Ontwerpdocument

Groep 15

Inhoud

[Ontwerpdocument groep 15 6](#_Toc11088965)

[Sprint 1 6](#_Toc11088966)

[Sprint 2 6](#_Toc11088967)

[2.1. User story 1 - Als gast wil ik mijzelf kunnen registreren zodat ik later mee kan bieden en evt. kan veilen. 6](#_Toc11088968)

[2.1.1. Wireframe’s 6](#_Toc11088969)

[2.1.2. Functioneel ontwerp 7](#_Toc11088970)

[2.1.3. Technisch ontwerp 7](#_Toc11088971)

[2.1.4. Pseudo code 7](#_Toc11088972)

[2.1.5. Activity Diagram 8](#_Toc11088973)

[2.1.6. Testcases 8](#_Toc11088974)

[2.2. User story 2 - Als beheerder wil ik dat gebruiker accounts worden geverifieerd, zodat de kans op nepaccounts wordt verkleind. 9](#_Toc11088975)

[2.2.1. Wireframe’s 9](#_Toc11088976)

[2.2.2. Functioneel ontwerp 9](#_Toc11088977)

[2.2.3. Technisch ontwerp 9](#_Toc11088978)

[2.2.4. Pseudo code 10](#_Toc11088979)

[2.2.5. Activity Diagram 10](#_Toc11088980)

[2.2.6. Testcases 10](#_Toc11088981)

[2.3. User story 3 - Als gebruiker wil ik kunnen inloggen zodat ik meet interactie met de site kan hebben. 10](#_Toc11088982)

[2.3.1. Wireframe’s 10](#_Toc11088983)

[2.3.2. Functioneel ontwerp 10](#_Toc11088984)

[2.3.3. Technisch ontwerp 10](#_Toc11088985)

[2.3.4. Pseudo code 10](#_Toc11088986)

[2.3.5. Activity Diagram 10](#_Toc11088987)

[2.3.6. Testcases 10](#_Toc11088988)

[2.4. User story 4 - Als gebruiker wil ik op een voorwerp kunnen bieden zodat ik deze mogelijk kan aanschaffen. 11](#_Toc11088989)

[2.4.1. Wireframe’s 11](#_Toc11088990)

[2.4.2. Functioneel ontwerp 11](#_Toc11088991)

[2.4.3. Technisch ontwerp 11](#_Toc11088992)

[2.4.4. Pseudo code 11](#_Toc11088993)

[2.4.5. Activity Diagram 11](#_Toc11088994)

[2.4.6. Testcases 11](#_Toc11088995)

[2.5. User story 5 - Als gebruiker wil ik kunnen uitloggen zodat anderen niet met mijn account kunnen werken. 11](#_Toc11088996)

[2.5.1. Wireframe’s 11](#_Toc11088997)

[2.5.2. Functioneel ontwerp 11](#_Toc11088998)

[2.5.3. Technisch ontwerp 11](#_Toc11088999)

[2.5.4. Pseudo code 11](#_Toc11089000)

[2.5.5. Activity Diagram 11](#_Toc11089001)

[2.5.6. Testcases 11](#_Toc11089002)

[2.6. User story 6 - Als gebruiker wil ik een verkoper worden om daarna zelf voorwerpen aan te kunnen bieden. 11](#_Toc11089003)

[2.6.1. Wireframe’s 11](#_Toc11089004)

[2.6.2. Functioneel ontwerp 11](#_Toc11089005)

[2.6.3. Technisch ontwerp 11](#_Toc11089006)

[2.6.4. Pseudo code 11](#_Toc11089007)

[2.6.5. Activity Diagram 11](#_Toc11089008)

[2.6.6. Testcases 11](#_Toc11089009)

[2.7. User story 7 - Als verkoper wil ik zelf voorwerpen kunnen aanbieden zodat deze kunnen worden geveild. 12](#_Toc11089010)

[2.7.1. Wireframe’s 12](#_Toc11089011)

[2.7.2. Functioneel ontwerp 12](#_Toc11089012)

[2.7.3. Technisch ontwerp 12](#_Toc11089013)

[2.7.4. Pseudo code 12](#_Toc11089014)

[2.7.5. Activity Diagram 12](#_Toc11089015)

[2.7.6. Testcases 12](#_Toc11089016)

[2.8. User story 8 - Als verkoper wil ik op de hoogte gebracht worden wanneer mijn veiling beëindigd, zodat ik afspraken kan maken met een eventuele winnaar. 12](#_Toc11089017)

[2.8.1. Wireframe’s 12](#_Toc11089018)

[2.8.2. Functioneel ontwerp 12](#_Toc11089019)

[2.8.3. Technisch ontwerp 12](#_Toc11089020)

[2.8.4. Pseudo code 12](#_Toc11089021)

[2.8.5. Activity Diagram 12](#_Toc11089022)

[2.8.6. Testcases 12](#_Toc11089023)

[2.9. User story 9 - Als beheerder wil ik zien hoe de website op bedrijfsniveau presteert, zodat ik de strategie daarop kan aanpassen. 12](#_Toc11089024)

[2.9.1. Wireframe’s 12](#_Toc11089025)

[2.9.2. Functioneel ontwerp 12](#_Toc11089026)

[2.9.3. Technisch ontwerp 12](#_Toc11089027)

[2.9.4. Pseudo code 12](#_Toc11089028)

[2.9.5. Activity Diagram 12](#_Toc11089029)

[2.9.6. Testcases 12](#_Toc11089030)

[2.10. User story 10 - Als beheerder wil ik een overzichtelijke logging en presentatie daarvan zodat in de gaten kan houden of de site operationeel betrouwbaar is. 13](#_Toc11089031)

[2.10.1. Wireframe’s 13](#_Toc11089032)

[2.10.2. Functioneel ontwerp 13](#_Toc11089033)

[2.10.3. Technisch ontwerp 13](#_Toc11089034)

[2.10.4. Pseudo code 13](#_Toc11089035)

[2.10.5. Activity Diagram 13](#_Toc11089036)

[2.10.6. Testcases 13](#_Toc11089037)

