Pulse
- Contributors**
- Traffic
- Commits
- Code frequency
- Dependency graph
- Network
- Forks

Mar 24, 2019 – May 19, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits



- Overview**
- Yours
- Active
- Stale
- All branches

Search branches...

**Default branch**

master	Updated 24 minutes ago by VandenAbbeeleBram	Default
--------	---	---------

**Your branches**

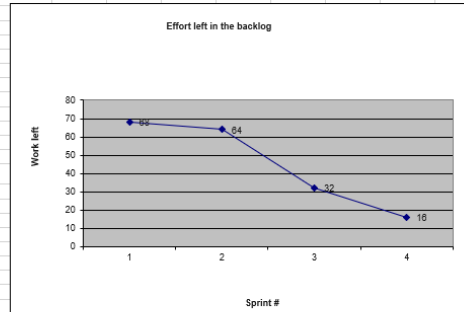
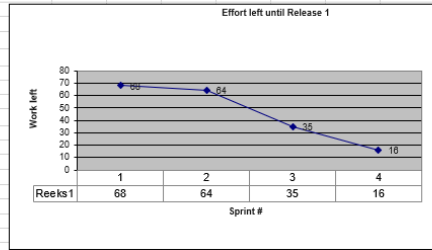
fabian	Updated 3 days ago by oBoii	25   2	New pull request	🗑
Glenn	Updated 3 hours ago by GlennvanWezel	5   0	New pull request	🗑

**Active branches**

Glenn	Updated 3 hours ago by GlennvanWezel	5   0	New pull request	🗑
Luis	Updated 18 hours ago by SaponLucasLuis	14   4	New pull request	🗑
Bram	Updated a day ago by VandenAbbeeleBram	40   1	New pull request	🗑
fabian	Updated 3 days ago by oBoii	25   2	New pull request	🗑

## 2.4 Sprint planning

OneHouseStand										
ID		Sprint #	1	2	3	4	5	6	7	8
4	import en privileges		3	0	0	0	0	0	0	0
1	installatie IntelliJ		2	0	0	0	0	0	0	0
3	installatie MySQL db		2	0	0	0	0	0	0	0
7	domeinklassen omvormen tot POJO en annoteren ten behoeve van Hibernate		11	11	0	0	0	0	0	0
10	details van een pand bekijken		7	7	7	7	0	0	0	0
<b>sprint 1</b>										
8	overeenkomstige tabellen creëren		7	7	0	0	0	0	0	0
9	lijst van pand objecten ophalen		7	7	0	0	0	0	0	0
11	pand toevoegen		4	4	0	0	0	0	0	0
12	pand keuren		5	5	5	5	0	0	0	0
7	aangepast pand updaten		3	3	3	0	0	0	0	0
<b>sprint 2</b>										
10	acceptatie testen schrijven met selenium + testen : panden weergeven		8	8	8	0	0	0	0	0
11	bijeenbrengen paper + review + indienen		1	1	1	0	0	0	0	0
13	design		3	3	3	0	0	0	0	0
<b>sprint 3</b>										
14	authenticatie		4	4	4	4	0	0	0	0
15	Pand toevoegen (nieuwe pagina)		2	2	2	2	0	0	0	0
16	huurder kan na huur pand reviewen		4	4	4	4	4	4	4	4
17	lijst van goedgekeurde panden voor huurder		2	2	2	2	0	0	0	0
18	maken van een reservatie op een pand		4	4	4	4	0	0	0	0
<b>sprint 4</b>										
19	toevoegen van een mat aan een pand		4	4	4	4	0	0	0	0
20	syndicus kan een mat herstellen		3	3	3	3	0	0	0	0
21	screener kan informatie aanpassen over het pand op basis van reviews		4	4	4	4	4	4	4	4
22	syndicus kan een mat markeren als versleten		2	2	2	2	0	0	0	0
23	een lijst van matten is zichtbaar bij de details van een pand		3	3	3	3	0	0	0	0
<b>sprint 5</b>										
24	Mock api maken voor Verimpex		5	5	5	5	5	0	0	0
25	Mock api maken voor Start-Up-Circus		5	5	5	5	5	0	0	0
26	OneHouseStand-ruimte via Start-Up-circus te huur aanbieden		5	5	5	5	5	0	0	0
27	OHS api maken		5	5	5	5	5	0	0	0
28	start-up-circus ruimtes kunnen reserveren via OHS		7	7	7	7	7	0	0	0
29	de matten in de verhuurde ruimtes kunnen bestellen en laten herstellen door Verimpex		9	9	9	9	9	0	0	0
<b>sprint 6</b>										
30	frontend in angular		7	7	7	7	7	7	0	0
31	Microservices		6	6	6	6	6	6	0	0
<b>sprint 7</b>										
			141	137	108	98	74	57	43	0



## 2.5 Daily standups:

### 2.5.1 Sprint 1

#### sprint 1 – dag 1 (zo 31/3)

- Fabian:
  - o Gedaan:
    - Installatie mysql db
    - Lokaal: project v leerkracht werkende gekregen + personen/rollen verwijderen
  - o Wat ga ik doen:
    - Persoon updaten binnen project
  - o Verwacht ik problemen?
    - Nee
- Bram:
  - o Gedaan:
    - Installatie mysql db
    - Aanmaken van database
    - Importeren van sql script in database
  - o Wat ga ik doen:
    - Gebruikersaccounts toevoegen en permissies geven
  - o Verwacht ik problemen?
    - Nee
- Glenn:
  - o Gedaan:
    - Installatie IntelliJ Ultimate
    - Importeren van project code
    - Schematisch nadenken over hoe POJO's schrijven



- Wat ga ik doen:
  - POJO's schrijven voor elke domein klasse
- Verwacht ik problemen?
  - Nein
- Luis:
  - Gedaan:
    - Installatie IntelliJ Ultimate
  - Wat ga ik doen:
    - Helpen waar ik kan
  - Verwacht ik problemen?
    - Nee
- Emmanuel :
  - Gedaan:
    - CRUD screencast nog eens doornemen
  - Wat ga ik doen:
    - Pand controller – view klassen schrijven
  - Verwacht ik problemen?
    - Nee

#### **sprint 1 – dag 2 (wo 3/4)**

- Fabian:
  - Gedaan:
    - Lokaal: project v leerkracht persoon updaten
  - Wat ga ik doen:
    - Springboot project v leerkracht runnende gekregen
  - Verwacht ik problemen?
    - Nee
- Bram:
  - Gedaan:
    - Gebruikersaccounts toevoegen en rechten geven
  - Wat ga ik doen:
    - Niets
  - Verwacht ik problemen?
    - Nee
- Luis:
  - Gedaan:
    - Klaar
  - Wat ga ik doen:
    - Helpen waar ik kan
  - Verwacht ik problemen?
    - Nee

#### **sprint 1 – dag 3 (za 6/4)**

- Fabian:
  - Gedaan:
    - Springboot project v leerkracht runnende gekregen + ontleed
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?

- Nee
- Bram:
  - Gedaan:
    - Niets
  - Wat ga ik doen:
    - Niets
  - Verwacht ik problemen?
    - Nee
- Luis:
  - Gedaan:
    - Niets
  - Wat ga ik doen:
    - Niets
  - Verwacht ik problemen?
    - Nee

### Einde sprint 1

ID		Sprint #	1	2	3
			71	67	67
4	import en privileges		3	0	0
1	installatie intellij		2	0	0
3	installatie MySQL db		2	0	0
7	domeinklassen omvormen tot POJO en annoteren ten behoeve van Hibernate		11	11	11
10	details van een pand bekijken		7	7	7
<b>sprint 1</b>					
8	overeenkomstige tabellen creëren		7	7	7
9	lijst van pand objecten ophalen		7	7	7
6	succesvolle test van voorbeelden		5	5	5
10	acceptatie testen schrijven met selenium + testen		8	8	8
12	pand keuren		5	5	5
<b>sprint 2</b>					
6	pand gegevens aanpassen in frontend		5	5	5
11	pand toevoegen		4	4	4
7	aangepast pand updaten in db		3	3	3
8	pand verwijderen		3	3	3
11	bijeenbrengen paper + review + indienen		1	1	1
	presentatie en demo voorbereiden		1	1	1
<b>sprint 3</b>					
	<b>Verkoopbare versie</b>				
<b>Effort in the whole backlog</b>			71	67	67

### Sprint retrospective, sprint 1 (do 11/4):

**Glenn:** sprint is slecht verlopen omdat deze in de projectweek viel.

**Luis:** het was te verwachten dat het niet zo goed ging gaan, door de projectweek onder ander.

**Fabian:** beter planning, wanneer er mensen afhankelijk zijn van je werk, gelieve niet uit te stellen tot de laatste dag. Er ontstaan dan nog verwickelingen waardoor er iets niet werkt en het niet af geraakt tegen sprint 1.

**Bram:** positief, iedereen was aanwezig daily stand ups.

Conclusies voor volgende sprint is om op tijd te beginnen en op tijd hulp vragen wanneer nodig.

## 2.5.2 Sprint 2

### sprint 2 – dag 1 (za 13/4)

- Fabian:
  - o Gedaan:
    - Springboot toepassing gelinkt met xamp mysql db
    - Relatie annotaties toegevoegd aan domain klassen
    - Lijst van panden weergegeven
  - o Wat ga ik doen:
    - Opmaak van lijst verbeteren
  - o Verwacht ik problemen?
    - Nee
- Bram:
  - o Gedaan:
    - Toevoegen van een pand aan database via form
    - Validatie van form input + error weergave
    - Opmaak formulier
  - o Wat ga ik doen:
    - Algemene opmaak applicatie verbeteren
  - o Verwacht ik problemen?
    - Nee
- Glenn
- Gedaan:
- - structuur van bestaand project omvormen naar vereiste structuur (controllers, DAO, repositories, services)
- Wat ga ik doen:
- - mijn taak is af
- Verwacht ik problemen?
- - Nee

### sprint 2 – dag 2 (ma 15/4)

- Fabian:
  - o Gedaan:
    - Opmaak van overzicht panden verbeterd
  - o Wat ga ik doen:
    - /
  - o Verwacht ik problemen?
    - Nee
- Bram:
  - o Gedaan:
    - Algemene opmaak applicatie verbeteren
  - o Wat ga ik doen:
    - Algemene opmaak applicatie verbeteren
  - o Verwacht ik problemen?
    - Nee

### sprint 2 – dag 3 (di 16/4)

- Fabian:
  - Gedaan:
    - Al klaar
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?
    - Nee
- Bram:
  - Gedaan:
    - Algemene opmaak applicatie verbeteren
  - Wat ga ik doen:
    - Niets, einde sprint
  - Verwacht ik problemen?
    - Nee
- Glenn
  - Gedaan:
    - required annotaties toevoegen
  - wat ga ik doen:
    - niets
  - verwacht ik problemen?
    - nee
- Emmanuel :
  - Gedaan:
    - Details van een pand weergeven controller klasse implementeren
  - Wat ga ik doen:
    - Code debuggen en testen
  - Verwacht ik problemen?
    - Nee
- Luis:
  - Gedaan:
    - Update en delete van crud element opzoeken
  - Wat ga ik doen:
    - dat proberen te implementeren in ons project
  - Verwacht ik problemen?
    - Ja, ik krijg problemen om het te booten

### **Sprint retrospective:**

**Fabian:** Studenten die vorige keer dingen uitstellen hebben zich herpakt en hebben deze sprint hun werkt onmiddellijk uitgevoerd. 2 andere studenten zijn niet naar de daily standups gekomen en konden dus geen hulp vragen waardoor ze hun werk niet hebben uitgevoerd.

