

**Productvarianten
Test Plan**

Versie 0.1

Productvarianten	Versie: 0.1
Test Plan	Datum: 27-5-2024

Inhoudsopgave

1.	Inleiding	3
1.1	Doel van dit document	3
1.2	Referenties	3
2.	Opdrachtformulering	4
2.1	Binnen scope	4
2.2	Buiten scope	4
2.3	Randvoorwaarden	4
3.	Testbasis	5
4.	Teststrategie	5
4.1	Samenvatting productrisico-analyse	5
4.2	Testsoorten	5
4.3	Verdeling van de testinspanning	7
5.	Testaanpak	7
5.1	Testtechnieken	7
5.2	Testproducten	7
5.3	Reviews	7
5.4	Testuitvoering	7
6.	Tooling en beheer	8
6.1	Testtools	8
6.2	Bevindingenprocedure	8

Bijlage I. Productrisico-analyse

Error! Bookmark not defined.

Productvarianten	Versie: 0.1
Test Plan	Datum: 27-5-2024

1. Inleiding

1.1 Doel van dit document

Dit Test Plan beschrijft de testaanpak die binnen het ontwikkelteam zal worden gehanteerd bij het testen van het productvarianten project. Het besteedt aandacht aan: testscope, testorganisatie, testtypen, testhulpmiddelen, testtooling en testomgeving.

1.2 Referenties

Titel	Versie	Auteur	Vindplaats
Requirements	V1	Jordy de Vries	GitLab > docker_and_docs
Acceptatie Plan	V1	Jordy de Vries	GitLab > docker_and_docs
Use Case Model	V1	Jordy de Vries	GitLab > docker_and_docs
SA-document	V1	Jordy de Vries	GitLab > docker_and_docs

Productvarianten	Versie: 0.1
Test Plan	Datum: 27-5-2024

2. Opdrachtformulering

De testinspanningen binnen het ontwikkelteam hebben de volgende doelen:

- Een bijdrage leveren aan de kwaliteit van de uiteindelijke feature.
- Meetbaar maken van de kwaliteit van feature.
- Beoordelen van de geschiktheid voor oplevering van feature.

2.1 Binnen scope

- Functionele tests
 - Creëren van producten/productvarianten
 - Verwijderen van producten/productvarianten
 - Weergave van varianten
 - Admin inlog functie
 - Varianten toevoegen aan een winkelwagen

2.2 Buiten scope

- End-to-end tests door derden
 - Externe systemen
 - Betaalproviders
- Productie-omgeving tests
- Security tests
- Load tests
- Non-functionele tests
 - UI/UX tests
 - Snelheid tests

2.3 Randvoorwaarden

- Planning
 - Deadline voor het inleveren: 28-05 om 11:59
 - Testplanning: 27-05 – 28-05
- Backend tests
 - Gemaakt met Junit
 - Unit tests
- Frontend tests
 - Gemaakt met Cypress
 - Integratie tests
 - E2E tests
- Acceptatie tests
 - Use Case document
- Server moet draaien
- Alle functionaliteit moet af zijn voordat begonnen kan worden aan de tests

Productvarianten	Versie: 0.1
Test Plan	Datum: 27-5-2024

3. Testbasis

De volgende werkproducten worden gebruikt als basis voor testgevallen:

- Acceptatie Plan
- Use Cases/requirements

4. Teststrategie

Er zullen een aantal dingen getest worden. Er zullen vooral veel functies getest worden die een directe link hebben met de requirements (zie requirements document). De functionaliteit daarvan wordt op verschillende wijze getest en het behalen van de tests wordt deels beoordeeld door het Use Case document (zie Use Case document). Aan de hand van de Use cases kan namelijk makkelijk bepaald worden of een bepaalde functionaliteit werkt zoals deze zou moeten werken.

Onder andere zal het volledige administrator paneel getest worden wat overigens ook gelijk het grootste deel van de requirements is. Onder het administrator paneel vallen producten toevoegen en verwijderen en inloggen. Verder zullen er ook nog een aantal gebruikers testen komen voor klanten functies. Denk hierbij aan producten toevoegen aan een winkelmandje en producten bestellen. De meeste test tijd zal op gaan in het testen van het administrator paneel. Dit omdat het administrator paneel simpelweg zeer veel complexe onderdelen bevat en de meeste functionaliteit bevat over het hele project.

4.1 Samenvatting productrisico-analyse

Kenmerk/ - onderdeel	Risicoklasse	Argumentatie
Use Case 3 t/m 8	Hoog	Alleen voor intern gebruik, risico bij falen is hoog wegens administratieve redenen
Use Case 2	Midden	Klant gebruik, risico bij falen is gemiddeld, er is een kans dat een order niet geregistreerd wordt.
Use Case 1	Laag	Klant gebruik, geen risico bij falen omdat het systeem niks zal registreren en/of opslaan

4.2 Testsoorten

Binnen het ontwikkelteam worden de volgende testsoorten toegepast.