[2.11. User story 11 - Als Product Owner wil ik de opgekochte voorwerpen batchgewijs naar de site converteren, zodat ik kan starten met een gevulde website. 13](#_Toc11089038)

[2.11.1. Wireframe’s 13](#_Toc11089039)

[2.11.2. Functioneel ontwerp 13](#_Toc11089040)

[2.11.3. Technisch ontwerp 13](#_Toc11089041)

[2.11.4. Pseudo code 13](#_Toc11089042)

[2.11.5. Activity Diagram 13](#_Toc11089043)

[2.11.6. Testcases 13](#_Toc11089044)

[Sprint 3 14](#_Toc11089045)

[3.1. User story 1 - Als beheerder wil ik gebruikers kunnen blokkeren wanneer deze zich niet aan de voorschriften houden. 14](#_Toc11089046)

[3.1.1. Wireframe’s 14](#_Toc11089047)

[3.1.2. Functioneel ontwerp 14](#_Toc11089048)

[3.1.3. Technisch ontwerp 14](#_Toc11089049)

[3.1.4. Pseudo code 14](#_Toc11089050)

[3.1.5. Activity Diagram 14](#_Toc11089051)

[3.1.6. Testcases 14](#_Toc11089052)

[3.2. User story 2 - Als beheerder wil ik veilingen kunnen blokkeren wanneer deze zich niet aan de voorschriften houden. 14](#_Toc11089053)

[3.2.1. Wireframe’s 14](#_Toc11089054)

[3.2.2. Functioneel ontwerp 14](#_Toc11089055)

[3.2.3. Technisch ontwerp 14](#_Toc11089056)

[3.2.4. Pseudo code 14](#_Toc11089057)

[3.2.5. Activity Diagram 14](#_Toc11089058)

[3.2.6. Testcases 14](#_Toc11089059)

[3.3. User story 3 - Als gebruiker wil ik contact op kunnen nemen met de verkoper van een voorwerp, zodat ik mijn probleem kan voorleggen. 14](#_Toc11089060)

[3.3.1. Wireframe’s 14](#_Toc11089061)

[3.3.2. Functioneel ontwerp 14](#_Toc11089062)

[3.3.3. Technisch ontwerp 14](#_Toc11089063)

[3.3.4. Pseudo code 14](#_Toc11089064)

[3.3.5. Activity Diagram 14](#_Toc11089065)

[3.3.6. Testcases 14](#_Toc11089066)

[3.4. User story 4 - Als beheerder wil ik dat de website veilig genoeg is zodat bindende financiële transacties kunnen worden gedaan. 15](#_Toc11089067)

[3.4.1. Wireframe’s 15](#_Toc11089068)

[3.4.2. Functioneel ontwerp 15](#_Toc11089069)

[3.4.3. Technisch ontwerp 15](#_Toc11089070)

[3.4.4. Pseudo code 15](#_Toc11089071)

[3.4.5. Activity Diagram 15](#_Toc11089072)

[3.4.6. Testcases 15](#_Toc11089073)

[3.5. User story 5 - Als gebruiker wil ik mijn profiel kunnen aanpassen zodat ik mijn gegevens up-to-date kan houden. 15](#_Toc11089074)

[3.5.1. Wireframe’s 15](#_Toc11089075)

[3.5.2. Functioneel ontwerp 15](#_Toc11089076)

[3.5.3. Technisch ontwerp 15](#_Toc11089077)

[3.5.4. Pseudo code 15](#_Toc11089078)

[3.5.5. Activity Diagram 15](#_Toc11089079)

[3.5.6. Testcases 15](#_Toc11089080)

[3.6. User story 6 - Als beheerder wil ik de rubriekenboom kunnen beheren door het toevoegen, verwijderen, sorteren, hernoemen en/of uitfaseren van rubriekenboom. 15](#_Toc11089081)

[3.6.1. Wireframe’s 15](#_Toc11089082)

[3.6.2. Functioneel ontwerp 15](#_Toc11089083)

[3.6.3. Technisch ontwerp 15](#_Toc11089084)

[3.6.4. Pseudo code 15](#_Toc11089085)

[3.6.5. Activity Diagram 15](#_Toc11089086)

[3.6.6. Testcases 15](#_Toc11089087)

[3.7. User story 7 - Als gebruiker wil ik op de hoogte worden gehouden van de status van een veiling, zodat ik eventueel naar vergelijkbare voorwerpen op zoek kan gaan. 16](#_Toc11089088)

[3.7.1. Wireframe’s 16](#_Toc11089089)

[3.7.2. Functioneel ontwerp 16](#_Toc11089090)

[3.7.3. Technisch ontwerp 16](#_Toc11089091)

[3.7.4. Pseudo code 16](#_Toc11089092)

[3.7.5. Activity Diagram 16](#_Toc11089093)

[3.7.6. Testcases 16](#_Toc11089094)

[3.8. User story 8 - Als gebruiker wil ik een nieuw wachtwoord kunnen opgeven wanneer ik het wachtwoord ben vergeten. 16](#_Toc11089095)

[3.8.1. Wireframe’s 16](#_Toc11089096)

[3.8.2. Functioneel ontwerp 16](#_Toc11089097)

[3.8.3. Technisch ontwerp 16](#_Toc11089098)

[3.8.4. Pseudo code 16](#_Toc11089099)

[3.8.5. Activity Diagram 16](#_Toc11089100)

[3.8.6. Testcases 16](#_Toc11089101)

[3.9. Overige taken. 16](#_Toc11089102)

# Ontwerpdocument groep 15

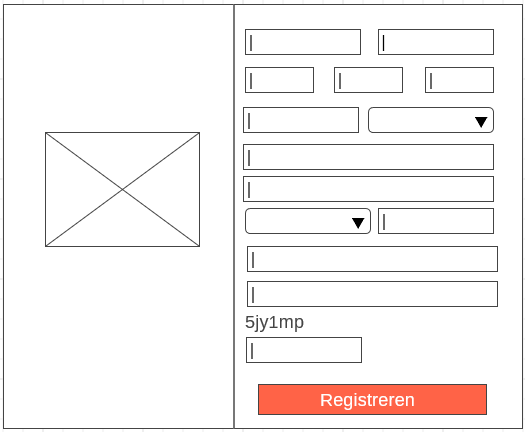
# Sprint 1

# Sprint 2

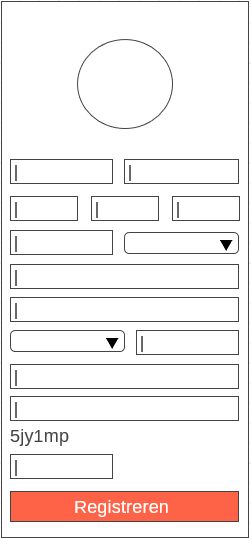
## 2.1. User story 1 - Als gast wil ik mijzelf kunnen registreren zodat ik later mee kan bieden en evt. kan veilen.