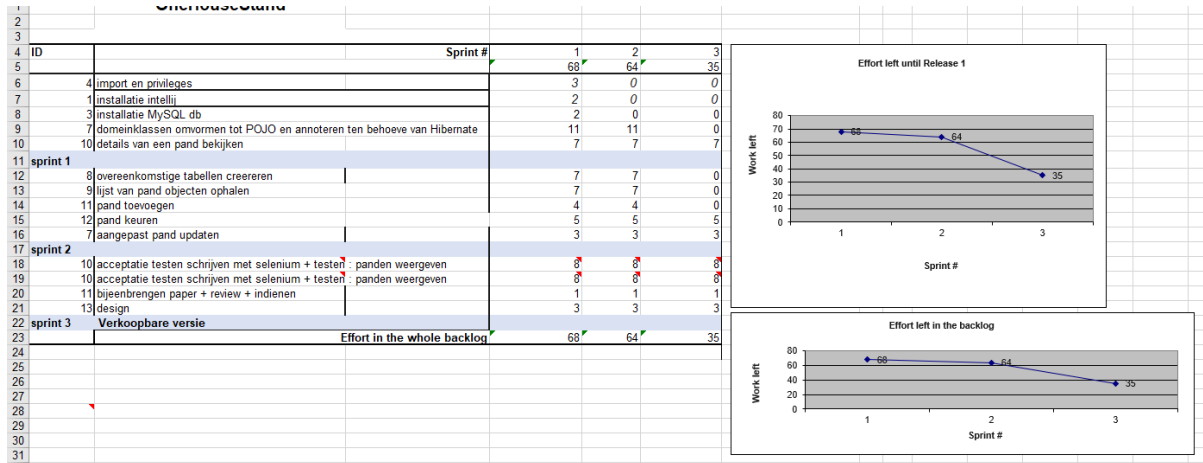
**Bram:** Sprint verliep goed, na de tweede dag was ik al klaar met mijn taak.

**Luis:** Omdat ik niet aanwezig kon zijn was ik niet klaar met mijn deel van mijn taak.

**Glenn:** de eerste taak bracht wat problemen mee die opgelost moesten worden in groepsverband, eens dit gedaan was had ik eigenlijk amper nog werk in deze sprint.

**Emmanuel:** Door persoonlijke omstandigheden (werk) heb ik de meeting gemist waardoor ik met vertraging liep.

Einde sprint 2:



## 2.5.3 Sprint 3

### sprint 3 – dag 1 (ma 15/4)

- Glenn:
  - o Gedaan:
    - Fabian geholpen met zijn cucumber test
  - o Wat ga ik nog doen:
    - Eigen cucumber test schrijven
- Fabian:
  - o Gedaan:
    - Cucumber test over: pand toevoegen
  - o Wat ga ik doen:
    - De test kijkt nu enkel of het pand aan de lijst is toegevoegd, dus wanneer Emmanuel klaar is met het weergeven van pand gegevens kan ik die gegevens ook verifiëren.
  - o Verwacht ik problemen?
    - Moeilijkste is gedaan, dus neen
- Bram:
  - o Gedaan:
    - Probleem oplossen met het toevoegen van een nieuw pand, ontdekt door de cucumber test.
  - o Wat ga ik doen:
    - Als aller laatste het design mooi maken.
  - o Verwacht ik problemen?
    - Neen, als iedereen op tijd klaar is met zijn deel niet.
- Luis:
  - o Gedaan:
    - Deel pand keuren proberen te implementeren, zonder succes
  - o Wat ga ik doen:

- Mijn errors zoeken.
- Verwacht ik problemen?
  - Ja, ik heb het moeilijk om verder te geraken omdat ik het ik het moeilijk heb om met dat programma te werken. Ik begrijp het niet.

### **sprint 3 – dag 2 (woe 17/4)**

- Luis:
  - Gedaan:
    - Probeerde update te doen maar dat liet ik momenteel vallen en nu concentreer ik me op delete
  - Wat ga ik doen:
    - Delete proberen te implementeren
  - Verwacht ik problemen?
    - Ja, door voorgaande problemen
- Bram:
  - Gedaan:
    - Prototype in Balsamiq gemaakt
  - Wat ga ik doen:
    - Design implementeren
  - Verwacht ik problemen?
    - Problemen neen, maar ik wacht wel op de anderen om een design te implementeren
- Fabian:
  - Gedaan:
    - Niets, ik wacht nog steeds op Emmanuel die het overzicht van panden toevoegen, zodat ik mijn cucumber test kan aanvullen
  - Wat ga ik doen:
    - Het zelfde als vorige keer.
  - Verwacht ik problemen?
    - /

### **sprint 3 – dag 3 (za 21/4)**

- Luis:
  - Gedaan:
    - Het verwijderen van een pand succesvol geïmplementeerd
  - Wat ga ik doen:
    - Code van update kunnen implementeren
  - Verwacht ik problemen?
    - Waarschijnlijk zoals altijd
- Glenn:
  - Gedaan:
    - Was al klaar
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?
    - /
- Bram:
  - Gedaan:
    - Design verbeteren
  - Wat ga ik doen:

- Niets, ik ben klaar
  - Verwacht ik problemen?
    - Neen
- Fabian:
  - Gedaan:
    - Was al klaar
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?
    - /
- Emmanuel:
  - Gedaan:
    - Niets.. maar ik ga stoppen, dus jullie mogen mijn werk nu doen.
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?
    - /

## 2.5.4 Sprint 4

### sprint 4 – dag 1 (za 21/4)

- Fabian:
  - Gedaan:
    - Authenticatie project van leerkracht proberen nabootsen
  - Wat ga ik doen:
    - Publieke pagina maken
  - Verwacht ik problemen?
    - /
- Bram:
  - Gedaan:
    - Aanpassen van een pand afwerken
  - Wat ga ik doen:
    - Een pand toevoegen verplaatsen van pagina
  - Verwacht ik problemen?
    - Nee
- Glenn:
  - Gedaan:
    - /
  - Wat ga ik doen:
    - Code bestuderen om mijn taken te kunnen uitvoeren volgens de werkwijze die tot nu toe gebruikt werd
  - Verwacht ik problemen?
    - nee
- Luis:
  - Gedaan:

- /
- Wat ga ik doen:
  - /
- Verwacht ik problemen?
  - /

#### **sprint 4 – dag 2 (zo 22/4)**

- Fabian:
  - Gedaan:
    - Pand bewerk pagina verwacht nu admin rol + lijst panden is publiek toegankelijk. Er is een login pagina
  - Wat ga ik doen:
    - Wanneer glenn en bram hun pagina's af hebben ga ik hier ook rechten op zetten + een registratie pagina maken
  - Verwacht ik problemen?
    - /
- Bram:
  - Gedaan:
    - Een pand verplaatsen naar een nieuwe pagina
  - Wat ga ik doen:
    - Niets, ik ben klaar
  - Verwacht ik problemen?
    - Nee
- Glenn:
  - Gedaan:
    - Code bestudeerd en geexperimenteerd om te begrijpen hoe alles in elkaar zit
  - Wat ga ik doen:
    - Bestaande pagina's kopiëren om een tweede detail pagina en tweede overzicht pagina te maken voor de huurders ipv de screeners
  - Verwacht ik problemen?
    - nee
- Luis:
  - Gedaan:
    - /
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?
    - /

#### **sprint 4 – dag 3 (ma 23/4)**

- Glenn:
  - Gedaan:
    - Serviceimplementatie aanvullen zodat er een methode is die enkel de goedgekeurde panden teruggeeft, en dit gebruikt om de nieuwe overzicht pagina voor huurders te vervolledigen
  - Wat ga ik doen:
    - Reservaties kunnen plaatsen via de nieuwe detail pagina die speciaal voor huurders dient.



- Verwacht ik problemen?
  - Mogelijk probleem met communicatie met de database, omdat ik dit tot nu toe nog niet heb moeten doen.
- Fabian:
  - Gedaan:
    - Registratie pagina toegevoegd
  - Wat ga ik doen:
    - Pagina's van glenn en luis beveiligen wanneer zij er klaar mee zijn
  - Verwacht ik problemen?
    - /
- Luis:
  - Gedaan:
    - /
  - Wat ga ik doen:
    - Beginnen met mijn taak huurder kan na huur het gehuurde pand reviewen
  - Verwacht ik problemen?
    - /
- Bram:
  - Gedaan:
    - Niets, ik ben klaar
  - Wat ga ik doen:
    - Niets, ik ben klaar
  - Verwacht ik problemen?
    - Nee

### **Sprint retrospectieve:**

- Fabian: Authenticatie is niet volledig afgeraakt. Er is een login formulier en als een screener inlogt kan hij een pand bewerken. Rollen voor syndicus en klant moeten nog gemaakt worden. dit zal ik tegen morgen in orde brengen
- Glenn: Lijst van panden voor huurders die enkel de panden toont die goedgekeurd zijn is afgeraakt, het maken van reservaties voor panden ook. Het verliep niet al te makkelijk omdat ik eerst de code moest begrijpen en dit toch wat tijd in beslag nam, maar eens dit lukte ging het vlot vooruit
- Bram: Het ging goed, ik heb mijn taken deze sprint volledig kunnen afwerken en ik was van niemand afhankelijk. Hierdoor kon ik meteen beginnen zonder te moeten wachten.
- Luis: Persoonlijk ben ik pas gisteravond aan mijn deel begonnen, ik had maandag AD test en dit heeft prioriteit.

## 2.5.5 Sprint 5

### **sprint 5 – dag 1**

- Fabian:
  - Gedaan:
    - Achterstand van vorige sprint was de dag zelf nog ingehaald, daarnaast kan de syndicus nu ook een mat aanduiden als “hersteld”
  - Wat ga ik doen:

- Een sign out knop toevoegen voor authenticatie
    -
- Bram:
  - Gedaan:
    - Een pand keuren
  - Wat ga ik doen:
    - Niets, ik ben klaar
  - Verwacht ik problemen?
    - Nee
- Glenn:
  - Gedaan:
    - Een mat toevoegen aan een pand
  - Wat ga ik doen:
    - Een mat markeren als versleten
  - Verwacht ik problemen?
    - ja
- Luis:
  - Gedaan:
    - Pagina reviewToevoegen toegevoegd
  - Wat ga ik doen:
    - Review pagina error vinden
  - Verwacht ik problemen?
    - Ja, nadat ik op reviewToevoegen pagina kom en daarna op review druk kom ik niet op de juiste pagina

## **sprint 5 – dag 2**

- Fabian:
  - Gedaan:
    - De sign out knop geïmplementeerd
  - Wat ga ik doen:
    - Wanneer een mat bewerkt wordt, moet het veld “geïmplementeerd op” nog aangepast worden
- Bram:
  - Gedaan:
    - Niets, ik ben klaar
  - Wat ga ik doen:
    - Niets, ik ben klaar
  - Verwacht ik problemen?
    - Nee
- Glenn:
  - Gedaan:
    - Html pagina’s aangepast en controller aangepast om lijst van matten te kunnen tonen
  - Wat ga ik doen:
    - Errors oplossen
  - Verwacht ik problemen?