Testsoort	Beschrijving	Doel
Bouwtest	Testsoort die door programmeurs wordt uitgevoerd. Hierbij wordt zowel gebruik gemaakt van Unit Tests als andere testtechnieken zoals integratietesten.	Verifiëren dat de code test gereed is en dat aangebrachte wijzigingen geen ongewenste neveneffecten hebben gehad.
Systeemtest	Testsoort die door testers wordt uitgevoerd. Aan de hand van Use Cases en acceptatiecriteria uit het Acceptatie Plan worden testgevallen uitgewerkt en uitgevoerd.	Metten en daarmee helpen verbeteren van de kwaliteit van code en andere werkproducten, verifiëren dat de gerealiseerde software voldoet aan de overeengekomen specificaties en beoordelen van de oplevergereedheid van nieuwe en eerder gerealiseerde functionaliteit.
GAT ondersteuning	Testsoort die door gebruikers en andere domeindeskundigen wordt uitgevoerd. Het testteam ondersteunt deze testinspanningen.	Beoordelen of de gerealiseerde functionaliteit voldoet aan de bedoeling die de domeindeskundige voor ogen stond bij het formuleren van de specificaties.

Productvarianten	Versie: 0.1
Test Plan	Datum: 27-5-2024

Productvarianten	Versie: 0.1
Test Plan	Datum: 27-5-2024

4.3 Verdeling van de testinspanning

Kenmerk/ - onderdeel	Risicoklasse	Bouwtest	Systeemtest	GAT ondersteunen
UC01	Laag	●	●	●
UC02	Midden	●	●	●
UC03	Hoog	●●	●●●	●
UC04	Hoog	●●●	●●	●
UC05	Hoog	●●●	●●	●
UC06	Hoog	●●●	●●	●
UC07	Hoog	●●●	●●	●
UC08	Hoog	●●	●	●

5. Testaanpak

Alle frontend testen zullen worden uitgevoerd met Cypress. Dit is een programma wat gebruikersgedrag kan simuleren op een webpagina. Voor de backend maken we gebruik van Junit, aangezien de backend geprogrammeerd is in Java en Junit.

5.1 Testtechnieken

Voor de frontend waar we dus gebruik maken Cypress. Deze Cypress testen zijn integratie testen. Voor de backend maken we gebruik van Junit en zullen we unit testen uitvoeren. Allebei de testen voor frontend en backend zijn bouw testen die dus door de developers zullen worden uitgevoerd.

5.2 Testproducten

Werkproduct	Toelichting
Test Plan	In het test plan wordt beschreven hoe de tests tot stand zullen komen, wat de aanpak zal zijn, wat de testtechnieken zullen zijn en hoe de planning eruit zal komen te zien.
Testontwerp	Een uitgebreide beschrijving van alle tests die uitgevoerd worden.
Testscript	De scripts die we programmeren in, in dit geval Junit en Cypress, om de webshop functies volledig automatisch te kunnen testen.
Laadscript	Een stresstest script om de performance van een website te testen onder zware omstandigheden
Bevinding	Een korte beschrijving van het resultaat van een test
Testrapport	Een uitgebreid rapport van alle tests die zijn uitgevoerd. Hierin is alles te vinden over de tests van de geteste feature

5.3 Reviews

Er zijn twee belangrijke reviews. De eerste is een review van de code om te kijken hoeveel code er getest wordt zodat er voldaan wordt aan de eis van 80% code coverage. De tweede review is een review van de tests zelf. Hierbij wordt er gekeken naar het ontwerp van de testen en of deze testen wel valide zijn en daadwerkelijk hetgeen testen waarvan wordt beschreven dat ze dat zouden testen

5.4 Testuitvoering

De tests zullen worden uitgevoerd door eerst per onderdeel de test uit te voeren en daarna voor elk onderdeel de test resultaten en bevindingen te noteren. De tests bestaan uit verschillende onderdelen waaronder acceptatietesten, unittesten en integratietesten en acceptatietesten. Daarna zal er een test rapport worden geschreven waarin alle tests met de bijbehorende testresultaten worden gedocumenteerd. Er wordt vervolgens nog een conclusie geschreven waarin kort wordt samengevat wat de testresultaten zijn.

Productvarianten	Versie: 0.1
Test Plan	Datum: 27-5-2024

6. Tooling en beheer

Voor het testen van de software is minimaal nodig:

- Een laptop of desktop
- IDE
- Microsoft Word, libre office of Google Docs voor het noteren van de testresultaten en bevindingen
- Voor frontend testen Cypress
- Voor backend testen Junit
- Voor Docker testen Docker desktop

Voor het operating systeem van de laptop wordt geadviseerd een Windows of MacOS machine te gebruiken in combinatie met IntelliJ voor de backend en VSCode of Webstorm voor de frontend.

6.1 Testtools

Voor unit testen wordt er gebruik gemaakt van Junit in Java

Voor integratie testen wordt er gebruik gemaakt van Cypress voor de frontend

Voor de acceptatie testen wordt de functie simpelweg uitgevoerd en gekeken of de functie doet wat de functie zou moeten doen volgens de acceptatie test

6.2 Bevindingenprocedure

Alle bevindingen komen in het testrapport te staan, dit zal een regulier Microsoft Word of Google Docs document zijn waar ook de conclusie zal komen te staan. Elke bevinding zal in een tabel komen te staan met

- Een naam
- Beschrijving van de test
- Verwachte resultaat
- Daadwerkelijk resultaat
- Prioriteit/Risico
- Screenshots/logs

Er komt ook een korte beschrijving hoe een gefaalde test eventueel zou kunnen worden opgelost. Om bij te houden welke tests uitgevoerd moeten worden, wordt gebruik gemaakt van een KanBan board.