### 2.1.1. Wireframe’s

**Registreren pagina wireframe:**Dit staat onder de header en boven de footer.



**Registreren pagina wireframe 2 (Gekozen):**  
Dit hebben we uiteindelijk gekozen, omdat ieder teamlid hierop heeft gestemd als eindproduct voor het registreren pagina. Dit staat onder de header en boven de footer.



### 2.1.2. Functioneel ontwerp

Op deze pagina heeft de gebruiker de mogelijkheid om zichzelf te kunnen registreren, om later een bod uit te brengen op een veiling. Wanneer er op de knop “Registreren” wordt geklikt, nadat alle velden gevuld zijn wordt de gebruiker doorgestuurd naar de volgende stap. De volgende stappen zullen worden behandeld in de volgende user story. Dit heeft namelijk betrekking op het verifiëren van het account.

### 2.1.3. Technisch ontwerp

De gegevens Voornaam, Achternaam, Adresregel, Adresregel 2 (optioneel), Postcode, Plaatsnaam, Land, Datum, Emailadres, Vraag, Antwoord, Gebruikersnaam, Wachtwoord, captcha. De data wat hier wordt ingevuld zal worden opgeslagen in de database nadat de gebruiker op de knop “Registreren” heeft geklikt. De captcha wordt gecheckt of de tekst op de afbeelding hetzelfde is als in het tekst vak. Als er een veld niet is ingevuld dat zal er een pop-up komen, na het drukken op het “Registreren”, met dat het betreffende veld nog niet is ingevuld.

### 2.1.4. Pseudo code

Verbind met database

Als ingevulde adresregel2 leeg is zet $adresregel2 = null

$verkoper is standaard 0

$wachtwoord is de hash van ingevulde wachtwoord

(alle andere ingevulde waarden worden in hun respectievelijke variabelen gestopt)

QUERY

INSERT alle waarden op volgorde in de Gebruiker tabel

Redirect naar index

### 2.1.5. Activity Diagram

### 2.1.6. Testcases

Erkan Alper 613535

|  |  |  |  |
| --- | --- | --- | --- |
| ***Registreren*** | | | |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Een juiste input van alle velden die verplicht zijn. | Voornaam: <voornaam>  Achternaam: <achternaam>  Adresregel: <adresregel>  Postcode: <postcode>  Plaatsnaam: <plaatsnaam>  Emailadres: <emailadres>  Antwoord: <antwoord>  Gebruikersnaam: <gebruikersnaam>  Wachtwoord: <wachtwoord>  Captcha: <captcha> | Data wordt naar de database gestuurd. | Data wordt ingevoerd in de database. |
| Onjuiste captcha | Captcha: <captcha> | Ingevoerde code komt overeen met de afbeelding | De gebruiker |
| Symbolen invoeren in plaats van tekst | Voornaam: <Erkan@>  Achternaam: <Alper@>  Adresregel: <adresregel@>  Postcode: <postcode#>  Plaatsnaam: <plaatsnaam%>  Antwoord: <antwoord\*>  Gebruikersnaam: <gebruikersnaam>  Wachtwoord: <wachtwoord> | Foutmelding tonen. | Input wordt goedgekeurd. |
| HTML-tags invoeren in tekst vakken | In alle tekstvakken </div> ingevoerd | Foutmelding tonen. | Input wordt goedgekeurd. |

Hieruit blijkt dat de velden nog moeten worden gesanitised op HTML-tags en symbolen.

## 2.2. User story 2 - Als beheerder wil ik dat gebruiker accounts worden geverifieerd, zodat de kans op nepaccounts wordt verkleind.

### 2.2.1. Wireframe’s

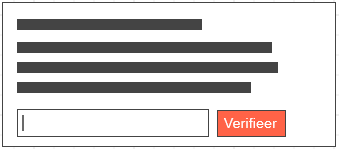
**Mailversturen pagina wireframe:**

Dit staat onder de header en boven de footer.



**Mailverstuurd pagina wireframe:**

Dit staat onder de header en boven de footer.



### 2.2.2. Functioneel ontwerp

**Mailversturen pagina:**

Op deze pagina moet de gebruiker zijn mail bevestigen, door in het tekst vak zijn emailadres nogmaals in te voeren ter bevestiging. Wanneer er op de knop “Verzend” wordt geklikt, krijgt de gebruiker een bevestiging mail in zijn postvak. Hierin staat een code.

**Mailverstuurd pagina:**

Op deze pagina moet de gebruiker zijn code invoeren, wat vanuit de mail komt. Wanneer dit is ingevoerd en op “Verzend” is geklikt zal de gebruiker doorgestuurd worden naar het dashboard van de account.

### 2.2.3. Technisch ontwerp

**Mailversturen pagina:**

Nadat er op de knop wordt gedrukt. Zal er worden gecheckt of het opgegeven emailadres in het tekst vak hetzelfde is als voorheen. Als dit overeen komt zal er een bevestigingsmail gestuurd worden naar de gebruiker.

**Mailverstuurd pagina:**

Hier wordt gecontroleerd of de code vanuit de mail overeenkomt met het ingevoerde code in het tekst vak, Als er op de knop “Verifieer” wordt geklikt.