- Enorm veel frustratie
- Luis:
  - Luis: Gedaan:
    - Alle nodige ServiceImplementatie en ReviewRepository gemaakt
  - Wat ga ik doen:
    - Ervoor zorgen dat ik in de juiste pagina zit met de juiste id van review en pand
  - Verwacht ik problemen?
    - Ja, ik kan mijn ServiceImplementatie niet correct laten gebruiken in mijn klantController

### **sprint 5 – dag 3**

- Fabian:
  - Gedaan:
    - Datum veld updaten bij mat herstelling
- Luis:
  - Luis: Gedaan:
    - Mijn error van van ReviewPagina in orde brengen
  - Wat ga ik doen:
    - Ervoor zorgen dat ik in de juiste pagina zit met de juiste id van review en pand
  - Verwacht ik problemen?
    - Ja, ik kan mijn ServiceImplementatie niet correct laten gebruiken in mijn klantController, de connectie is niet correct
- Bram:
  - Gedaan:
    - Niets, ik ben klaar
  - Wat ga ik doen:
    - Niets, ik ben klaar
  - Verwacht ik problemen?
    - Nee
    - /
- Glenn:
  - Gedaan:
    - Bug hunting
  - Wat ga ik doen:
    - Bug hunting
  - Verwacht ik problemen?
    - Enorm veel frustratie
- 

### **Sprint retrospective:**

- **Bram:** De sprint ging heel goed, ik was snel klaar met mijn taken waardoor ik mij kon concentreren op andere vakken.
- **Fabian:** Ik heb al mijn taken volbracht en heb hulp aangeboden waar nodig
- **Luis:** ik heb mijn taken niet volledig kunnen afmaken, ik heb problemen gekregen wanneer ik op de juiste Id van Review en pand wou terechtkomen in mijn object. En ik kom niet op de juiste pagina terecht.
- **Glenn:** Mijn taken zijn niet volledig af geraakt voor het einde van de sprint, de logica achter de functionaliteiten klopt maar de conentie met de HTML pagina's zorgen voor parsing errors die enorm frustrerend zijn om op te lossen

## 2.5.6 Sprint 6

### sprint 6 – dag 1

- Fabian:
  - Gedaan:
    - Panden api geschreven
  - Wat ga ik doen:
    - Api die reserveren van een pand mogelijk maakt
    - Cucumber test moet geupdate worden
- Luis:
  - Gedaan:
    - Mijn pagina van review werkte goed, mijn getmapping is in orde alleen krijg ik mijn postmapping niet werkend.
  - Wat ga ik doen:
    - Experimenteren om en zien als ik mijn fout kan vinden
  - Verwacht ik problemen?
    - Enorm veel frustratie
- Bram:
  - Gedaan:
    - Ik heb de consume API van mat bestellen geschreven
  - Wat ga ik doen:
    - Een mock api schrijven zodat ik de consume kan testen
  - Verwacht ik problemen?
    - Nee

### sprint 6 – dag 2

- Fabian:
  - Gedaan:
    - /
  - Wat ga ik doen:
    - /
- Luis:
  - Gedaan:
    - Geëxperimenteerd om mijn fout te vinden zonder succes
  - Wat ga ik doen:
    - Verder zoeken
  - Verwacht ik problemen?
    - Enorm veel frustratie

- Bram:
  - Gedaan:
    - Een mock api schrijven zodat ik de consume kan testen
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?
    - Nee

### **sprint 6 – dag 3**

- Fabian:
  - Gedaan:
    - Luis geholpen met zijn problemen bij het toevoegen van een review
    - Reservatie api
  - Wat ga ik doen:
    - Cucumber testen updaten
- Luis:
  - Gedaan:
    - Fabian en Glenn hebben me geholpen met mijn probleem, ik verstond de logica niet van hoe ik de review met de juiste pand id linkte
  - Wat ga ik doen:
    - Reviews pagina maken en ervoor zorgen dat Screener de reviews kan zien en de pand kan editen
- Bram:
  - Gedaan:
    - /
  - Wat ga ik doen:
    - /
  - Verwacht ik problemen?
    - Nee

### **Retrospective**

#### **Fabian:**

- Over het algemeen ging de sprint wat minder en is er maar 60% bereikt van wat er bereikt moest worden, ik veronderstel dat dit komt door de AD deadline die aan het naderen is.

#### **Luis:**

- Ik had het echt moeilijk om mijn probleem te vinden omdat ik niet altijd alles begreep, hierdoor was ik langzaam. Maar nu ben ik zeker dat ik beter mee ben. Ik hoop dat ik nu minder problemen zal krijgen met de rest.

#### **Bram:**

- Ook voor mij ging deze sprint veel moeilijker, de deadline van AD en DD zijn dichtbij, hier gaat nu een groot deel van mijn tijd naartoe.

## 2.5.7 Sprint 7

### **sprint 7 – dag 1**

- Bram:
  - Gedaan:

- Angular project met Fabian
  - Wat ga ik doen:
    - Consume mat herstellen
  - Verwacht ik problemen?
    - Nee
- Fabian:
  - Gedaan:
    - Angular project samen met Bram gemaakt
  - Wat ga ik doen:
    - Paper + acceptatie test updaten
- Luis:
  - Gedaan:
    - Verder aan mijn taak gewerkt en mijn probleem proberen te vinden
  - Wat ga ik doen:
    - PPT

### **sprint 7 – dag 2**

- Bram:
  - Gedaan:
    - Consume mat herstellen
    - Bug fixes in project
  - Wat ga ik doen:
    - Angular project met Fabian
  - Verwacht ik problemen?
    - Nee, ik heb nog niet veel ervaring met Angular, maar ik verwacht veel en goede documentatie.
- Fabian:
  - Gedaan:
    - Acceptatie test geüpdatet adhv beveiliging

### **sprint 7 – dag 3**

- Bram:
  - Gedaan:
    - Werken aan de paper en presentatie
  - Wat ga ik doen:
    - Niets, deze sprint is klaar
  - Verwacht ik problemen?
    - Nee
- Fabian:
  - Gedaan:
    - Toevoegen van een reservatie
    - Paper
- Glenn:
  - Gedaan:
    - Mock API's gemaakt voor Verimpex en Standup Circus
    - Microservice design
  - Wat ga ik doen:

- Sprint is klaar
  - Verwacht ik problemen?
    - Nee
- Luis:
  - Gedaan:
    - ppt
  - Wat ga ik doen:
    - Sprint is klaar

### **Sprint retrospective**

**Bram:** Deze sprint ging vrij goed, ik heb niet meer helemaal in het begin alles kunnen afwerken omdat ik veel tijd nodig had voor application development en database development

### **Fabian:**

Achterstanden zijn ingehaald, alles is up to date nu.

### **Glenn:**

Sprint verliep goed, wel tijd verloren op microservices.

### **Luis:**

Sprint verliep snel voor mij, ik had geprobeerd om zo veel mogelijk klaar te krijgen aangezien ik me daarna moest focussen op Application.

## 2.6 Prototype in Spring Boot – zwarte user stories

### Als verhuurder kan ik een pand voor verhuur aan OneHouseStand voorstellen

**Opmerking:** vermits we al een voorbeeld van een volledige CRUD hebben voorgesteld, samen met code, zullen we u deze hier enkel uitleg geven over wat de pagina doet, niet hoe de code werkt.

Als verhuurder kunne we op de lijst van panden terecht komen en hier op nieuw pand drukken om een nieuw pand toevoegen

### Lijst van panden

[Nieuw pand](#)

- 1 | goedgekeurd | 1.0 Euro/uur
- 2 | goedgekeurd | 100.0 Euro/uur
- 3 | goedgekeurd | 299.0 Euro/uur
- 4 | goedgekeurd | 120.0 Euro/uur

### Pand toevoegen

We hebben je aanvraag ontvangen, gelieve enkele dagen te wachten zodat wij je pand kunnen verwerken.

HuisNummer:

PrijsPerUur:

StraatNaam:

### Lijst van panden voor screener

- 1 | goedgekeurd | 1.0 Euro/uur [Update](#)
- 2 | goedgekeurd | 100.0 Euro/uur [Update](#)
- 3 | goedgekeurd | 299.0 Euro/uur [Update](#)
- 4 | goedgekeurd | 120.0 Euro/uur [Update](#)
- 5 | ongekeurd | 1.0 Euro/uur [Update](#)
- 6 | ongekeurd | 1.0 Euro/uur [Update](#)

Wanneer een aanvraag ingediend wordt, is het zichtbaar op de panden pagina van de screener. Nog niet op de publieke lijst dus!

### Als screener kan ik een pand aan OneHouseStand toevoegen

### Edit pand

Pand id: 6  
Status: ongekeurd  
Bron: ohs  
HuisNummer:

PrijsPerUur:

StraatNaam:

### Matten

[Mat Toevoegen](#)

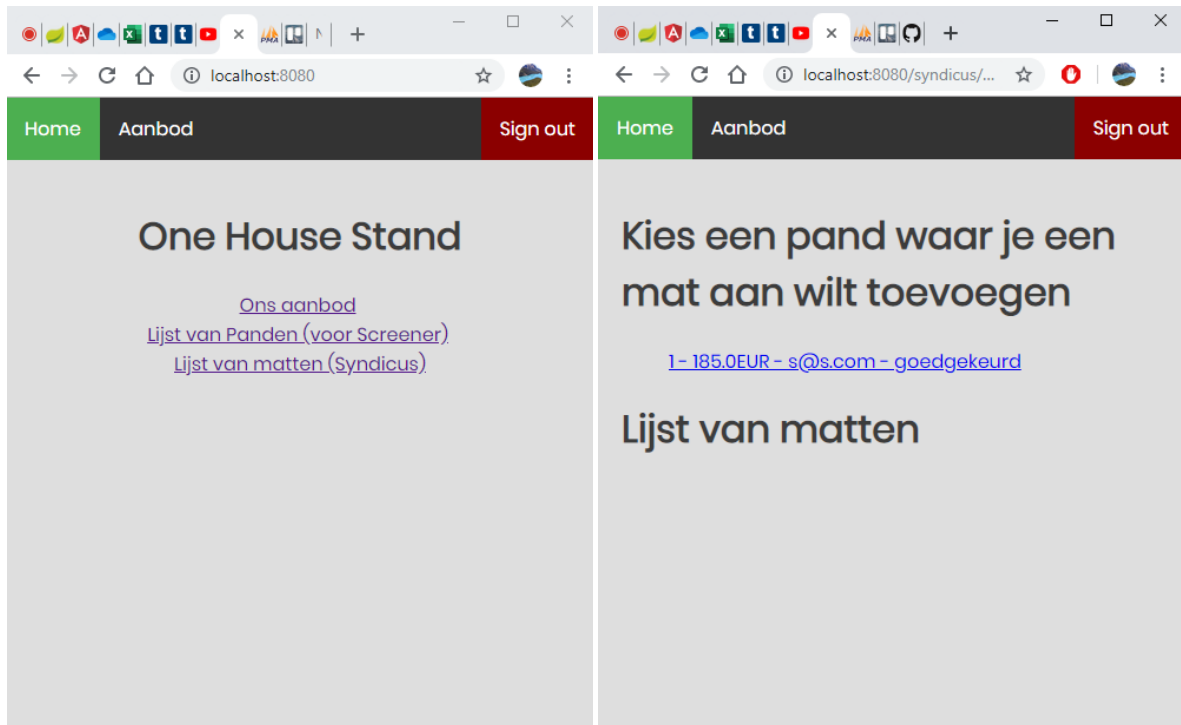
Wanneer de screener uit de lijst van panden een element selecteert komt hij op de Edit pand pagina.

Hier kan de klikken op “Goedkeuren” of “Afkeuren”. Wanneer het pand goedgekeurd is, is het pand zichtbaar op de publieke lijst van panden. De knoppen goedkeuren en afkeuren zijn enkel zichtbaar wanneer de status van een pand ongekeurd is.

