2016

Alex Abdalla, Michael Drent, Thijs van den Boogaard, Joyce Lam, Jonathan Ratnavel, Bryan Fernand

Sef01

1-1-2016

Systeemdocumentatie verslag

****

# Voorwoord

Dit verslag is onderdeel van project PAD. Bij het maken van dit verslag zijn de taken zo eerlijk mogelijk verdeeld over de projectleden. Ik wil graag de projectleden bedanken voor het nauw samenwerken waardoor dit verslag tot stand is gekomen. Om zo goed mogelijk geïnformeerd te worden over het product raden wij sterk aan eerst het volledige verslag te lezen en daarna de samenvatting een keer door te nemen. Ook willen wij graag Eric Nieuwenhuis bedanken voor zijn coaching tijdens het project.

Inhoud

[Voorwoord 1](#_Toc453154474)

[Inleiding 3](#_Toc453154475)

[1. Achtergrond opdrachtgever 4](#_Toc453154476)

[2. De applicatie killer 4](#_Toc453154477)

[3. Requirements 5](#_Toc453154478)

[4. Use-cases 6](#_Toc453154479)

[4.1 Use-case scenarios 6](#_Toc453154480)

[4.2 Use-casediagram 8](#_Toc453154481)

[5. Systeemoverzicht 9](#_Toc453154482)

[5.2 Gebruikte pakketten 9](#_Toc453154483)

# Inleiding

In het systeemdocumentatie verslag wordt het product dat tot stand komt tijdens het project volledig beschreven. Er wordt begonnen met een korte uitleg over wie onze opdrachtgever SEFlab precies is, een beschrijving gegeven van het product dat wordt gecreëerd tijdens het project, worden alle requirements uitgebreid beschreven met de MosCoW methode , worden de use cases samen met een use case diagramweegteven, de klassendiagrammen weergeven en worden ten slot de gebruikte pakketten weergeven.

# Achtergrond opdrachtgever

SEFlab is een instelling die zich inzet voor een lager energieverbruik voor software. Dit doet zij door hardware te bestuderen en te kijken hoe de software op de hardware energiezuiniger kan worden geprogrammeerd. SEFlab realiseert dit soort projecten door nauw samen te werken met software specialisten van de Software Improvement Group (SIG), Elektrische ingenieurs van de HVA en vele andere partners. De huidige doelen van het SEFLab zijn:

* Een model dat datacenters en IT afdelingen helpt met het voorspellen en verminderen van hun energie serverconsumptie.
* Het maken van richtlijnen en instrumenten voor software developers om te helpen met groene coderingsprincipes.
* Het maken van een energielabel voor software applicaties om eindgebruikers te helpen het kiezen van aankopen op basis van energie-efficiëntie

# De applicatie killer

Voor het project moet een applicatie worden gemaakt die zuiniger omgaat met energie dat de huidige software. De applicatie die wij voor het project realiseren is een app genaamd “Appkiller”. De Appkiller zorgt ervoor dat de gebruiker applicaties kan uitzetten zodat de computer minder stroom verbruikt. De gebruiker kan dit doen door de standaard configuratie te gebruiken of door een zelf aangemaakte configuratie te maken. Doordat je applicaties “killt” zal de computer minder stroom verbruiken.

# Requirements

Voordat er wordt begonnen met het maken van een applicatie moeten eerst alle requirements vastgesteld worden. Hierdoor weet het projectteam zeker dat alle functionaliteiten aanwezig zijn in de applicatie en weet het projectteam wat voor functies er geprogrammeerd moeten worden. De requirements zijn onderverdeelt in must haves, should haves, could haves en would haves (MoSCoW). Hieronder wordt er gegevens welke requirements er zijn vastgesteld en in welke categorie van de ‘haves’ ze horen.

|  |  |
| --- | --- |
| Requirements | MoSCoW |
| Een knop waarmee je processen killt. | Must have |
| Een standaard configuratie die processen killt. | Must have |
| Gebruiker kan zelf een configuratie maken. | Must have |
| Gebruiker kan lijst die alle processen weergeeft zien. | Must have |
| Gebruiker moet processen kunnen unkillen. | Must have |
| Een database voor de configuraties. | Must have |
| Verwijderen van processen op de unkill pagina. | Must have |
| Sorteren op mb in de processenlijst | Should have |
| Gebruiker kan zien hoeveel energie er wordt bespaard in een bepaalde tijd | Could have |

1. Use-cases

*In dit hoofdstuk worden de requirements uitgewerkt tot een use case. De use cases helpen de programmeurs een beter beeld te geven van hoe de applicatie geprogrammeerd moet worden. Door de use cases weet de programmeur beter hoe het systeem gebruikt gaat worden en kan er beter rekening worden gehouden met de eindgebruiker.*

## 4.1 Use-case scenarios

**Use Case 1:** Gebruikt “Kill-knop”

**User story:** Als gebruiker wil ik de “Kill”-knop van de Proces Killer gebruiken om alle ongebruikte processen te stoppen.    
**Primaire actoren:** Gebruiker

**Preconditions:** Gebruiker heeft een computer/laptop en ApplicatieKiller gedownload

**Success garantie:** Ongebruikte processen worden verwijderd

**Main Succes Scenario:**  
 1. Gebruiker start computer op

2. Gebruiker klikt op shortcut ApplicatieKiller

3. Gebruiker klikt op knop “Kill”  
 4. Systeem stopt ongebruikte processen  
 5. Gebruiker begint met huiswerk/gamen/surfen op het web etc..  
  
**Extensions:**

1a. Computer start niet op.

2a. Shortcut werkt niet, kan de directory van de Appkiller niet vinden.

4a. Systeem herkend de processen niet.