### 2.2.4. Pseudo code

### 2.2.5. Activity Diagram

### 2.2.6. Testcases

Erkan Alper 613535

|  |  |  |  |
| --- | --- | --- | --- |
| ***Emailadres invoeren voor bevestigingsmail met code*** | | | |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Een juiste input van het veld, wat verplicht is | <Emailadres> | De gebruiker krijgt een mail, met daarin een bevestigingscode. | De gebruiker krijgt een mail, met daarin een bevestigingscode. |
| Een onjuiste input | <Emailadres> | De gebruiker krijgt geen mail. | De gebruiker krijgt geen mail. |

|  |  |  |  |
| --- | --- | --- | --- |
| ***Bevestigingscode invoeren.*** | | | |
| Code invoeren - Een juiste input van het veld, wat verplicht is. | <Bevestigingscode> | De gebruiker verifieerd zijn account en wordt doorverwezen naar het dashboard van het account. | De gebruiker verifieerd zijn account en wordt doorverwezen naar het dashboard van het account. |
| Code invoeren - Een onjuiste input | <Bevestigingscode> | De gebruiker wordt niet geverifieerd en niet doorverwezen naar het dashboard van account. | De gebruiker wordt niet geverifieerd en niet doorverwezen naar het dashboard van account. |

Hieruit blijkt dat alles naar behoren werkt en de gebruiker succesvol kan worden geverifieerd.

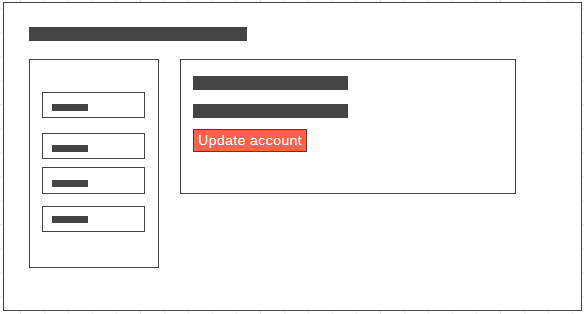
## 2.3. User story 3 - Als gebruiker wil ik kunnen inloggen zodat ik meet interactie met de site kan hebben.

### 2.3.1. Wireframe’s

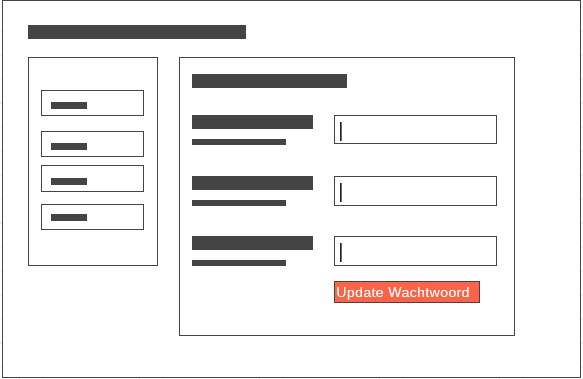
**Inloggen pagina wireframe:**Dit staat onder de header en boven de footer.



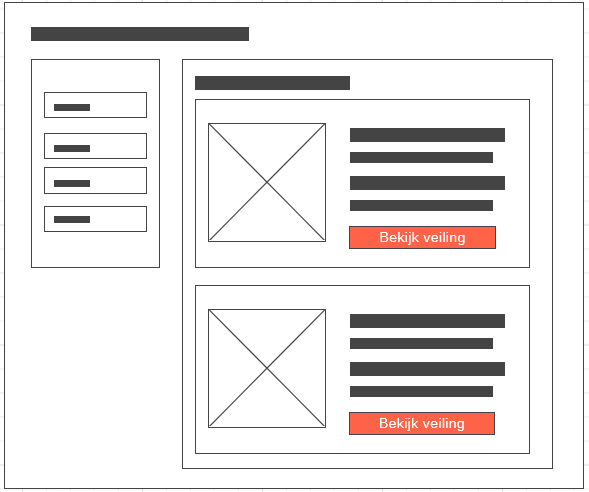
**Dashboard pagina wireframe:** Dit staat onder de header en boven de footer.



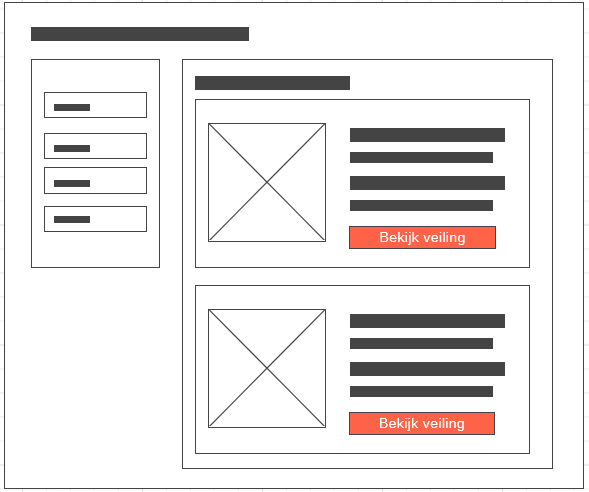
**Dashboard wachtwoord veranderen:**Dit staat onder de header en boven de footer.



**Dashboard lopende veilingen:**Dit staat onder de header en boven de footer.



**Dashboard uw vellingen:**Dit staat onder de header en boven de footer.



### 2.3.2. Functioneel ontwerp

**Inloggen pagina:**

Op deze pagina kan de gebruiker inloggen door zijn gebruikersnaam en wachtwoord in te vullen. Hierna wordt er geklikt op de knop “inloggen” om op het dashboard te komen.

**Dashboard pagina:**

Op deze pagina komt de ziet de gebruiker het dashboard van zijn profiel. Hier kan er worden gekozen om op de knop “Update account” te klikken, dat de gebruiker Verkoper wordt. De gebruiker kan via zijn dashboard verschillende handelingen inzien en uitvoeren namelijk: wachtwoord veranderen, Lopende veilingen inzien, en uw veilingen inzien.

**Dashboard wachtwoord veranderen pagina:**

Op deze pagina kan de gebruiker zijn wachtwoord veranderen door eerst het oude wachtwoord, en daarna het nieuwe wachtwoord met een herhaling. Door op de knop “Update wachtwoord” te klikken wordt het wachtwoord veranderd. Daarna wordt de gebruiker doorgestuurd naar het dashboard.

