

# Studentenhuisvesting

Probleemanalyse

**Auteur:** Groep 14

**Datum:** November 2025

# Inhoudsopgave

<b>1</b>	<b>Projectbeschrijving</b>	<b>1</b>
<b>2</b>	<b>Contextanalyse</b>	<b>2</b>
2.1	Overzicht van uitdagingen . . . . .	2
2.1.1	Operationele problemen . . . . .	2
2.1.2	Nieuwe mogelijkheden voor de toekomst . . . . .	2
2.1.3	Kostenreductie en inzicht . . . . .	3
<b>3</b>	<b>Probleemanalyse</b>	<b>4</b>
3.1	Scope . . . . .	4
3.2	Doelgroepen . . . . .	4
3.2.1	Studenten . . . . .	4
3.2.2	Beheerders . . . . .	5
3.2.3	Onderhoudsmedewerker . . . . .	5
3.2.4	Onderhoudsmanager . . . . .	5
3.3	Beperkingen en voorkeuren . . . . .	5
<b>4</b>	<b>Functieanalyse (MoSCoW)</b>	<b>6</b>
4.1	Must-have . . . . .	6
4.2	Should-have . . . . .	6
4.3	Could-have . . . . .	6
4.4	Won't-have . . . . .	6
<b>5</b>	<b>Use Case – Pixar Pitch</b>	<b>7</b>
<b>6</b>	<b>Projectplanning</b>	<b>8</b>
<b>7</b>	<b>Agile-setup en teamafspraken</b>	<b>9</b>
7.1	Definition of Done . . . . .	9
7.2	Definition of Ready . . . . .	9
7.3	Tools . . . . .	9
7.4	Samenwerkingsafspraken . . . . .	9
7.4.1	Scrumrollen . . . . .	9

7.4.2	Communicatie . . . . .	10
7.4.3	Documentatie . . . . .	10
7.4.4	Oplevermomenten . . . . .	10
7.4.5	Besluitvorming . . . . .	10
<b>8</b>	<b>Initiële Product Backlog (MVP)</b>	<b>11</b>
8.1	User Stories (INVEST) . . . . .	11
8.2	Toelichting backlog . . . . .	13
<b>9</b>	<b>Conclusie</b>	<b>14</b>

# 1. Projectbeschrijving

**Projectnaam:** Studentenhuisvesting

**Doel:** Ontwikkelen van een IT-systeem dat de belangrijkste operationele problemen van Student Housing B.V. oplost en schaalbaar is voor toekomstige uitbreidingen.

**Hoofdvraag:** Hoe kan een nieuw IT-systeem het klachtenproces, onderhoudsbeheer en huishoudelijke organisatie binnen studentenhuisvesting efficiënter en inzichtelijker maken?

## **2. Contextanalyse**

Student Housing B.V. beheert meerdere studentencomplexen met gedeelde voorzieningen. Het huidige systeem is verouderd en ondersteunt de processen onvoldoende. Het nieuwe systeem moet:

- het klachten- en onderhoudsproces optimaliseren;
- ondersteuning bieden voor huishoudelijke organisatie;
- inzicht geven in kosten en leefomstandigheden;
- starten met een haalbare MVP;
- werken met CSV-data afkomstig uit het oude systeem.

### **2.1 Overzicht van uitdagingen**

#### **2.1.1 Operationele problemen**

- Onvoldoende naleving van schoonmaaktaken
- Onbetaalde gezamenlijke boodschappen
- Afval wordt niet tijdig afgevoerd
- Ongemelde feestjes of overlast
- Klachten over verwarming/airconditioning
- Storingen of slecht onderhoud van brandalarmen
- Geen traceerbaarheid van onderhoudsdiensten

#### **2.1.2 Nieuwe mogelijkheden voor de toekomst**

- Delen van studiehulpmiddelen
- Boodschappen- en voorraadbeheer

- Monitoring van luchtkwaliteit
- Smart-home integraties
- Communityfunctionaliteiten

