

Testplan

Live performance

Michelle Broens | S21M | 2017

Inhoudsopgave

[Inleiding 3](#_Toc472443777)

[Testmatrix 4](#_Toc472443778)

[Testcases 5](#_Toc472443779)

[Conclusie 6](#_Toc472443780)

# Inleiding

In dit document worden er tests geschreven voor de requirements uit het analysedocument. Dit is het acceptatierapport voor de applicatie. Dit document is belangrijk om te kijken of alle M requirements afgedekt zijn, dit is in één oogopslag te zien.

Voor elke requirement is er één testcase en voor elke testcase is er een stappenplan geformuleerd. Elke stap die hier aangegeven staat wordt ook uitgevoerd in de werkelijke applicatie. Dit zorgt ervoor dat de functionele correctheid van de applicatie goed vastgelegd kan worden. Voor elke testcase wordt er een score genoteerd:

1. FAILED: de functie is niet werkend of niet aanwezig.
2. PASSED: de functie werkt zoals er in de requirements genoteerd staat.

De acceptatietest keurt de applicatie goed als deze criteria afgetekend zijn:

1. De testgevallen die gerelateerd zijn aan de requirements met MoSCoW criteria M de status PASSED hebben.
2. De testgevallen die gerelateerd zijn aan de requirements met MoSCoW criteria S de status FAILED hebben.
3. De testgevallen die gerelateerd zijn aan de requirements met MoSCoW criteria C de status FAILED hebben.
4. De testgevallen die gerelateerd zijn aan de requirements met MoSCoW criteria W de status FAILED hebben.
5. Alle testcases zijn uitgevoerd.

De test cases worden uitgevoerd op een pc waar het Windows 10 systeem op draait. De applicatie wordt getest door het programma Visual Studio 2015 Community.

# Testmatrix

De onderstaande matrix geeft de dekking van de functional requirements door de testcases weer. De aangegeven requirements zijn terug te vinden in het analysedocument.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirement** | **T\_WR\_1** | **T\_WR\_2** | **T\_WR\_3** | **T\_WR\_4** | **T\_WR\_5** | **T\_WR\_6** | **T\_BR\_1** | **T\_BR\_2** |
| **WR1** | x |  |  |  |  |  |  |  |
| **WR2** |  | x |  |  |  |  |  |  |
| **WR3** |  |  | x |  |  |  |  |  |
| **WR4** |  |  |  | x |  |  |  |  |
| **WR5** |  |  |  |  | x |  |  |  |
| **WR6** |  |  |  |  |  | x |  |  |
| **BR1** |  |  |  |  |  |  | x |  |
| **BR2** |  |  |  |  |  |  |  | x |

# Testcases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Testcase**  **identificatie** | **Uit te voeren stappen** | **Verwacht resultaat** | **Geobserveerd**  **resultaat** | **Resultaat**  **testcase** |
| **T\_WR\_1** | Bij het inladen van de database worden de waarnemingen en bezoeken die al gedaan zijn bij de juiste kaart toegevoegd | Als de gebruiker een gebied bekijkt waar hij al eerder waarnemingen voor gedaan heeft kan hij nu daar in de applicatie heen gaan en staan zijn oude waarnemingen daar |  | PASSED |
| **T\_WR\_2** | Als de gebruiker op een plek op de kaart klikt krijgt hij het “Nieuwe waarneming scherm” te zien en kan hij daar de informatie invullen, deze informatie wordt dan toegevoegd aan de kaart | Als de gebruiker op een plek op de kaart klikt waar hij een nieuwe waarneming wil toevoegen, krijgt hij een scherm waar hij de informatie over de waarneming kan invullen, hij klikt dan op de “Opslaan” knop en zijn ingevulde gegevens worden toegevoegd aan de kaart |  | PASSED |
| **T\_ WR\_3** | Als de gebruiker op het “Kaartscherm” zit en klikt op een waarneming die er al staat krijgt hij het “Waarneming detail scherm” te zien | Als te gebruiker op het “Kaartscherm” een al aanwezige waarneming klikt krijgt hij hiervan de details te zien |  | PASSED |
| **T\_WR\_4** | De gebruiker kan op het “Waarnemingen per teller scherm” op de knop “Uitdraaien” klikken, er wordt dan een tekstbestand gemaakt van de waarnemingen die één teller heeft gedaan | De gebruiker zit op het “Waarnemingen per teller scherm” en ziet in een lijst alle waarnemingen die hij in dat gebeid gedaan heeft. Hij klikt dan op de knop “Uitdraaien” dan wordt er een tekstbestand opgeslagen met de waarnemingen op het scherm |  | PASSED |
| **T\_WR\_5** | In de database wordt er uitgerekend hoe veel broedparen er zijn per vogelsoort en dit wordt geupdate met de informatie van de waarnemingen die er gedaan worden | Als de gebruiker net een waarneming heeft gedaan die het aantal broedparen ophoogt kan hij dit zien in het “Waarneming detail scherm” |  | PASSED |
| **T\_WR\_6** | Als de gebruiker op het “Waarnemingen per teller scherm” zit en op de knop “Uitdraaien” klikt terwijl een van de waarneming in de lijst niet aan de afgesproken voorwaarden voldoet, komt er een pop-up scherm met de melding: “Een van de waarnemingen voldoet niet aan de afgesproken voorwaarden” | De gebruiker klikt op de “Uitdraaien” knop op het “Waarnemingen per teller scherm”, maar niet alle waarnemingen in de lijst voldoen aan de afgesproken voorwaarden dus komt er een pop-up scherm met de melding: “Een van de waarnemingen voldoet niet aan de afgesproken voorwaarden” |  | PASSED |
| **T\_BR\_1** | Tijdens het aanmaken van een nieuwe waarneming wordt de informatie die de gebruiker invoert naar een xml-bestand gestuurd en lokaal opgeslagen | Als de gebruiker een paar waarnemingen heeft gedaan en deze lokaal opgeslagen zijn kan hij naar het “Lokale waarnemingen scherm” gaan om te zien dat zijn waarnemingen ook daadwerkelijk opgeslagen zijn |  | PASSED |
| **T\_BR\_2** | Als de gebruiker een aantal lokale opgeslagen waarnemingen heeft gaat hij naar het “Lokale waarnemingen scherm” hier klikt hij dan op de knop “Opslaan” deze stuurt dan de lokale xml-bestanden naar de database | Als de gebruiker een aantal lokale waarnemingen heeft en deze ook lokaal opgeslagen zijn, kan hij zien dat deze na op de “Opslaan” knop te klikken weg zijn en ook daadwerkelijk op het “Kaartscherm” te zien zijn |  | PASSED |

# Conclusie

Van alle requirements die in het analysedocument staat zijn alle “M” (Must have) requirements …

De “S” requirements zijn …

Volgens het criteria kan de applicatie …