**Dashboard lopende veilingen pagina:**

Op deze pagina kan de gebruiker zien op welke veilingen zijn geboden vanaf het ingelogde account. Wanneer er op de knop ”Bekijk veiling” wordt geklikt, wordt de gebruiker doorgestuurd naar de betreffende veiling.

**Dashboard uw veilingen pagina:**

Op deze pagina kan de gebruiker zijn veilingen zien die zijn geplaatst vanaf het ingelogde account. Wanneer er op de knop “Bekijk veiling” wordt geklikt, wordt de gebruiker doorgestuurd naar de betreffende veiling.

2.3.3. Technisch ontwerp  
**Inloggen pagina:**

Hier worden de gegevens van de velden gebruikersnaam en wachtwoord vergleken met de gegevens uit de database. Als de data overeenkomen wordt de gebruiker toegelaten tot het dashboard.

**Dashboard pagina:**

Hier wordt pill menu getoond.

**Dashboard wachtwoord veranderen pagina:**

Hier wordt het nieuwe wachtwoord veranderd met het oude wachtwoord uit de database, Als er op de knop “Update wachtwoord” wordt geklikt.

**Dashboard lopende veilingen pagina:**

Hier worden alle lopende veilingen van de gebruiker uit de database gehaald waarop de gebruiker heeft geboden.

**Dashboard uw veilingen pagina:**

Hier worden alle veilingen wat door de gebruiker zijn geplaats uit de database gehaald.

### 2.3.4. Pseudo code

**Inloggen**

Verbind met database

Start sessie.

$gebruikersnaam = ingevulde gebruikersnaam

$wachtwoord = ingevulde wachtwoord

QUERY

Selecteer wachtwoord van gebruiker tabel waar gebruikersnaam gelijk is aan $gebruikersnaam

Als $wachtwoord verifieerd met wachtwoord hash uit database maak dan een sessie variabele userID met de waarde $gebruikersnaam

Als wachtwoord niet verifieerd redirect naar volledige login pagina met foutmelding

### 2.3.5. Activity Diagram

### 2.3.6. Testcases

Erkan Alper 613535

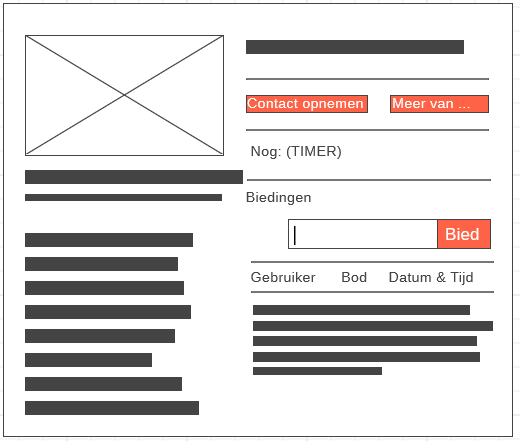
|  |  |  |  |
| --- | --- | --- | --- |
| ***Wachtwoord veranderen*** | | | |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Juiste input | <Wachtwoord>  <Nieuw wachtwoord>  <Nieuw wachtwoord herhalen> | Wachtwoord van de gebruiker wordt veranderd naar het nieuwe ingevoerde wachtwoord. | Wachtwoord van de gebruiker wordt veranderd naar het nieuwe ingevoerde wachtwoord. |
| Juiste input met meer dan 10 karakters. | <Nieuw wachtwoord>  <Nieuw wachtwoord> | Wachtwoord wordt niet veranderd. | Wachtwoord wordt niet veranderd. |
| Onjuiste input | <Wachtwoord> | Wachtwoord wordt niet veranderd. | Wachtwoord wordt niet veranderd. |
| Onjuiste input | <Nieuw wachtwoord>  <Nieuw wachtwoord herhalen> | Wachtwoord wordt niet veranderd. | Wachtwoord wordt niet veranderd. |

Hieruit blijkt dat de velden nog moeten worden gesanitised wanneer, er niet hetzelfde wachtwoord twee keer is ingevuld.

## 2.4. User story 4 - Als gebruiker wil ik op een voorwerp kunnen bieden zodat ik deze mogelijk kan aanschaffen.

### 2.4.1. Wireframe’s

**Biedingspagina wireframe:**  
Dit staat onder de header en boven de footer:



### 2.4.2. Functioneel ontwerp

**Biedingspagina:**

Links op het scherm wordt de advertentie getoond met daarboven de afbeelding(en) met een slider. Onderin wordt de titel en de omschrijving getoond.

Rechtsboven wordt de naam van de verkoper getoond, met daaronder een timer en de biedingen. Daaronder wordt een lijst met: Gebruiker, Bod, Datum & Tijd getoond.

Op deze pagina kan de gebruiker na het inloggen een bod uitbrengen, door op de knop “Bied” te drukken na een prijs te hebben ingevoerd in het tekst vak ernaast met een bedrag.

### 2.4.3. Technisch ontwerp

### 2.4.4. Pseudo code

### 2.4.5. Activity Diagram

### 2.4.6. Testcases

Erkan Alper 613535

|  |  |  |  |
| --- | --- | --- | --- |
| ***Biedingspagina*** | | | |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Juiste input van het bod. | <Bedrag> | Het ingevoerde bedrag wordt geaccepteerd en weergegeven als hoogste bod. | De ingevoerde prijs wordt geaccepteerd en weergegeven als hoogste bod. |
| Onjuiste input van het bod | <Bedrag> | Het ingevoerde bedrag wordt niet geaccepteerd en krijgt een melding “Selecteer een waarde die niet lager is dan <Bedrag>.” | Het ingevoerde bedrag wordt niet geaccepteerd en krijgt een melding “Selecteer een waarde die niet lager is dan <Bedrag>.” |
| Onjuiste input van het bod. Tekst en of karakters in plaats van bedragen | <Bedrag> | Het ingevoerde bedrag wordt niet geaccepteerd en krijgt een melding “voer een getal in.” | Het ingevoerde bedrag wordt niet geaccepteerd en krijgt een melding “voer een getal in.” |

Hieruit blijkt dat de veld voor het bieden werkt en is gesanitised voor onjuiste inputs.

