**Systeemtestplan**

Project: BestelMar

Project team: Rik Peeters

Versie: 0.1

Versiedatum: 26 November 2019

Status: Concept

# Documenthistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Wijzigingen | Auteur | Datum |
| 0.1 | Opzet document | Nico Kuijpers | 11 jun 2018 |
| 1.0 | Hoofdstuk teststrategie toegevoegd | Rik Peeters | 26 nov 2019 |

Inhoud

Documenthistorie 2

H1 Inleiding 4

H2 Uitgangssituatie 5

H3 Productrisicoanalyse 6

H3 Logische testcases 7

H4 Fysieke testcases 8

H5 Testcoverage 9

H6 Unittesten en code coverage 10

H7 Statische code analyse 11

H8 Conclusie 12

# H1 Inleiding

De applicatie gaat dienen als een order systeem voor bijv. bioscopen en cafés. De applicatie moet het makkelijker maken voor bijv. in een bioscoop een drankje te bestellen. Wanneer een klant iets besteld zal dit live worden gedeeld met de medewerkers. Dit zorg voor een nog fijnere bestel ervaring en draagt bij aan een innovatieve manier van horeca.

Het doel van dit document is aan te geven wat we allemaal gaan testen en hoe we dit gaan doen. Ook vind je hoe groot het risico is van bepaalde stukken van de applicatie en hoe groot de kans is dat dit fout gaat. Hierop wordt gebaseerd hoe uitgebreid we een onderdeel gaan testen.

De stukken die we gaan testen halen we uit de requirements. Die vind je het document “AnalyseDocument”

# H2 Uitgangssituatie

De versie waarmee getest moet worden is een compleet functionele versie. Maar niet gelijk alle componenten moeten af zijn om te kunnen testen. De verschillende componenten kunnen los van elkaar getest worden. Waar gebruikt gemaakt wordt van de database of api moet er wel functionele test data in staan.

# H3 Productrisicoanalyse

In dit hoofdstuk wordt de product risicoanalyse beschreven. We hebben geprobeerd onze testen zo goed mogelijk te verdelen over de verschillende testdoelen.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testdoel | Schade | Faalkans | Risicoklasse | Ernstklasse |
| UC1: Tafel Registeren | 3 | 1 | 3 | C |
| UC2: Tafel Verwijderen | 1 | 1 | 1 | C |
| UC3: Tafel Updaten | 1 | 1 | 1 | C |
| UC4: Orders Inzien | 3 | 2 | 6 | A |
| UC5: Orders Sluiten | 2 | 2 | 4 | C |
| UC6: Order maken | 3 | 2 | 6 | A |
| UC7: Gebruiker Registreren | 2 | 2 | 2 | C |
| UC8: Gebruiker Inloggen | 2 | 2 | 2 | A |
| UC9: Gebruikers inzien | 3 | 2 | 6 | A |
| UC10: Gebruiker Wijzigen | 2 | 2 | 4 | B |
| UC11: Koppelen Tafel | 2 | 2 | 4 | B |
| UC12: Korting Gebruiker | 1 | 1 | 1 | C |
| UC13: Artikel Toevoegen | 3 | 1 | 3 | C |
| UC14: Artikel Verwijderen | 1 | 1 | 1 | C |
| UC15: Artikel Wijzigen | 1 | 1 | 1 | C |
| UC16: Gebruikers Rechten Veranderen | 2 | 3 | 6 | A |
| UC17: Bon Printen | 1 | 1 | 1 | C |