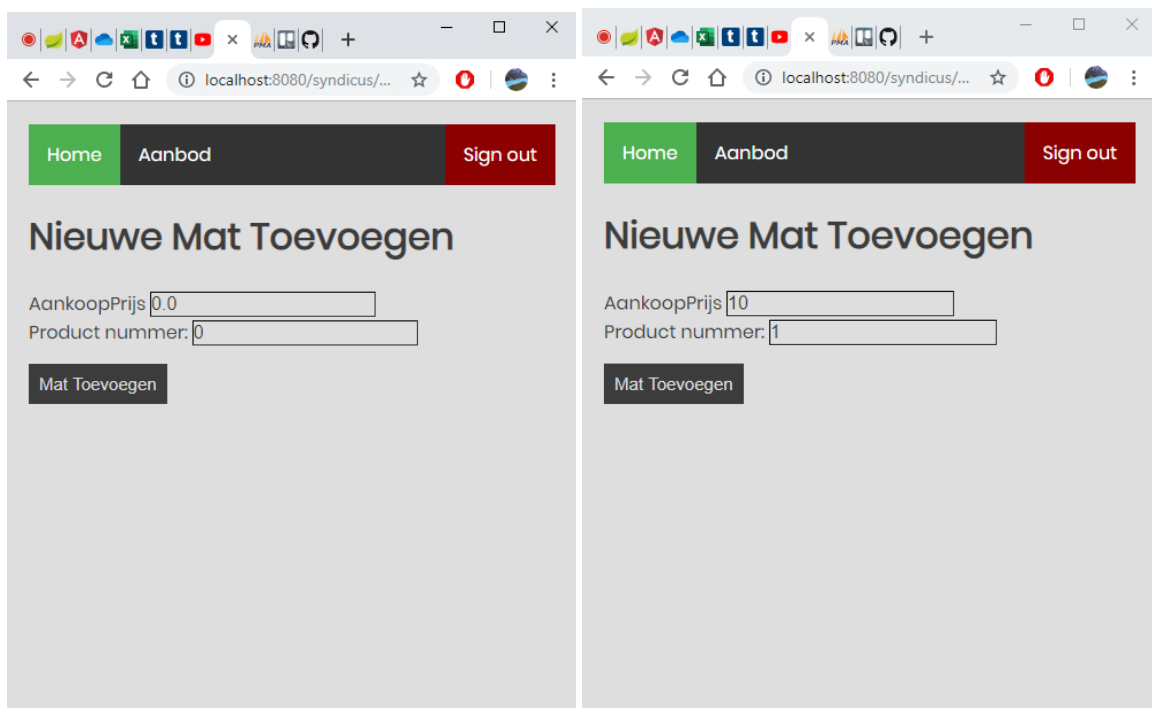


## Als syndicus kan ik een pand van de nodige Verimpex-matten voorzien

Als syndicus komen we op ons homepagina terecht. Daarna klikken we op lijst



We beginnen op de startpagina. We navigeren naar Lijst van matten (syndicus) en melden ons aan met een account dat toebehoort tot het syndicus. Hierna zien we een overzicht van de panden en een lijst van de matten eronder. Als je beslist om een mat toe te voegen aan een pand, klik je op een van de panden.

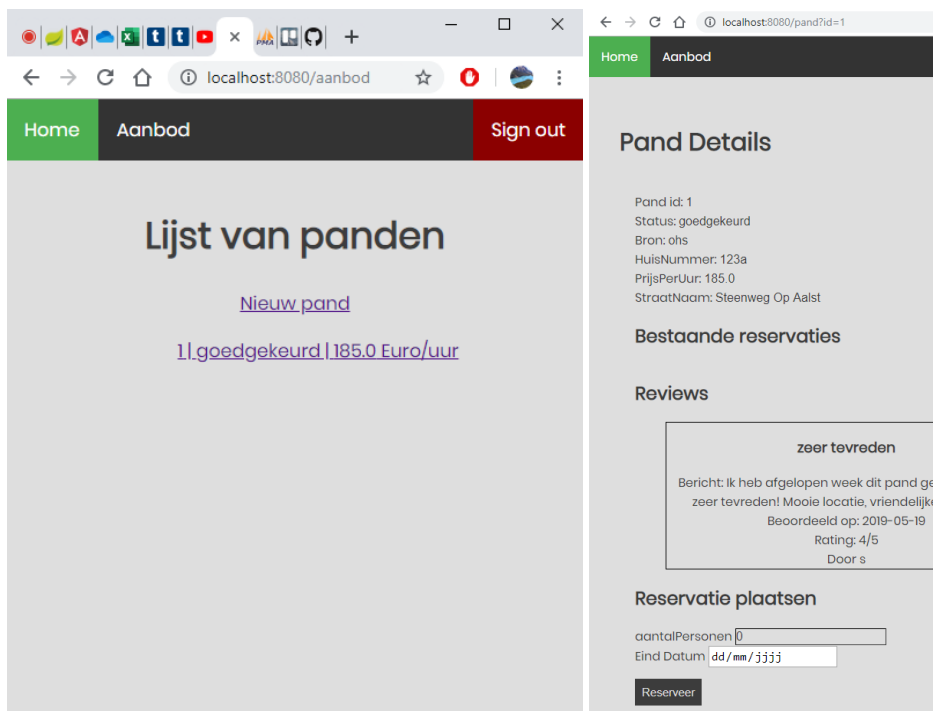


Vervolgens kom je terecht op deze pagina. Hier kan je de aankoopprijs van de nieuwe mat invoeren, evenals het productnummer van de mat. Als je vervolgens op mat toevoegen klikt wordt deze mat toegevoegd.



Hier kunnen we zien dat de mat toegevoegd werd. We kunnen opmerken dat de status automatisch op "goed" gezet wordt, en dat laatst gecheckt de datum van vandaag krijgt.

### Als huurder kan ik een geschikt pand voor mijn evenement zoeken en vastleggen



We beginnen op de pagina met het aanbod van de panden. We klikken op het pand dat we zouden willen huren. We zien dat er nog geen bestaande reservaties zijn.

### Reservatie plaatsen

Gelieve de problemen op te lossen.

aantalPersonen  Er moet minstens 1 persoon aanwezig zijn

Eind Datum  Vul een geldige datum in

[Reserveer](#)

### Pand Details

Pand id: 1  
Status: goedgekeurd  
Bron: ohs  
HuisNummer: 123a  
PrijsPerUur: 185.0  
StraatNaam: Steenweg Op Aalst

### Bestaande reservaties

- 2 personen 2019-05-24

### Reviews

Als we een foute input doen kunnen we zien dat we foutmeldingen krijgen. Na een juiste input zien we dat de reservatie is toegevoegd

### Als syndicus kan ik een probleem met de matten (laten) doorgeven aan Verimpex

We melden ons aan als syndicus en gaan naar het overzicht van de matten. We zien een mat met status 'goed' en klikken op update.

Update'."/>

← → ↻ 🏠 localhost:8080/syndicus/matten

Home Aanbod

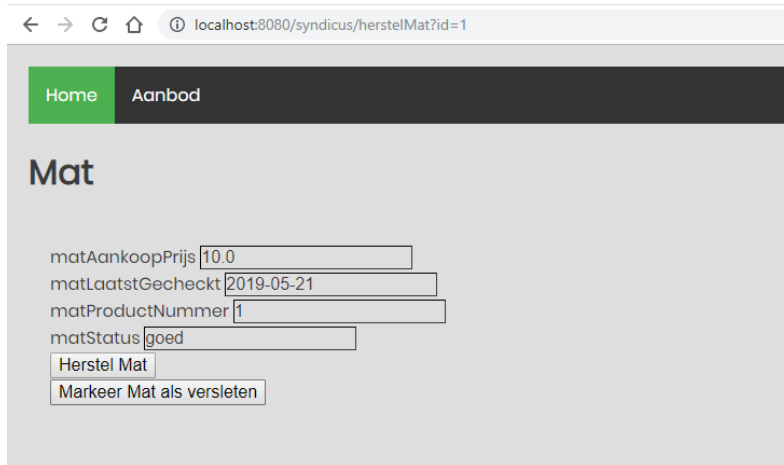
## Kies een pand waar je een mat aan wilt toevoegen

[1 - 185.0EUR - s@s.com - goedgekeurd](#)  
[2 - 260.0EUR - s@s.com - goedgekeurd](#)  
[3 - 105.0EUR - s@s.com - goedgekeurd](#)

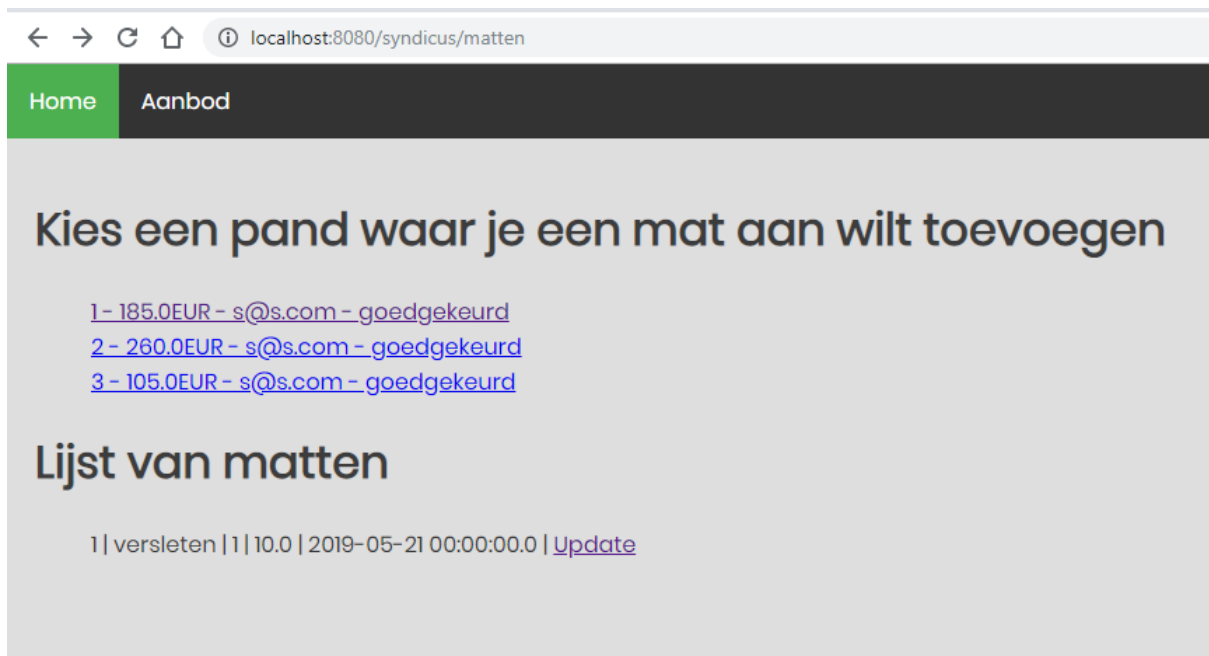
### Lijst van matten

1 | goed | 1 | 10.0 | 2019-05-19 00:00:00.0 | [Update](#)

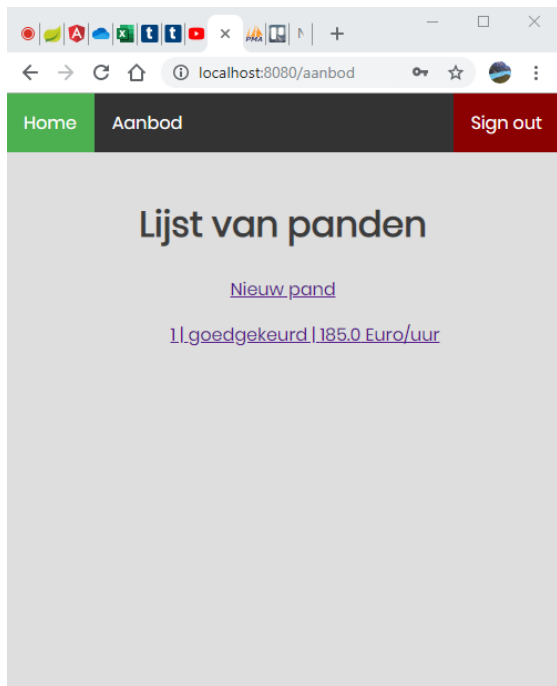
Na het klikken op update komen we terecht op de volgende pagina en klikken we op markeer mat als versleten:



Als we nu terug gaan naar de pagina met alle matten kunnen we zien dat de status van de mat geüpdatet is, en nu versleten is:

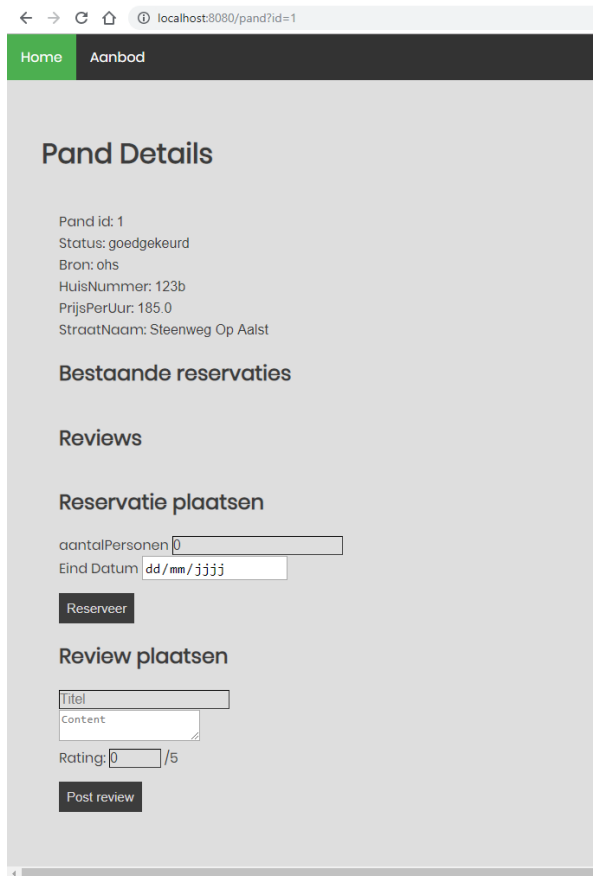


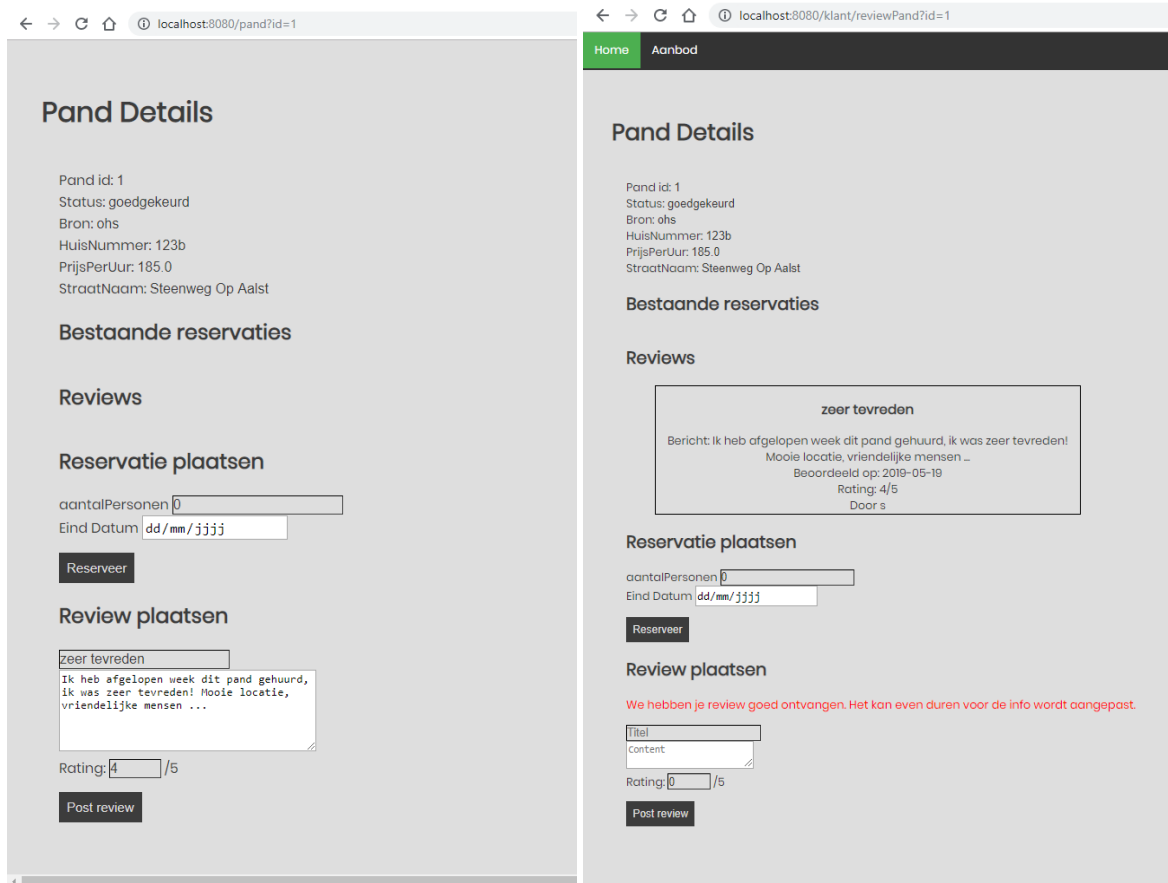
Als huurder kan ik na huur het gehuurde pand reviewen



Allereerst gaan we via aanbod als een klant naar een pand waar we een review over willen schrijven.

Op de pagina van een pand kunnen we onderaan een review plaatsen. Bij het plaatsen van een review geef je een titel, content en een rating op 5 mee.





Als je vervolgens op post review klikt zal je een bericht krijgen dat de review goed ontvangen is, en zal je zien dat de review nu bekeken kan worden onder het titeltje reviews.

**Als screener kan ik de informatie over het pand aanpassen op basis van de reviews door de huurder**

Als screener ga ik naar het aanbod, en klik ik op een pand.



## Pand Details

Pand id: 1  
Status: goedgekeurd  
Bron: ohs  
HuisNummer: 123b  
PrijsPerUur: 185.0  
StraatNaam: Steenweg Op Aalst

### Bestaande reservaties

### Reviews

zeer tevreden	Fout adres
Bericht: Ik heb afgelopen week dit pand gehuurd, ik was zeer tevreden! Mooie locatie, vriendelijke mensen ... Beoordeeld op: 2019-05-19 Rating: 4/5 Door s	Bericht: Het adres van dit pand is fout! het is op 123a! Beoordeeld op: 2019-05-19 Rating: 1/5 Door s

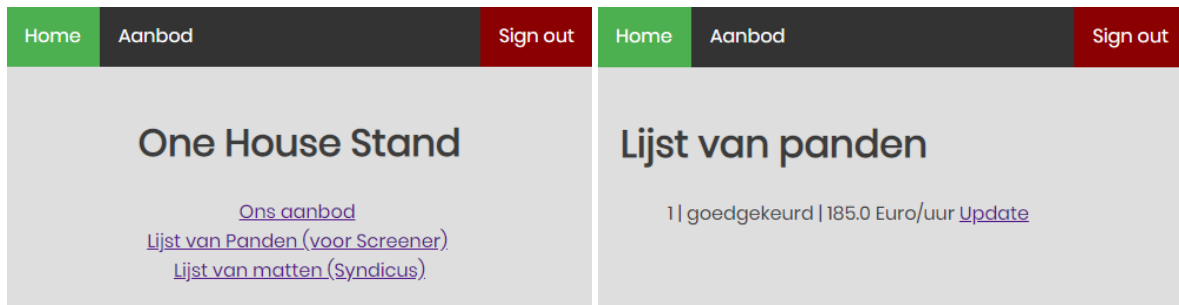
### Reservatie plaatsen

aantalPersonen   
Eind Datum

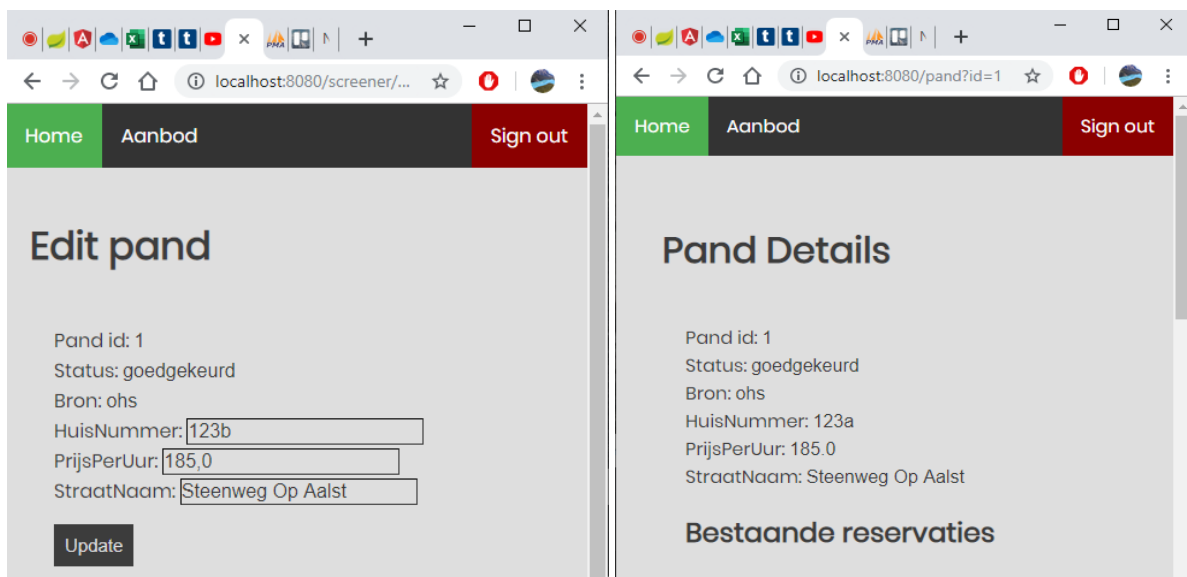
### Review plaatsen

Rating:  /5

Daar kan ik zien dat er een review is, met de titel fout adres. Uit de beschrijving blijkt dat het adres 123a is in plaats van 123b. Als screener moet ik dit rechtzetten.



Aangemeld als screener navigeer ik naar de home pagina en klik ik op lijst van panden (Screener). Daar zie ik het pand staan, en klik ik op update.



Ik kan zien dat het adres momenteel fout is (123b), volgens de review zou het 123a moeten zijn. We passen dit aan en klikken op update. We worden nu teruggebracht naar de lijst van panden. Als we terug naar aanbod gaan en op het desbetreffende pand klikken, zullen we nu zien dat het huisnummer aangepast is.

## 2.7 Validatie en Authenticatie

In onderstaande tabel kan u zien welke rollen elke user had, dit is ook effectief toegepast op de applicatie, een user kan enkel de delen van de applicatie gebruiken die hij/zij volgens deze tabel mag gebruiken.

Originele rollen:



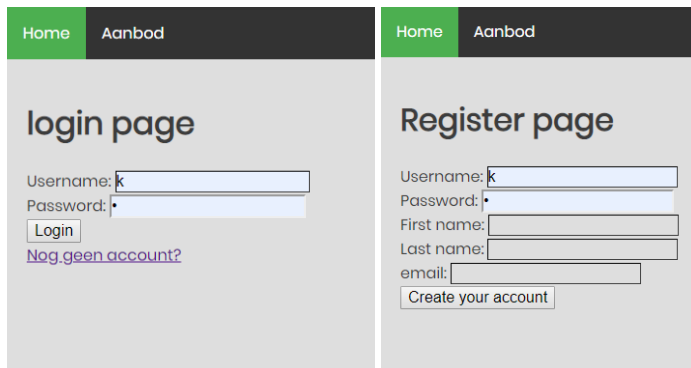
	huurder	verhuurder	syndicus	screener
pand toevoegen		X		
pand bewerken				X
pand verwijderen				X
pand lezen	X	X	X	X
review toevoegen	X			
review verwijderen	X			X
review bewerken				X
review lezen	X			X
mat toevoegen			X	X
mat verwijderen			X	X
mat bewerken			X	X
mat lezen			X	X

Bij de uitwerking van authenticatie werd duidelijk dat huurder en verhuurder eigenlijk 1 rol konden zijn, resulterend in:

	iedereen	klant	syndicus	screener
pand toevoegen		X		
pand bewerken				X
pand verwijderen				X
pand lezen	X			
review toevoegen		X		
review verwijderen				
review bewerken				
review lezen	X			
mat toevoegen			X	
mat verwijderen			X	
mat bewerken			X	
mat lezen			X	X

id	email	first_name	last_name	password	role	username
1	k@h.com	k	k	{bcrypt}\$2a\$10\$SC1jNPtskmQmsW40FHDmRegkJT1qFEcKqR0...	KLANT	k
2	screener@h.com	s	s	{bcrypt}\$2a\$10\$6sYp4K3yYwm827OpuMW2gerTtjZH15vbpzG...	SCREENER	screener
3	s@h.com	s	s	{bcrypt}\$2a\$10\$Ah1ov7cg32s1C0bW///xJQeRcfQQ4kFrBjdf...	SYNDICUS	s

Hierboven ziet u een screen van de user tabel waarin u ziet dat de passwords geencrypteerd zijn. Een user omvat een email, voornaam, achternaam, wachtwoord, rol en gebruikersnaam.