## 2.5. User story 5 - Als gebruiker wil ik kunnen uitloggen zodat anderen niet met mijn account kunnen werken.

### 2.5.1. Wireframe’s

### 2.5.2. Functioneel ontwerp

### 2.5.3. Technisch ontwerp

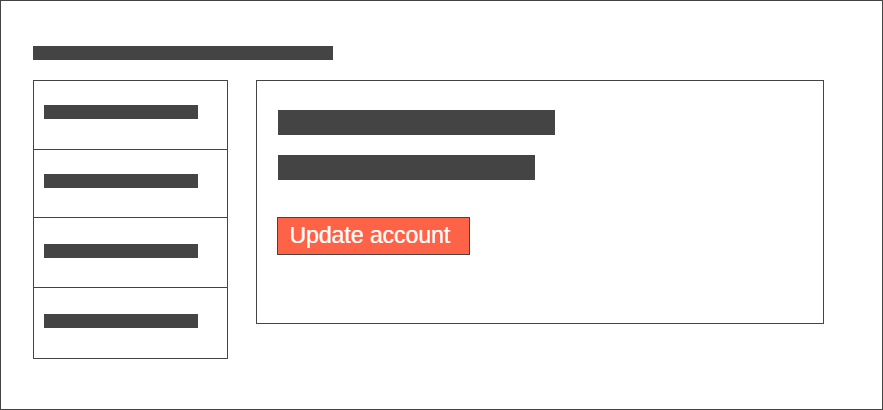
### 2.5.4. Pseudo code

### 2.5.5. Activity Diagram

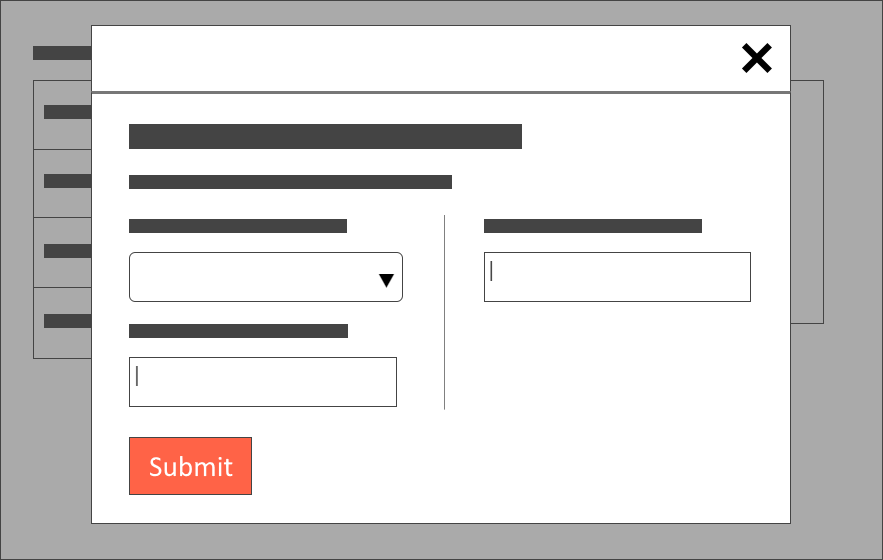
### 2.5.6. Testcases

## 2.6. User story 6 - Als gebruiker wil ik een verkoper worden om daarna zelf voorwerpen aan te kunnen bieden.

### 2.6.1. Wireframe’s

**Wireframe update naar verkoper knop:**Als je ingelogd bent kun je naar je dashboard, op dit scherm staat een pillmenu met in de eerste optie een scherm waarop je je account kunt upgraden tot verkoper.  
  


**Wireframe modal upgrade:**Als op de update account knop gedrukt komt er een venster tevoorschijn waar betaalgegevens ingevuld kunnen worden die nodig zijn om een verkoper te worden. Links staat een dropdown met banken en een vak om je IBAN in te vullen, rechts staat een vak om je creditcard nummer in te vullen.



### 2.6.2. Functioneel ontwerp

Om verkoper te worden moet dit formulier ingevuld worden, je betaalgegevens moeten namelijk bekend zijn. Bij het invullen van het formulier worden een twee checks uitgevoerd:

* Er wordt gecheckt of er minimaal 1 van de 2 betaalmanieren is ingevuld.
* Er wordt gecheckt of de ingevulde waardes echte bank/creditcard gegevens zijn.

Als de input door deze checks heen komt word de data in de database gezet, dit moet dan nog goedgekeurd worden door een beheerder. Als dit gebeurt is wordt de gebruiker gezien als een verkoper.

### 2.6.3. Technisch ontwerp

Als op submit wordt geklikt wordt er met behulp van javascript gekeken of er minimaal 1 tekstvak is ingevuld, als dit tekstvak de IBAN is wordt er ook gecheckt of de gebruiker een bank heeft geselecteerd uit het dropdown menu.

Vervolgens wordt er met behulp van php gecheckt of de gegevens kloppen, dit wordt gedaan door de IBAN/cc nummers langs gaan en kijken of ze de algoritmes volgen die gebruikt worden bij het aanmaken van zo een nummer.

Als deze gegevens kloppen worden ze in de Verkoper tabel gezet met in de kolom Controle\_optie de waarde “In afwachting”. De beheerder kan deze mensen dan goedkeuren door in de Gebruiker tabel in de kolom Verkoper de 0 in een 1 te veranderen.