### **2.1.3 Kostenreductie en inzicht**

- Inzicht in energieverbruik
- Betere onderhoudsplanning
- Inzicht in leefomstandigheden
- Betere studentplaatsing
- Minder schadegevallen

## **3. Probleemanalyse**

### **3.1 Scope**

#### **Binnen scope**

- Webapplicatie met rolgebaseerde toegang
- Klachtenregistratie
- Huishoudtakken en rooster
- Onderhoudsbeheer
- Eenmalige CSV-import
- Basisdashboard
- Veilig modelnetwerk

#### **Buiten scope**

- Smart-home integraties
- Communityfunctionaliteiten
- Externe integraties
- Veranderprocessen

### **3.2 Doelgroepen**

#### **3.2.1 Studenten**

Studenten wonen in de studentenhuizen die Student Housing B.V. aanbiedt. Zij hebben behoefte aan een overzichtelijk systeem waarmee zij klachten kunnen melden, de status van meldingen kunnen volgen en inzicht hebben in de afhandeling. Daarnaast willen zij duidelijkheid over huishoudtakken: wie welke taak uitvoert, wanneer dit moet gebeuren en wat al is gedaan.

### **3.2.2 Beheerders**

Beheerders zijn verantwoordelijk voor het dagelijkse reilen en zeilen binnen de studentenhuizen. Zij moeten inzicht hebben in alle openstaande en afgeronde klachten, de planning en uitvoering van huishoudtakken, de onderhoudsstatus en operationele kosten. Het systeem moet hen ondersteunen bij coördinatie, besluitvorming en communicatie met studenten en onderhoudspartijen.

### **3.2.3 Onderhoudsmedewerker**

Onderhoudsmedewerkers, zowel intern als extern, voeren reparaties en onderhoudstaken uit binnen de studentenhuizen. Zij hebben behoefte aan een duidelijke en actuele takenlijst, met per taak de locatie, omschrijving, prioriteit, benodigde materialen en eventuele veiligheidsinstructies. Het systeem moet hen helpen om werkzaamheden efficiënt en traanbaar uit te voeren.

### **3.2.4 Onderhoudsmanager**

De onderhoudsmanager is verantwoordelijk voor de planning, prioritering en toewijzing van onderhoudstaken. Deze rol vereist inzicht in de totale onderhoudsbehoefte, beschikbare capaciteit, kostenramingen en materiaalgebruik. De onderhoudsmanager moet eenvoudig taken kunnen aanmaken, prioriteiten kunnen wijzigen, voortgang kunnen monitoren en rapportages kunnen genereren.

## **3.3 Beperkingen en voorkeuren**

- CSV-data van onbekende kwaliteit
- Beperkte programmeerervaring team
- Hardware pas ná MVP
- Voorkeur voor heldere ICT-infrastructuur

## 4. Functieanalyse (MoSCoW)

### 4.1 Must-have

- Rolgebaseerde webapplicatie
- Klachtenregistratie
- Huishoudrooster
- Onderhoudsbeheer
- CSV-import
- Dashboard

### 4.2 Should-have

- Notificaties
- Prioriteiten in onderhoud
- Gebruikersprofielen

### 4.3 Could-have

- Documentupload
- Kalender
- Extra dashboards

### 4.4 Won't-have

- Smart-home koppelingen
- Communityfuncties

## **5. Use Case – Pixar Pitch**

Er was eens een studentenhuis dat werd beheerd door Student Housing B.V. Iedere dag worstelden studenten met klachten en huishoudtaken. Op een dag besloot Student Housing B.V. een webportaal te ontwikkelen. Daardoor konden studenten eenvoudig klachten indienen en taken volgen. Tot op een dag waren de problemen opgelost en was het woon-comfort verbeterd.