Een gebruiker kan een account aanmaken, dit is standaard van de rol KLANT. Hij kan dus panden bekijken, reserveren, toevoegen, op een pand reviewen. Om een andere rol te maken, zou ik aanraden eerst te registreren, en vervolgens zelf in de databank de rol aan te passen in "SCREENER" OF "SYNDICUS".

```

@Autowired
DataSource datasource;

@Override
protected void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth
        .jdbcAuthentication()
        .dataSource(datasource)
        .usersByUsernameQuery("select username, password, true as enabled from users where username=?")
        .authoritiesByUsernameQuery("select username, role as authority from users" +
            " where username=?");
}

@Override
protected void configure(HttpSecurity http) throws Exception {
    http
        .formLogin().formLoginConfigurer<HttpSecurity>
        .loginPage("/login") FormLoginConfigurer<HttpSecurity>
        .defaultSuccessUrl( defaultSuccessUrl: "/aanbod", alwaysUse: false) FormLoginConfigurer<HttpSecurity>
        .failureUrl("/login-error") FormLoginConfigurer<HttpSecurity>
        .and() HttpSecurity
        .logout().logoutSuccessUrl("/byebye") LogoutConfigurer<HttpSecurity>
        .and() HttpSecurity
        .httpBasic() HttpBasicConfigurer<HttpSecurity>
        .and() HttpSecurity
        .authorizeRequests() ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/pand").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/aanbod/**").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/login").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/registreer*").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/api/**").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry

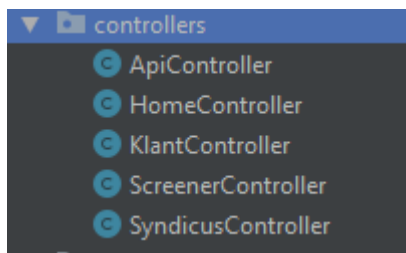
        .antMatchers(antMatchers: "/screeener/**").hasAuthority("SCREENER") ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/klant/**").hasAuthority("KLANT") ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/syndicus/**").hasAuthority("SYNDICUS") ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry

        .antMatchers(antMatchers: "/byebye*").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/403*").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .antMatchers(antMatchers: "/style.css").permitAll() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry

        .anyRequest().authenticated() ExpressionUriAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUriRegistry
        .and() HttpSecurity
        .csrf().disable() //stackoverflow.com/questions/24655203/enable-http-request-post-in-spring-boot
        .exceptionHandling().accessDeniedPage("/403");
}

/**
 * The default PasswordEncoder is now DelegatingPasswordEncoder which is a non-passive change.
 * This change ensures that passwords are now encoded using BCrypt by default
 */
@Bean
public static PasswordEncoder passwordEncoder() {
    return PasswordEncoderFactories.createDelegatingPasswordEncoder();
}

```



Hier kan de structuur zien van de controllers. Elke controller wordt toegewezen aan een bepaalde rol, met uitzondering op de HomeController en ApiController.

De homecontroller is publiek toegankelijk, en is beschikbaar via '/' in de browser url.

```
@Controller  
public class HomeController {
```

```
@RequestMapping("/klant")  
@Controller  
@PostAuthorize("#model.get('role').user.username == authentication.principal.username")  
@Secured({"KLANT"})  
public class KlantController {
```

```
@RequestMapping("/screener")  
@Controller  
@PostAuthorize("#model.get('role').user.username == authentication.principal.username")  
@Secured({"SCREENER"})  
public class ScreenerController {
```

```
@RequestMapping("/syndicus")  
@Controller  
@PostAuthorize("#model.get('role').user.username == authentication.principal.username")  
@Secured({"SYNDICUS"})  
public class SyndicusController {
```

Hierboven kan je de 3 beveiligde controllers zien, elke controller is beveiligd aan zijn eigen toebehorende rol. Vervolgens kan je content van elke controller navigeren via de naam van de rol; /syndicus, /screener, /klant.

## Inloggen en registreren:

Inloggen gebeurt achter de schermen, registreren gebeurt via een form die verwerkt wordt door de publieke HomeController. De controller roept vervolgens "verwerkRegistratie()" op binnen de OhServiceImplementatie. Hier wordt de nieuwe gebruiker zijn rol en wachtwoord toegewezen. + geencrypteerd!

```
@PostMapping(path = "/registreer")
public String registreerUitvoeren(@Valid User u, Errors err, Model model) {
    if (err.hasErrors())
        return "registreer";
    service.verwerkRegistratie(u);
    return "login";
}
```

```
public void verwerkRegistratie(User u) {
    u.setRole("KLANT");
    u.setPassword("{bcrypt}" + BCrypt.hashpw(u.getPassword(), BCrypt.gensalt()));
    addUser(u);
}
```

Home

Aanbod

403 page

We are awfully sorry, but you are not allowed what you are trying to do

Wanneer een gebruiker naar een pagina navigeert waar hij niet op toegewezen is, wordt hem verteld dat hij niet toegestaan is naar deze pagina te gaan.

## 2.8 REST, Rode en Groene user stories

### 2.8.1 [provide] een OneHouseStand-ruimte via Start-Up-circus te huur aanbieden

Startup circus kan gebruik maken van onze API om een pand op te halen en te vertonen op hun site.

```
@RequestMapping(path = "/api", produces = "application/json")
@RestController
@CrossOrigin(origins = "**")
public class ApiController {

    @Autowired
    private OhsService ohsService;

    @GetMapping("/pand")
    public @ResponseBody
    Pand apiPand(@RequestParam("id") int id) { return ohsService.geefPand(id); }

    @GetMapping("/panden")
    public @ResponseBody
    Iterable<Pand> apiPanden() { return ohsService.geefAlleGoedgekeurdePanden(); }

    @RequestMapping(value = {"/reserveer"}, method = RequestMethod.POST)
    @ResponseStatus(HttpStatus.CREATED)
    public @ResponseBody
    Reservatie reserveer(@RequestBody Reservatie reservatie, HttpServletResponse response, @RequestParam(
        Reservatie r = ohsService.processReservatie(reservatie, pandId);
        return reservatie;
    }
}
```

Ook kan a.d.h.v. de API een reservatie geplaatst worden op een van onze panden.

We bieden volgende functionaliteiten aan:

- Het getten van een pand op basis van een ID
- Het getten van een lijst van alle panden in onze db
- Het plaatsen van een reservatie op een bepaald pand

Start-up-circus kan onze API dus gebruiken om 1 specifieke of een lijst van alle panden op te vragen en te vertonen op hun website, wanneer een klant dan een reservatie wil plaatsen kan start-up-circus deze doorgeven aan onze API waardoor er in ons systeem geregistreerd wordt voor wanneer dit pand gereserveerd is (dit voorkomt dat wij het pand zouden verhuren op hetzelfde moment).

### 2.8.2 [consume] de matten in de verhuurde ruimtes kunnen bestellen en laten herstellen door Verimpex

Voor deze story hebben we een mock api gemaakt om de api van Verimpex na te bootsen sinds deze groep achterstond en samenwerking vrij moeilijk verliep.

```

13 @RequestMapping("/verimpexapi")
14 @RestController
15 @CrossOrigin(origins = "**")
16 public class MockVerimpexApiController {
17     @RequestMapping(value={"/matBestellen"},method=RequestMethod.GET)
18     public @ResponseBody Mat[] matBestellen(){
19         Mat m1 = new Mat();
20         m1.setStatus("goed");
21         Mat m2 = new Mat();
22         m2.setStatus("goed");
23         Mat m3 = new Mat();
24         m3.setStatus("goed");
25         return new Mat[]{m1,m2,m3};
26     }
27     @RequestMapping(value={"/matHerstellen"},method=RequestMethod.PUT)
28     @ResponseBody Mat matHerstellen(@RequestBody Mat mat){
29         mat.setStatus("goed");
30         return mat;
31     }
32 }

```

Wij (OneHouseStand) zouden gebruik maken van deze API door de syndicus telkens een nieuwe mat te doen opvragen om dan door te geven aan een pand in ons systeem, eens deze mat tekens van schade toont zal de syndicus dit melden aan Verimpex en wordt dit doorgegeven via de MeldMat methode in de Verimpex API.

Ten slotte wanneer het nodig is om de mat te repareren wordt dit vermeld via de HerstelMat methode in de Verimpex API, en regelt Verimpex het herstellen van de mat.

### 2.8.3 [consume] Start-Up-circus-rumtes via OneHouseStand kunnen verhuren

Voor deze story hebben we een mock api gemaakt om de api van Start-Up-circus na te bootsen sinds deze groep achterstond en samenwerking vrij moeilijk verliep.

```

@RequestMapping("/Circusapi")
@Controller
@CrossOrigin(origins = "**")
public class MockCircusApiController {

    @Autowired
    private OhsService ohsService;
    private List<Pand> fakeCircusDB = Arrays.asList(new Pand(), new Pand(), new Pand());

    @GetMapping("/pand")
    public @ResponseBody
    Pand apiPand(@RequestParam("id") int id) {
        Pand p = fakeCircusDB.get(id);
        return p;
    }

    @RequestMapping(value = {"/reserveerPand"}, method = RequestMethod.POST)
    @ResponseStatus(HttpStatus.OK)
    public @ResponseBody
    Reservatie reserveer(@RequestBody Reservatie reservatie, HttpServletResponse response, @RequestParam("pandId") int pandId) {
        Pand pand = fakeCircusDB.get((int)pandId);
        pand.reserveer(reservatie);
        return reservatie;
    }

    @GetMapping("/panden")
    public @ResponseBody
    Iterable<Pand> apiPanden() { return fakeCircusDB; }
}

```

Wij (OneHouseStand) zouden deze API gebruiken door de lijst van panden op te vragen, deze te mergen met onze database samen met een extra kolom "oorsprong" waarin dan Start-up-circus zou staan, en wanneer een klant 1 van deze panden probeert te reserveren dan doen we een oproep naar de API van start-up-circus en geven we de reservatie mee samen met het ID van het pand in kwestie.

Wanneer “/aanbod” opgehaald gerequest wordt, zullen er panden uit onze databank gehaald worden en panden uit de api. Deze worden vervolgens gemerged in 1 lijst, die aan het model wordt toegevoegd.

```
@GetMapping("/aanbod")
public String aanbod(Model model) {
    RestTemplate rt = new RestTemplate();
    Pand[] circuspanden =
rt.getForObject("http://localhost:8080/circusapi/circuspanden", Pand[].class);
    assert circuspanden != null;

    Iterable<Pand> panden = ohsService.geefAlleGoedgekeurdePanden();
    ArrayList<Pand> mergedPanden = new ArrayList<>(Arrays.asList(circuspanden));
    for (Pand pand : panden) {
        mergedPanden.add(pand);
    }

    model.addAttribute("lijstPanden", mergedPanden);
    return "aanbod";
}
```

Natuurlijk, omdat we met een mockup api werkten en geen echte ids hadden, konden we de details van dit pand niet weergeven. Hierdoor zal wanneer een pand van startup circus geselecteerd is, genavigeerd worden naar een andere pagina.