### 2.6.4. Pseudo code

<html>

<head>

…

</head>

<header>

…

</header>

<body>

<?php

if (gebruiker == verkoper) {

if (gekeurde gebruiker == true) {

<button>Maak veiling</button>

} else {

<p>U moet wachten tot uw betaalgegevens goedgekeurd worden door een beheerder</p>

}

} else {

<button>Update account</button>

}

?>

if (geklikt op update account) {

pop-up window met volgende gegevens:

* Banknaam
  + Rekeningnummer
* Creditcardnummer

// minimaal 1 van de 2 moet worden ingevuld

if (geklikt op send) {

<?php

// Check of minimaal 1 is ingevuld

// Check of het een echte IBAN/creditcard is

//maak de veranderingen aan in de database gebaseerd op wat is ingevuld

?>

}

}

</body>

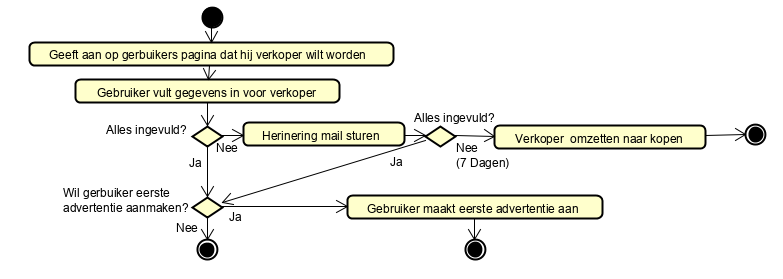
<footer>

…

</footer>

</html>

### 2.6.5. Activity Diagram



2.6.6. Testcases

Peiwand Ismaiel 619856

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Een juiste IBAN + Bank | Bank: <bijbehorende bank>  IBAN: <geldige IBAN>  Cc nummer: *leeg* | Data wordt in de verkoper tabel toegevoegd en de gebruiker krijgt een melding hierover | Data wordt in de verkoper tabel toegevoegd en de gebruiker krijgt een melding hierover |
| Een juiste creditcard | Bank: *leeg*  IBAN: *leeg*  Cc nummer: <geldige cc nummer> | Data wordt in de verkoper tabel toegevoegd en de gebruiker krijgt een melding hierover | Data wordt in de verkoper tabel toegevoegd en de gebruiker krijgt een melding hierover |
| Een juiste IBAN + Bank en juiste creditcard | Bank: <bijbehorende bank>  IBAN: <geldige IBAN>  Cc nummer: <geldige cc nummer> | Data wordt in de verkoper tabel toegevoegd en de gebruiker krijgt een melding hierover | Data wordt in de verkoper tabel toegevoegd en de gebruiker krijgt een melding hierover |
| Een onjuiste IBAN + Bank | Bank: *maakt niet uit*  IBAN: <ongeldige IBAN>  Cc nummer: *leeg* | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover |
| Een onjuiste creditcard | Bank: *leeg*  IBAN: *leeg*  Cc nummer: <ongeldige cc nummer> | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover |
| Een onjuiste IBAN + Bank en juiste creditcard | Bank: *maakt niet uit*  IBAN: <ongeldige IBAN>  Cc nummer: <geldige cc nummer> | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover |
| Een juiste IBAN + Bank en onjuiste creditcard | Bank: <bijbehorende bank>  IBAN: <geldige IBAN>  Cc nummer: <ongeldige cc nummer> | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover |
| Een onjuiste IBAN + Bank en onjuiste creditcard | Bank: *maakt niet uit*  IBAN: <ongeldige IBAN>  Cc nummer: <ongeldige cc nummer> | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover |
| Gebruiker staat al in database | Geen input mogelijk | Knop wordt disabled | Knop wordt disabled |

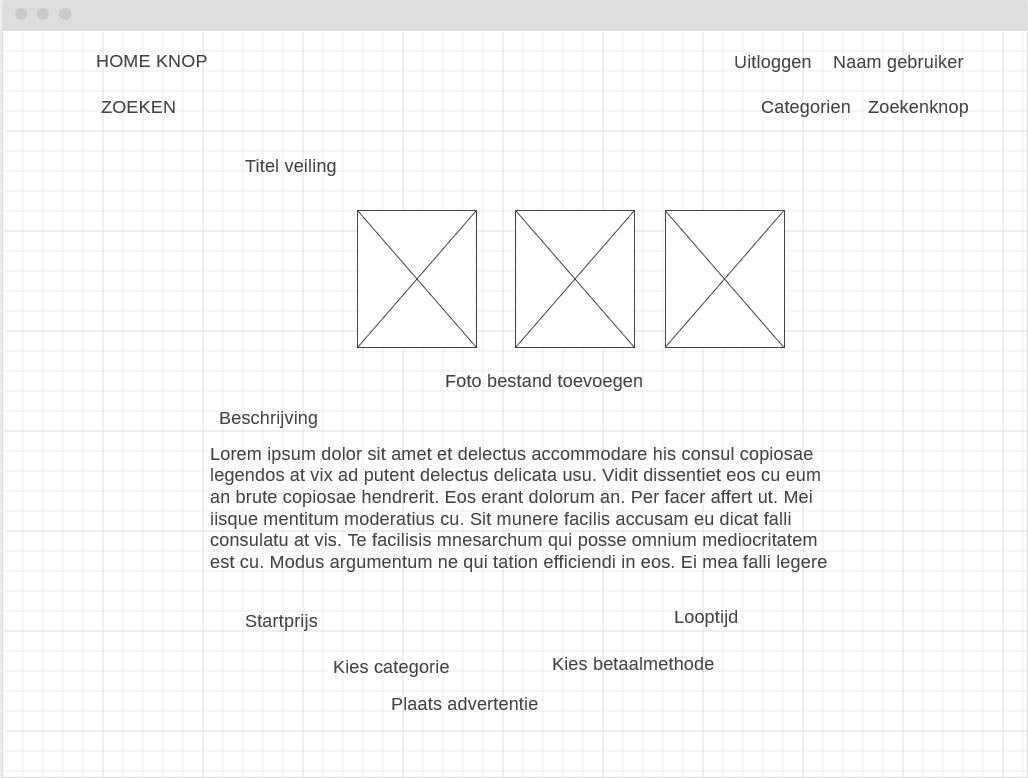
Geldige IBAN via:  
<https://www.tuxx.nl/betalingsverkeer/iban_voorbeelden/>

Bij testen voor ongeldige IBAN gebruik:  
<https://www.ibancalculator.com/iban_validieren.html>

Geldige cc nummer en test voor ongeldige via:  
<https://www.freeformatter.com/credit-card-number-generator-validator.html>

## 2.7. User story 7 - Als verkoper wil ik zelf voorwerpen kunnen aanbieden zodat deze kunnen worden geveild.

### 2.7.1. Wireframe’s



### 2.7.2. Functioneel ontwerp

Hierboven is een wireframe te zien van de pagina waar een veiling kan worden aangemaakt door een verkoper. We zien hier meerdere mogelijkheden voor de verkoper. De verkoper kan een titel verzinnen, een beschrijving toevoegen, een startprijs bepalen, de looptijd, de categorie waarin de veiling moet plaats vinden en hoe er betaald zal moeten worden door de uiteindelijke koper.

