ThijmenOS Systeem Test Plan

Opdrachtgever: Thijmen Brand

Versie: 0.1

Versiedatum: 30-10-2022

Status: Concept

# Inleiding

ThijmenOS is een virtueel operating systeem wat draait in de browser. Met dit project probeer ik meer te weten te komen van operating systemen, Maar ook een diepere kennis te ontwikkelen van typescript en javascript.

Het doel is om een compleet werkend operating system te schrijven met features zoals een kernel, applicaties, notificaties etc.

In dit document kunt u meer lezen over de het plan van aanpak met betrekking tot het testen van de applicatie. Zo zal ik nadenken over unit tests, maar ook over het testen van het grotere geheel.

Inhoud

[Inleiding 2](#_Toc124248704)

[2. Introductie 4](#_Toc124248705)

[2.1 Test onderdelen 4](#_Toc124248706)

[2.2 Test scope 4](#_Toc124248707)

[1.1 Scope 4](#_Toc124248708)

[1. Test strategie 5](#_Toc124248709)

[1.1 Scope 5](#_Toc124248710)

[1.2 Test type 5](#_Toc124248711)

[1.3 Test logistiek 5](#_Toc124248712)

[CI/CD toelichting 5](#_Toc124248713)

# Introductie

## 2.1 Test onderdelen

De systeem test dient te valideren dat alle Must have requirements van zowel de functionele- als non-functionele requirements correct werken op kleine unit schaal zowel als complete applicatie schaal.

Deze requirements zijn de volgende:

* Een gebruiker moet kunnen inloggen met zijn unieke gegevens
* Een gebruiker moet zijn persoonlijke gegevens kunnen veranderen
* Een gebruiker moet zijn inlog gegevens kunnen veranderen
* Een gebruiker moet mappen op zijn bureau blad kunnen zetten
* Een gebruiker moet applicatie shortcuts op zijn bureaublad kunnen zetten
* Een gebruiker moet bestanden op zijn bureaublad kunnen zetten
* Een gebruiker kan een applicatie openen
* Een applicatie kan notificaties sturen
* Een applicatie heeft permissies nodig om verschillende dingen te doen
* Een wachtwoord voor een account moet minstens 8 charters lang zijn, een Hoofdletter en een speciaal karakter bevatten
* Een applicatie moet enkel toegang hebben tot het os door een voor gedefinieerde set aan methoden

Wanneer al de bovenstaande requirements getest zijn biedt dit de zekerheid op een correct werkende basis operating system.

## 2.2 Test scope

De systeem tests zullen inlog-, bureaublad-, bepaalde instellingen-, permissie- en notificatie fuctionaliteit. Zowel als wachtwoord veiligheid en de applicatie – kernel communicaties omvatten.

Dit testplan is ontworpen om het ThijmenOS operating system te testen. In dit plan zal beschreven worden wat er getest zal worden aan het systeem. Daarnaast zal beschreven worden hoe, wanneer en met welke criteria deze verschillende tests plaats zullen vinden.

## 1.1 Scope

# 1. Test strategie

## 1.1 Scope

Een belangrijk onderdeel van JavascriptOS is de communicatie tussen een applicatie en de operating system kernel. Wanneer deze communicatie niet (correct) werkt zullen vele applicaties ook niet werken. Om zeker te weten dat dit kritische onderdeel werkt zal de gehele kernel getest worden.

Misschien nog wel belangijker is de backend api. Deze zorgt ervoor dat het operating system belangrijke informatie kan ophalen en weg schrijven zoals instellingen en überhaupt applicaties kan laden.

## 1.2 Test type

Om de kernel te testen zal gebruik gemaakt worden van unit tests. Omdat de kernel een component is zal iedere methode welke onderdeel is van de kernel als unit test getest worden.

De verschillende modulen binnen de backend api zullen getest worden doormiddel van unit tests. Deze modulen zijn kleine componenten en iedere methode kan individueel getest worden. Daarnaast zal de gehele api worden getest door middel van een api test.

## 1.3 Test logistiek

Unit tests zullen geschreven worden wanneer er een onderdeel word toegevoegd aan de kernel of wanneer en methoden / modulen worden toegevoegd aan de backend api. Wanneer deze nieuwe code vervolgens in een pull-request komt om in de development branch te komen worden de unit tests doormiddel van een CI pijpleiding worden gevalideerd.

De Api test zal worden gedaan wanneer deze door de CD pijpleiding naar productie word gepushed.

## CI/CD toelichting

In iedere ThijmenOS package heeft twee pipelines, een voor CI en een voor CD. De integratie pipeline zorgt er voor dat de code correct build, dat alle unit tests passeren en dat er geen lint errors zijn. Deze pipeline word getriggerd wanneer er een pull request naar de productie branch word geopend.

Naast de integratie pipeline is er ook een deployment pipeline. Deze checkt voor een laatste keer dat alles correct build en alle testen passeren en vervolgens deployed deze pipeline de code naar npm. Deze pipeline word getriggerd wanneer er een release gemaakt word op een package.