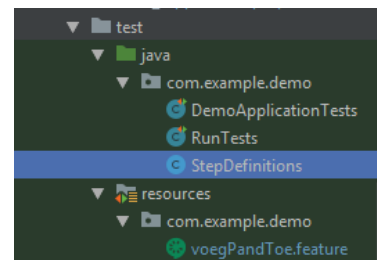
```
@GetMapping("/pand")
public String pandDetailForm(Model model, @RequestParam("id") long id) {
    if (id == 0) {
        return "startupcircuspand";
    } else {
        Pand pand = ohsService.geefPand((int) id);
        model.addAttribute("reviews", ohsService.getReviewsPandById(id));
        model.addAttribute("reservatie", new Reservatie());
        model.addAttribute("pand", pand);
        model.addAttribute("review", new Review());
        return "pand";
    }
}
```



## 2.9 Cucumber test

Deze werd al eens gemaakt, maar is nu geüpdatet met de authenticatie en aanpassingen op bepaalde pagina's.

```
voegPandToe.feature x
1 Feature: Add House
2
3 It should be possible to add a house to the list
4 Just after adding the house,
5 the list is shown and contains the newly added house
6
7 Scenario: Add one house
8   Given I am signed in as a klant
9   And I am on the page where I can add a new house
10  When I enter "800" in the prijsPerUur field
11  And I enter "Boulevardlaan" in the straatNaam field
12  And I enter "60" in the huisNummer field
13  And I press on the Submit button
14  And I sign in as screener
15  And i go to the ongekeurde panden page
16  Then I should see a list containing "ongekurd", "800"
17  And I close the browser
```



Oorspronkelijk werd er niet gevalideerd welke type gebruiker de actie deed, dus kon iedereen alles doen. Nu is dit niet meer mogelijk, en kan enkel de klant een pand toevoegen, en kan enkel de screener dat nieuwe toegevoegde (ongekeurde) pand bekijken.

```
@Given("I am signed in as a klant$")
public void iAmSignedInAsAKlant() throws Throwable {
    //Prepare driver
    System.setProperty("webdriver.gecko.driver", "D:\\onedrive\\OneDrive - ODISEE\\Fabian\\Jaar 2\\Semester 2\\Software\\w6\\geckodriver.exe");
    driver = new FirefoxDriver();

    //Sign out, if logged in
    driver.navigate().to("http://localhost:8080/");
    driver.findElement(By.name("logout")).click();
    new WebDriverWait(driver, timeOutInSeconds: 10).until(ExpectedConditions
        .textToBePresentInElementLocated(By.tagName("body"), text: "Je bent uitgelogd"));

    //Sign in
    driver.navigate().to("http://localhost:8080/login");
    new WebDriverWait(driver, timeOutInSeconds: 10).until(ExpectedConditions
        .textToBePresentInElementLocated(By.tagName("body"), text: "login page"));
    driver.findElement(By.id("username")).sendKeys("k");
    driver.findElement(By.id("password")).sendKeys("k");
    driver.findElement(By.name("submit")).click();

    new WebDriverWait(driver, timeOutInSeconds: 2); //pagina kan variëren, dus kan niet dynamisch
}

@Given("I am on the page where I can add a new house$")
public void iAmOnThePageWhereICanAddANewHouse() throws Throwable {
    driver.navigate().to("http://localhost:8080/klant/pandToevoegen");
}

@When("I enter \"([^\"]*)\" in the \"([^\"]*)\" field$")
public void iEnterInThePrijsPerUurField(String number, String fieldName) throws Throwable {
    driver.findElement(By.id(fieldName)).sendKeys(number);
}

@And("I press on the Submit button$")
public void iPressOnTheSubmitButton() throws Throwable {
    driver.findElement(By.name("submit")).click();
}

@And("I sign in as screener$")
public void iSignInAsScreener() {
    driver.navigate().to("http://localhost:8080/login");

    new WebDriverWait(driver, timeOutInSeconds: 10).until(ExpectedConditions
        .textToBePresentInElementLocated(By.tagName("body"), text: "login page"));

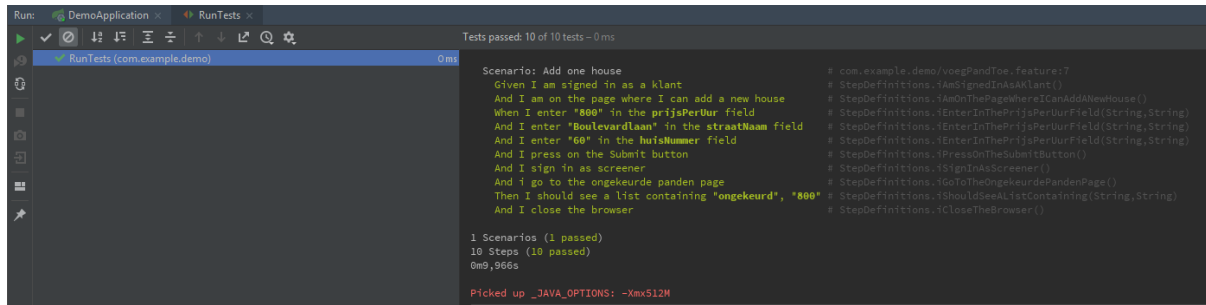
    driver.findElement(By.id("username")).sendKeys("screener");
    driver.findElement(By.id("password")).sendKeys("screener");
    driver.findElement(By.name("submit")).click();
}
```



Hier wordt er in het begin alles klaar gezet om dan het pand toe te voegen, vervolgens wordt er dan als screener ingelogd om het resultaat te kunnen bekijken.

```
@Then("^I should see a list containing \"([^\"]+)\"$, \"([^\"]+)\"$")
public void iShouldSeeAListContaining(String status, String prijs) throws Throwable {
    By bodyByTag = By.tagName("body");
    String newBodyText = driver.findElement(bodyByTag).getText();
    Assert.assertTrue( message: "Did not find price in text: " + prijs + "\n", newBodyText.contains(prijs));
    Assert.assertTrue( message: "Did not find house state in text: " + status + "\n", newBodyText.contains(status));
}
```

Hier wordt er getest dat zowel de prijs als de status van het net toegevoegde pand beschikbaar zijn.



```
Run: DemoApplication x RunTests x
Tests passed: 10 of 10 tests - 0 ms
RunTests (com.example.demo) 0 ms

Scenario: Add one house
  Given I am signed in as a klant
  And I am on the page where I can add a new house
  When I enter "800" in the prijsPerLur field
  And I enter "Boulevardlaan" in the straatNaam field
  And I enter "60" in the huisNummer field
  And I press on the Submit button
  And I sign in as screener
  And I go to the ongekeurde panden page
  Then I should see a list containing "ongekurd", "800"
  And I close the browser

1 Scenarios (1 passed)
10 Steps (10 passed)
0m9,966s
Picked up _JAVA_OPTIONS: -Xmx512M

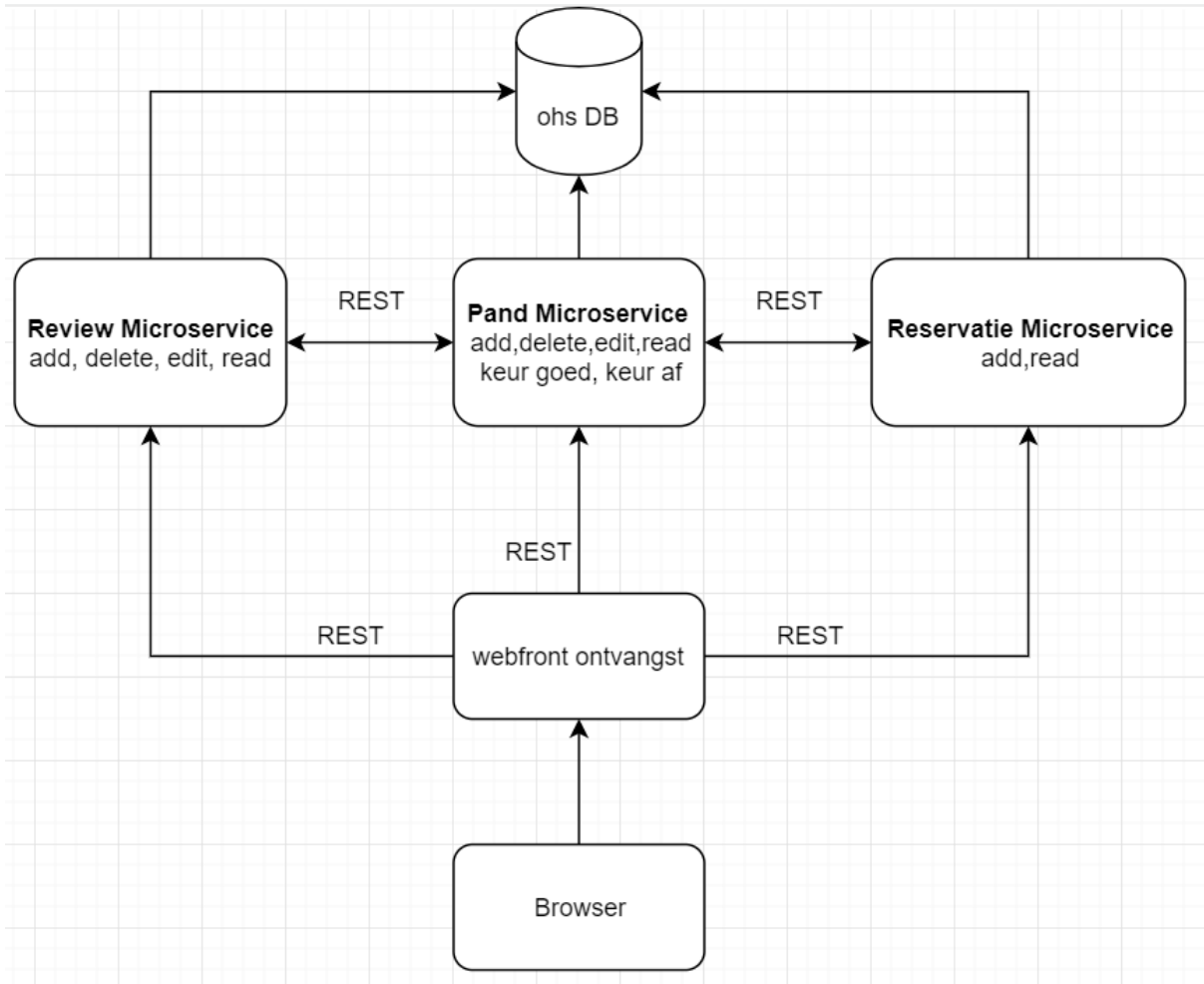
# com.example.demo.voegPandToe.feature:7
# StepDefinitions: IAmSignedInAsAKlant()
# StepDefinitions: IAmOnThePageWhereICanAddNewHouse()
# StepDefinitions: IEnterInThePrijsPerLurField(String, String)
# StepDefinitions: IEnterInThePrijsPerLurField(String, String)
# StepDefinitions: IEnterInThePrijsPerLurField(String, String)
# StepDefinitions: IPressOnTheSubmitButton()
# StepDefinitions: ISignInAsAScreener()
# StepDefinitions: IGoToTheOngekeurdePandenPage()
# StepDefinitions: IShouldSeeAListContaining(String, String)
# StepDefinitions: ICloseTheBrowser()
```

Wanneer de test gerund wordt, zien we dat de test succesvol is. Al zou dit niet het geval zijn wanneer er geen screener/klant profielen beschikbaar zouden zijn!

## 2.10 Microservices architectuur

Onze webapplicatie zouden kunnen omvormen om gebruik te maken van microservices, dit zouden we doen door alle interacties omtrent reviews, reservaties, en panden elk hun eigen microservice te geven.

Op onderstaand diagram ziet u hoe de structuur van de webapp er zou uitzien als men deze microservice structuur zou implementeren.



Elke microservice maakt natuurlijk gebruik van hun respectievelijke repositories (review gebruikt reviewRepository, reservatie gebruikt reservatieRepository, en pand pandrepository).

Elke microservice is bereikbaar via de REST en ze communiceren ook met elkaar via REST om bv een reservatie of review toe te voegen aan een pand.