## 6. Projectplanning

- 10 november — Start groepsproject
- 24 november — Sprint 0 opleveren
- 7 december — Portfolioreview 3
- 8 december — Sprint 1 opleveren
- 22 december — Sprint 2 opleveren
- 5 januari — Sprint 3 opleveren
- 12 januari — Conclusie schrijven
- 18 januari — Portfolioreview 4

## **7. Agile-setup en teamafspraken**

### **7.1 Definition of Done**

- Functionaliteit werkt volledig
- Code getest
- Documentatie bijgewerkt
- Reviewer akkoord
- Gereed voor sprintreview

### **7.2 Definition of Ready**

- INVEST-principe toegepast
- Acceptatiecriteria testbaar
- UI/databasediscussies gevoerd
- Dependencies duidelijk

### **7.3 Tools**

Zie de Technische analyse.

### **7.4 Samenwerkingsafspraken**

#### **7.4.1 Scrumrollen**

- Per sprint wordt één Scrum Master aangewezen.
- De groep vertegenwoordigt samen de Product Owner.

#### **7.4.2 Communicatie**

- WhatsApp: korte berichten
- Trello: sprintplanning
- Microsoft Teams: vaste weekly op donderdag 17:30
- GitHub: code en documentatie

#### **7.4.3 Documentatie**

- Actiepunten worden genoteerd in Trello
- Notulist rouleert
- Documentatie volgt: **Analyse – Adviseer – Design – Realisatie – Manage & Control**

#### **7.4.4 Oplevermomenten**

- Elke sprint: review + retrospectief

#### **7.4.5 Besluitvorming**

- Consensus waar mogelijk
- Anders stemmen
- Bij gelijk: muntje opgooien
- Alleen de aanwezigen hebben stemrecht.

## 8. Initiële Product Backlog (MVP)

### 8.1 User Stories (INVEST)

#### US-01a – Toegang tot portaal

**Als** gebruiker

**wil ik** kunnen inloggen op een online portaal

**zodat** ik toegang heb tot alle functionaliteiten van het studentenhuis.

**Acceptatiecriteria:**

- Inloggen met rol (Student, Beheerder, Onderhoudsmedewerker, Onderhoudsmanager)
- Onjuist wachtwoord geeft foutmelding
- Rolgebaseerde toegang: gebruiker ziet alleen wat relevant is

**MoSCoW:** Must-have

#### US-01b – Navigatie en dashboard

**Als** ingelogde gebruiker

**wil ik** een overzichtelijk dashboard

**zodat** ik snel kan zien welke taken, klachten of meldingen relevant zijn voor mij.

**Acceptatiecriteria:**

- Dashboard toont relevante modules per rol
- Klikken op een module opent de juiste pagina
- Overzichtelijk en eenvoudig te begrijpen

**MoSCoW:** Must-have

## **US-02a – Klacht indienen**

**Als** student

**wil ik** een klacht kunnen indienen via een formulier

**zodat** ik problemen snel kan melden.

**Acceptatiecriteria:**

- Formulier bevat categorieën van veelvoorkomende problemen
- Mogelijkheid tot extra toelichting
- Bevestiging na indienen

**MoSCoW:** Must-have

## **US-02b – Klachtstatus volgen**

**Als** student

**wil ik** de status van mijn ingediende klacht kunnen volgen

**zodat** ik transparantie heb over de afhandeling.

**Acceptatiecriteria:**

- Status zichtbaar: Open, In behandeling, Afgerond
- Toegewezen medewerker zichtbaar
- Historie van acties zichtbaar

**MoSCoW:** Must-have

## **US-03 – Klachten beheren**

**Als** beheerder

**wil ik** klachten kunnen inzien, toewijzen, prioriteren en feedback geven

**zodat** klachten efficiënt worden opgevolgd.

**Acceptatiecriteria:**

- Lijst van openstaande klachten beschikbaar
- Toewijzen aan onderhoudsmanager of medewerker mogelijk
- Prioriteit instellen (Laag, Midden, Hoog)
- Reacties toevoegen die student kan zien
- Status aanpassen