# H4 Teststrategie

Hier wordt op basis van de productrisicoanalyse de teststrategie bepaald en wordt de testaanpak beschreven.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testdoel | PRA | UT | CT | IT | ST handm. | ST autom. |
| Tafel Registeren | C | ● |  | ● |  |  |
| Tafel Verwijderen | C | ● |  | ● |  |  |
| Tafel Updaten | C | ● |  | ● |  |  |
| Orders Inzien | A | ●●● |  |  | ●● |  |
| Orders Sluiten | C | ● |  |  | ●● |  |
| Order maken | A | ●●● |  |  | ● |  |
| Gebruiker Registreren | C | ● |  | ● |  |  |
| Gebruiker Inloggen | A | ●●● | ● |  |  |  |
| Gebruikers inzien | A | ●●● | ● |  |  |  |
| Gebruiker Wijzigen | B | ●● |  | ● |  |  |
| Koppelen Tafel | B | ●● |  |  | ● |  |
| Korting Gebruiker | C | ● |  |  | ● |  |
| Artikel Toevoegen | C | ● |  | ● |  |  |
| Artikel Verwijderen | C | ● |  | ● |  |  |
| Artikel Wijzigen | C | ● |  | ● |  |  |
| Gebruikers Rechten Veranderen | A | ●●● | ● |  | ● |  |
| Bon Printen | C | ● |  | ● |  |  |
|  |  |  |  |  |  |  |

Toelichting bij bovenstaande tabel:

|  |  |
| --- | --- |
| PRA-RK | Risicoklasse uit de productrisicoanalyse (PRA): risicotabel |
| Toetsen | Toetsing/review van de verschillende tussenproducten zoals requirements, functioneel ontwerp, technisch ontwerp |
| Ontwikkeltest | Unittest en Unitintegratietest |
| ST | Systeemtest |
| IT | Integratie test |
| CT | Component test |
| UT | Unit Test |
| ● | beperkte dynamische test |
| ●● | gemiddelde dynamische test |
| ●●● | zware dynamische test |

# H5 Fysieke testcases

In dit hoofdstuk worden de logische testcases uitgewerkt tot fysieke testcases en wordt het verwachte resultaat bepaald. De verschillende resultaten worden met behulp van kleuren gerangschikt op succes. Hieronder staat de betekenis van de kleuren beschreven met daaronder de fysieke testcases.

|  |  |
| --- | --- |
|  | De test is geslaagd en er is niks op aan te merken. |
|  | De test lijkt goed te gaan, maar er is iets op het behaalde resultaat aan te merken. |
|  | De test is gefaald. |
|  | Deze test is niet van toepassing op de huidige versie van de applicatie en is ook niet uitgevoerd. |
|  | Dit is geen test maar een actie die ondernomen moet worden om daar op volgende tests uit te kunnen voeren (initialisatie actie). |

*Voorbeeld van fysieke testcases:*

Voor elke crud operatie zal er een happy en een unhappy flow worden gemaakt

Overige Speciale operaties worden getest

# H7 Testcoverage

Voor de test matrix verwijzen we naar de [test matrix spreadsheet](https://docs.google.com/spreadsheets/d/1Bh_RbLD_0VY0um1UNU7k75jPVPNL60BHKHPrhWIDdyk/edit?usp=sharing)

# H8 Unittesten en code coverage

Voor code coverage willen wij op zijn minst 80% van onze code getest hebben. Naast de standaard classes en de bijbehorende constructors, willen wij juist alle classes die een risico kunnen bevatten getest hebben. Van elke methode zal voor- en na de resultaten worden vergeleken met de verwachte resultaten.

# H9 Statische code analyse

*In dit hoofdstuk wordt beschreven hoe je statische code analyse gaat onderzoeken. Dit kan met behulp van SonarQube. Laat de resultaten van de analyse zien in het testrapport, bijvoorbeeld door een screenshot te maken. Beschrijf ook je conclusies op basis van de resultaten en geef aan op welke punten je mogelijkheden ziet om de code verder te verbeteren.*

# H10 Conclusie *(alleen van toepassing in testrapport)*

*Beschrijf hier de overall conclusie op basis van de resultaten van de uitvoering van de systeemtest, unittesten en statische code analyse. Geef ook aan of de applicatie kan worden opgeleverd, dat wil zeggen geef aan of de applicatie voldoet aan de requirements (functionele en niet-functionele eisen) zoals beschreven in het analyse document. Als dit niet zo is, geef dan aan wat er nog moet gebeuren voordat de applicatie opgeleverd kan worden.*