## 2.11 Webfront en REST backend

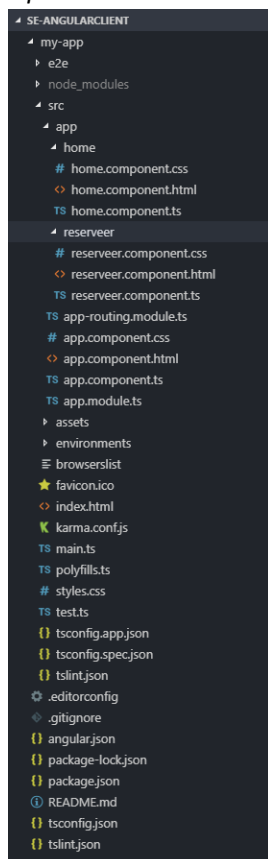
Wij hebben ervoor gekozen om met Angular te werken voor onze front-end. Het leek ons leuk om hier wat mee te experimenteren en nieuwe dingen te ontdekken. We hebben de documentatie gevolgd, en dit ging vrij vlot.

### Hoe dit project runnen?

Om dit project te runnen moet je angular cli installeren door ***npm install -g @angular/cli***. Hierna moet je in de folder van het project gaan, ***npm install*** (Om alle dependencies te installeren) en ***ng serve -o*** uitvoeren. Dit laatste commando zorgt ervoor dat een server opgestart wordt, en de browser het project opent. Voor meer info, lees hier: <https://angular.io/guide/quickstart>

Als alles goed gaat zul je de lijst van panden zien die momenteel goedgekeurd zijn. Je kan de id zien, de locatie, prijs per uur en een knop om te reserveren.

### Opbouw van de webclient:



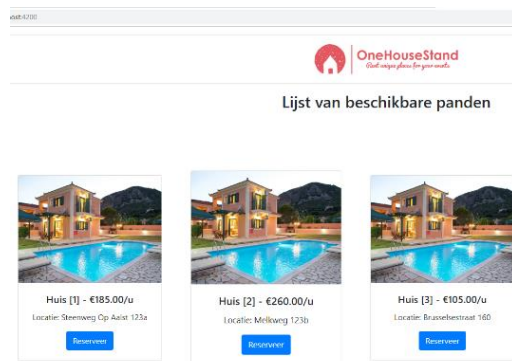
De applicatie bevat 1 hoofdcomponent, namelijk app.component.html. De content van deze “app.component” bevat de pagina die weergegeven wordt. Dit is “home.component” op de hoofdpagina, en “reserveer.component” als er naar de reserveer pagina genavigeerd wordt.

Vermits we 2 pagina’s nodig hadden, werd er ook gebruik gemaakt van een routing library. De routes werden meegegeven binnen “app.module.ts”.

```
const appRoutes: Routes = [  
  { path: '', component: HomeComponent },  
  { path: 'reserveer/:id', component: ReserveerComponent },  
];
```

Vermits de reserveer pagina moet weten welk pand er gereserveerd moet worden, wordt het id hiervan meegegeven bij de navigatie. Dit kan je zien bij “reserveer/:id”.

## Hoofdpagina:



Hier zie je de lijst van goedgekeurde panden, die beschikbaar zijn voor verhuur.

```
<section class="houses">
  <div *ngFor="let house of houses; let i = index" class="card">
    
    <div class="card-body text-center">
      <h5 class="card-title">Huis [{{house.id}}] - {{house.prijsPerUur | currency : '€'}}/u</h5>
      <p class="card-text">Locatie: {{house.streetName}} | {{house.huisnummer}}/d99</p>
      <a [routerLink]="['/reserveer', house.id]" routerLinkActive="active" class="btn btn-primary">Reserveer</a>
    </div>
  </div>
</section>
```

RouterLink toont aan hoe er genavigeerd wordt naar de reserveer pagina, en de id van het pand meegeeft.

Dit is de code achter de home pagina:

We kunnen zien dat er bij het laden een http request via axios wordt gedaan naar onze server (via loadHouses). Hier halen we onze goedgekeurde panden op. Deze worden in de variabele houses gezet, en in de html pagina wordt elk element dat in deze list zit getoond.

```
app.component.html TS app.module.ts reserveer.component.html TS home.component.ts
1 import { Component } from '@angular/core';
2 import axios from 'axios';
3
4 @Component({
5   selector: 'app-root',
6   templateUrl: './home.component.html',
7   styleUrls: ['./home.component.css']
8 })
9 export class HomeComponent {
10   houses = []
11   error = ""
12
13   constructor() {
14     this.loadHouses();
15   }
16
17   loadHouses() {
18     const that = this
19     axios.get('http://localhost:8080/api/panden')
20       .then(function (response) {
21         that.houses = response.data;
22       })
23       .catch(function (error) {
24         that.error = error
25       });
26   }
27 }
28
29
```

Als je op deze knop klikt zul je zien dat je naar een nieuwe pagina gaat, waar je een reservatie kan plaatsen. Dit formulier heeft standaard validatie.

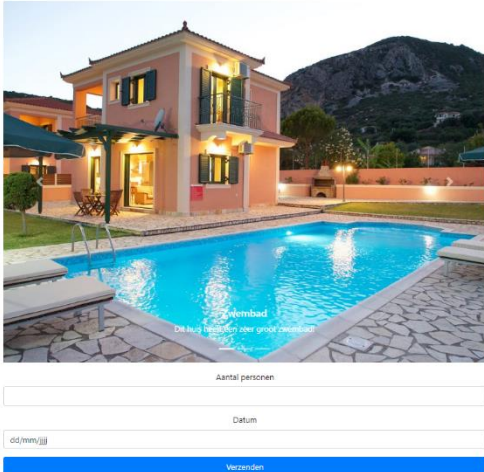
## Reservatie pagina

Als je de datum en het aantal personen invult, wordt de api aangesproken en wordt een pand gereserveerd.

Wanneer de reservatie component geladen wordt, wordt `ngOnInit()` opgeroepen. Deze methode wordt gebruikt om de id van het pand op te halen.

```
constructor(private _Activatedroute: ActivatedRoute) {
}
ngOnInit() {
  this.pandId = this._Activatedroute.snapshot.params['id'];
  console.log(this.pandId);
}

public open() {
  if (this.aantalPersonen > 1 && this.datum != null) {
    var error = false;
    axios.post(`http://localhost:8080/api/reserveer?pandId=${this.pandId}`, {
      "id": 0,
      "aantalPersonen": this.aantalPersonen,
      "eindDatum": this.datum
    })
    .then(function (response) {
      error = false;
      console.log(response);
    })
    .catch(function (error) {
      error = true;
    });
    if (error) {
      this.showErrorMsg = true;
      this.showSuccessMsg = false;
    }
    else {
      this.showErrorMsg = false;
      this.showSuccessMsg = true;
    }
  }
}
```



OneHouseStand  
Start your place for your needs

Maak een reservatie

Vermbad  
Deel het met vrienden of groep vrienden

Aantal personen

Datum

dd/mm/jj

Verstuur

Home

## 3 Teamreflectie

De opdracht begon tijdens de vakantie. Dit team had het geluk van 4 goede werkers te hebben, waardoor er na de vakantie weinig achterstand was. Gedeeltelijk wel door Emmanuel die moeite had met het combineren van school en werk.

Hierdoor waren we nog met 4 om dit project te vervullen. Dit bleek een positieve resultaat te geven omdat er nu meer zekerheid was dat wanneer een teamlid een taak kreeg, hij die taak volledig zou afwerken.

Af en toe waren er studenten die wat moeite hadden met een bepaald onderwerp, hier had het team eigenlijk wat beter op kunnen inspelen en vroeger reageren. Deze student werkte namelijk erg veel.

Ook al kreeg iemand een onderwerp niet zo snel onder knie, doordat we een team zijn van 4 goed werkende leden, is alles goed en wel afgeraakt.

Communicatie verliep erg vlot, bij afspraken/vragen werd er snel gereageerd. Alle studenten waren bijna op elke afspraak aanwezig.

## 4 Persoonlijke reflectie

### 4.1 Fabian

Tijdens de 2 opdrachten heb ik mijn best gedaan om de daily stand ups zo regelmatig mogelijk te houden. Ik had hierbij het geluk dat alle teamleden erg actief waren, waardoor dit niet zo veel werk vereiste. Verder hield ik me bezig met organisatie; wanneer moet wat af zijn, kijken of iedereen op schema zit en indien niet waarom dat het geval was,...

Mijn eigen taken heb ik goed volbracht. Af en toe had ik wat hulp nodig bij een bepaalde bug, hiervoor kon meestal bij mijn teamleden terecht.

Wat ik beter kon doen:

- Tijdens de opdrachten keek ik na waar iedereen was en probeerde ik ervoor te zorgen dat iedereen zijn taken volbracht. Dit is oké, maar voor in de toekomst zou ik me naast "Waarom heeft hij x niet gedaan?", ook moeten bezig houden met "Hoe ik de persoon kan helpen zijn taken te voltooien?".
- Daarnaast ben ik eerlijk en direct, maar wanneer iets niet goed is zeg ik dit ook onmiddellijk zonder tact. Dit zou ik moeten leren op een rustigere en minder confronterende manier te vertellen. Zo heeft de andere persoon er ook meer aan.

### 4.2 Bram

Ik vind dat deze opdracht bijzonder goed gelopen is. Persoonlijk kan ik zeggen dat mijn taken op één keer na altijd op tijd afgewerkt waren. Hierdoor moest ik niet op het laatste moment nog een oplossing bedenken. Wel jammer dat dit soms bij anderen en heel stuk minder vlot ging. Ook best jammer dat Emmanuel onze groep heeft verlaten. Al heeft hij nooit echt veel gewerkt, dus zo'n grote aanpassing was dit nu ook weer niet.

De sprints gingen vlot, we hebben ons best gedaan om steeds onze stand-ups en retrospectives te houden. Naar het einde toe

### 4.3 Luis

Persoonlijk verliep de 2 opdrachten op en neer voor mij. Ik was niet altijd mee met alle leerstof, en hierdoor werkte ik trager dan mijn teamleden. Zeker in het begin was alles Chinees voor mij, ik begreep echt niks van het programma, zelf het opstarten van het programma lukte niet. Dit was voor mij zeer frustrerend in het begin omdat mijn groep altijd goed werkte en ik voelde dat ik niet zo goed als hun werkte omdat ik steeds iets nieuws moest bijleren terwijl ik niet zo goed mee was met het vorig deel. Maar dit heeft me uiteindelijk goed geholpen omdat ik er enorm veel uit heb bijgeleerd.

Maar ik werkte wel altijd, ook al zat ik vast werkte ik altijd en deed ik altijd mijn best alles goed te begrijpen. Ook communiceerde ik altijd met mijn groep, was ik bijna altijd aanwezig op meetings, ik melde hun van waar ik stond en wat ik deed. Dit heeft geholpen omdat ik na een lange tijd goed mee was. Ook was ik beter mee omdat mijn teamleden geduldig met me waren en me de nodige uitlag gaven wanneer ik vast zat.

Uiteindelijk stond Application ook vaak in weg. Aangezien dit een prioriteit is voor mij, kon ik er af en toe er niet zijn met mijn groep.

## 4.4 Glenn

Ik vond mijn eigen prestaties tijdens deze opdracht nipt genoeg maar niet schitterend, vaak schreef ik code die in conflict kwam met de visie van het project, waardoor andere leden iets van mijn werk moesten aanpassen om het geheel werkende te krijgen.

Ook was ik traag met mijn deel van het werk te leveren, vooral in de laatste 2 weken, dit kwam door een enorme werklast van zowel AD4 als DD4 en SE4 tesamen, ik moest in stukken en brokken aan beide werken maar merkte dat dit niet zou lukken, dus besloot ik mijn werk per dag op te splitsen, 1 dag Ad, 1 dag SE, etc.

Het resultaat hiervan was dus dat mijn werk vaak trager geleverd werd.

Mijn teamgenote vind ik elk stuk voor stuk goeie werkers, ik spreek hierbij wel enkel van de teamgenoten die nog aanwezig zijn ten tijde van deze deadline, uitgesloten is dus Emanuel die eigenlijk niet veel gedaan heeft.