**MoSCoW:** Must-have

## **US-04 – Huishoudrooster beheren**

**Als** beheerder

**wil ik** huishoudtakken kunnen aanmaken, verdelen en aanpassen

**zodat** het schoonmaakrooster eerlijk wordt bijgehouden.

**Acceptatiecriteria:**

- Taken aanmaken met omschrijving, locatie, frequentie, student toegewezen
- Taken verdelen handmatig of automatisch
- Aanpassen of verwijderen van taken
- Studenten zien hun taken in rooster

**MoSCoW:** Must-have

## **US-05 – Huishoudtakken afvinken**

**Als** student

**wil ik** mijn toegewezen huishoudtakken kunnen afvinken

**zodat** iedereen kan zien wat is gedaan.

**Acceptatiecriteria:**

- Taak markeren als uitgevoerd
- Status real-time zichtbaar voor alle gebruikers
- Historie beschikbaar voor beheerder

**MoSCoW:** Must-have

## **8.2 Toelichting backlog**

- Alle stories zijn **INVEST-compliant**: klein, testbaar, waardevol, inschatbaar en onafhankelijk waar mogelijk.
- De **MVP-focus** ligt op de absolute basisfunctionaliteiten, zonder smart-home integraties, communityfunctionaliteiten of externe koppelingen.
- Stories kunnen later worden uitgebreid naar **Should-have** en **Could-have** functionaliteiten zoals notificaties, documentupload, uitgebreide dashboards en kalenderweergave.

## 9. Conclusie

Het project *Studentenhuisvesting* richt zich op het ontwikkelen van een schaalbare, webgebaseerde IT-oplossing die de belangrijkste operationele knelpunten binnen studentencomplexen van Student Housing B.V. oplost. De analyse laat zien dat vooral het klachtenproces, het onderhoudsbeheer en de huishoudelijke organisatie momenteel inefficiënt verlopen door het gebruik van verouderde systemen en beperkte traceerbaarheid.

Met de gekozen scope, duidelijke rolverdeling en een MVP die zich focust op de essentiële functionaliteiten, klachtenregistratie, huishoudtakken en onderhoudsbeheer, wordt een haalbaar en waardevol fundament gelegd. Door gebruik te maken van user stories die voldoen aan het INVEST-principe, een heldere Definition of Ready en Definition of Done, en een Agile-werkwijze met vaste sprintmomenten, ontstaat een structuur die zowel de kwaliteit van het eindproduct als de samenwerking binnen het team ondersteunt.

Het project creëert niet alleen een oplossing voor de huidige problemen, maar legt ook een basis voor toekomstige uitbreidingen. Daarmee vormt de MVP een solide eerste stap richting een modern, gebruiksvriendelijk en betrouwbaar IT-systeem dat de leefomgeving van studenten verbetert en het beheer voor Student Housing B.V. aanzienlijk efficiënter maakt.

# Bibliografie

- [1] OpenAI. (2025, 9 november). *Hoe ziet een sprintplanning eruit?* ChatGPT. Geraadpleegd op <https://chat.openai.com>
- [2] OpenAI. (2025, 10–24 november). *Prompt gebruikt voor taalkundige verbetering.* ChatGPT. Geraadpleegd op <https://chat.openai.com>
- [3] Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide.* Geraadpleegd op <https://scrumguides.org>
- [4] Visual Paradigm. (z.d.). *Definition of ready in Scrum.* Geraadpleegd op <https://www.visual-paradigm.com/scrum/what-is-definition-of-ready-in-scrum/>
- [5] Visual Paradigm. (z.d.). *Definition of done vs acceptance criteria.* Geraadpleegd op <https://www.visual-paradigm.com/scrum/definition-of-done-vs-acceptance-criteria/>
- [6] Visual Paradigm. (z.d.). *Write user story SMART goals.* Geraadpleegd op <https://www.visual-paradigm.com/scrum/write-user-story-smart-goals/>