**Use Case 2:** Kill geselecteerde processen

**User story:** Als gebruiker wil ik de knop “processes” gebruiken om te zien welke processen worden gedraaid op de computer en ik die handmatig kan verwijderen  
**Primaire actoren:** Gebruiker

**Preconditions:** Gebruiker heeft een computer/laptop en ApplicatieKiller gedownload

**Success garantie:** Gebruiker bekijkt processen

**Main Succes Scenario:**  
 1. Gebruiker start computer op

2. Gebruiker klikt op shortcut ApplicatieKiller

3. Gebruiker klikt op knop “Processen”  
 4. Gebruiker ziet de processen

5. Gebruiker selecteert proces

6. Gebruiker klikt op ‘Kill’

**Extentions:**

1a. PC start niet op

2a. Shortcut werkt niet, kan de directory van de Appkiller niet vinden.

4a. Systeem herkend de processen niet.

6a. Gebruiker killt verkeerde proces.

**Use Case 3:** Gebruikt “Unkill processes”

**User story:** Als gebruiker wil ik de knop “Unkill processes”, om de processen die ik heb ‘gekilled’ te ‘unkillen’, dat wil zeggen weer in werking te stellen.  
**Primaire actoren:** Gebruiker

**Preconditions:** Gebruiker heeft een computer/laptop en ApplicatieKiller gedownload

**Success garantie:** Processen die zijn verwijderd worden weer in werking gesteld.

**Main Succes Scenario:**  
 1. Gebruiker start computer op

2. Gebruiker klikt op shortcut ApplicatieKiller

3. Gebruiker klikt op knop “Kill”  
 4. Systeem stopt ongebruikte processen  
 5. Gebruiker begint met huiswerk/gamen/surfen op het web etc..  
 6. Gebruiker klikt op knop “Unkill processes”  
 7. Gebruiker klikt op “Undo”

**Extensions:**  
1a. Computer start niet op.

2a. Shortcut werkt niet, kan de directory van de Appkiller niet vinden.

4a. Systeem herkend de processen niet.

7a. Systeem herstart een proces correct.

**Use case 4:** Gebruiken configuratie

**User story:** Als gebruiker wil ik een configuratie maken van de processen die ik wil killen zodat ik uit meerdere configuraties kan kiezen die goed aansluiten op de handelingen die ik met de computer verricht.

**Primaire actoren:** Gebruiker

**Preconditions:** Gebruiker heeft een computer/laptop en Applicatiekiller gedownload

**Succes garantie:** Gebruiker heeft een configuratie gemaakt van de applicaties die hij aan wil laten staan ( een lijst van alle applicaties die hij wil killen).

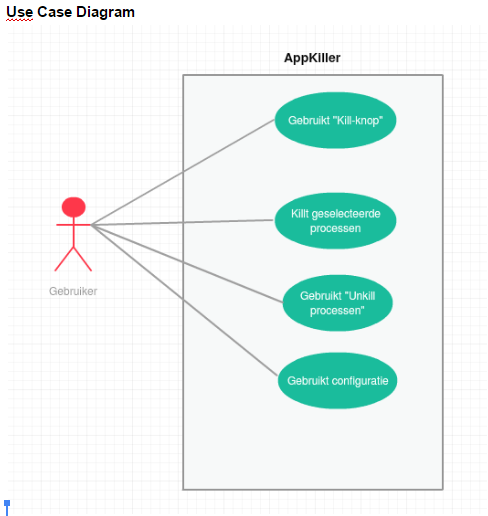
**Main Succes Scenario:**

1. Gebruiker start een computer op
2. Gebruiker klikt op shortcut applicatiekiller
3. Gebruiker klikt op ‘Kill’
4. Gebruiker klikt op ‘Unkill Processes’
5. Gebruiker geeft de configuratie een naam in het lege veld zodat hij hem later makkelijk terug kan vinden.
6. Gebruiker klikt op ‘in Database’
7. Systeem slaat lijst op in database
8. Gebruiker klikt op ‘Kill’
9. Gebruiker kiest zijn lijst
10. Gebruiker klikt op ‘Kill eigen lijst’
11. Systeem stopt de processen
12. Gebruiker gaat verder waarmee hij bezig was op de computer.

**Extensions:**

4.b: Gebruiker selecteert proces om te verwijderen uit de Unkill-lijst, zonder hem te unkillen.

## 4.2 Use-casediagram

****

# Systeemoverzicht

*In dit hoofdstuk worden alle hardware en software die tijdens dit project door het projectgroepje zijn gebruikt beschreven. Ook wordt er beschreven waar en hoe de gebruikte software en hardware is gebruikt.*

1. **Meetbordje SEFlab:**  hebben wij gebruikt om het stroomverbruik tussen de voeding en het moederbord mee te meten. Dit werd gedaan in TTH4 op de DELL computer die wij kregen van school.
2. **Dell pc school:** Werd gebruikt om de metingen op te verrichten aangezien het simpele pc’s zijn met geen extra’s en het werd aangeboden door school.
3. **USB 3.0:** gebruikt om Windows 10 op te zetten, zodat de DELL pc het Windows 10 besturingsprogramma had.
4. **Monitor School:** gebruikt om alle metingen op te bekijken.
5. **Laptops:** gebruikt om verslagen op te typen, de blog mee bij te houden, diagrammen op te maken, coderen en ontwerpen.
6. **Meetding Michael:**

## Gebruikte pakketten

1. **Netbeans:** Voor het programmeren in java.
2. **JDK:** om op Netbeans te kunnen programmeren.
3. **JavaFX:** voor het ontwerpen van de applicatie.
4. **Google docs:** Voor het maken van de documentatie.
5. **Meetsoftware:** Voor het meten van de processen in een computer.