Nadat alle velden zijn ingevuld zal de verkoper de mogelijkheid krijgen om de advertentie te plaatsen en is het gelijk te zien op de website van EenmaalAndermaal.

##### Argumentatie voor keuze van het ontwerp

Er is gekozen voor een simpel en overzichtelijke opmaak. Dit zodat de verkoper goed ziet wat hij doet, en geen verkeerde beslissingen maakt. Daarnaast is dit ook hoe de veiling eruit komt te zien , in grote lijnen, voor de kopers op de site. Met uitzondering van de foto’s , deze worden met een slider een voor een getoond.

### 2.7.3. Technisch ontwerp

Alle velden worden na het drukken op de “advertentie plaatsten” knop in de database gezet. De foto’s worden in de tabel “Bestand” geplaatst samen met een nieuw gemaakte voorwerpnummer. Verder wordt na het drukken op de knop “advertentie plaatsten” alle velden gecontroleerd of ze wel zijn ingevuld. Hiernaast is een belangrijke vereiste dat het gaat om een geregistreerde verkoper, je kan als normale gebruiker of gast geen veiling toevoegen. Dit is ook te zien in de wireframe.

### 2.7.4. Pseudo code

Require DB

Start Sesie

$cat = post

$titel = post

$startprijs = post

$looptijd = post

$looptijdbegin = date(“Y-m-d”)

$looptijdEind = lootijdbegin + looptijd

$betaalwijze = post

$betalinginstructie = null

$vekoopprijs = null

$veilinggesloten = “0”

$verzendkosten =post

$query = INSERT INTO Voorwerp

Functie Compres Image($Quality, $source){}

If($image1 == 1){

Move\_uloaded\_file (‘DT1\_’ . $Veilingnummer . $exstensie)

}

If($image2 == 1){

Move\_uloaded\_file (‘DT2\_’ . $Veilingnummer . $exstensie)

}

If($image3 == 1){

Move\_uloaded\_file (‘DT3\_’ . $Veilingnummer . $exstensie)

}

If($image4 == 1){

Move\_uloaded\_file (‘DT4\_’ . $Veilingnummer . $exstensie)

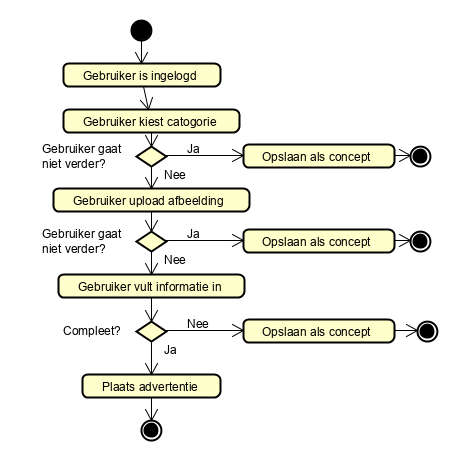
}

If($image1 == 0 and $image2 == 0 and $image3 == 0 and $image4 == 0){

Copy(standaard.png, dt1\_ $voorwerpnummer .png

}

### 2.7.5. Activity Diagram



### 2.7.6. Testcases

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Test input Titel | Titel: hierin komt de titel van de veiling | Data wordt in de voorwerp tabel toegevoegd | Data wordt in de voorwerp tabel toegevoegd |
| Input beschrijving | Beschrijving: hierin komt de beschrijving van de veiling voorwerp | Data wordt in de voorwerp tabel toegevoegd | Data wordt in de voorwerp tabel toegevoegd |
| Foto toevoegen | Foto: JPG, PNG foto via “bestand toevoegen” toevoegen | Data wordt in de Bestand tabel toegevoegd | Data wordt in de Bestand tabel toegevoegd |
| Startprijs aangeven | Startprijs: geef een startprijs voor de veiling | Data wordt in de voorwerp tabel toegevoegd | Data wordt in de voorwerp tabel toegevoegd |
| Geen startprijs aangegeven | Startprijs: leeg | Komt pop up met dat veld leeg is | Komt pop up met dat veld leeg is |

|  |  |  |  |
| --- | --- | --- | --- |
| Looptijd in dagen, categorie, verzendkosten, betaalmethode aangeven | Looptijd: keuze uit de mogelijkheden moet gemaakt zijn  Categorie: keuze uit de mogelijkheden moet gemaakt zijn  Verzendkosten : keuze uit de mogelijkheden moet gemaakt zijn  Betaalmethode : keuze uit de mogelijkheden moet gemaakt zijn | Data wordt in de voorwerp tabel toegevoegd | Data wordt in de voorwerp tabel toegevoegd |
| Looptijd in dagen, categorie, verzendkosten, betaalmethode. Een van deze is niet aangegeven | Een van de vakken is leeg | Komt pop up met dat veld leeg is | Komt pop up met dat veld leeg is |

## 2.8. User story 8 - Als verkoper wil ik op de hoogte gebracht worden wanneer mijn veiling beëindigd, zodat ik afspraken kan maken met een eventuele winnaar.

### 2.8.1. Wireframe’s

### 2.8.2. Functioneel ontwerp

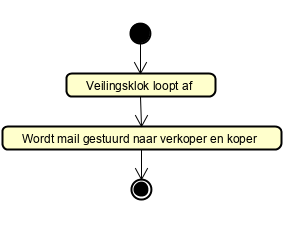
De website zal zodra een veiling eindigt kijken welke gebruiker het hoogste bod heeft geplaatst, vervolgens zal deze gebruiker een mail krijgen die hem informeert dat hij gewonnen heeft.

### 2.8.3. Technisch ontwerp

Als een veiling gesloten probeert te worden wordt er eerst gekeken of de looptijd van de veiling is verstreken, is dit niet het geval dan wordt de veiling niet gesloten en wordt er geen mail verstuurd. Als de looptijd wel verstreken is wordt er met een SQL query data van deze veiling en de gebruiker die gewonnen heeft opgehaald. Deze data wordt vervolgens met de mail() functie gebruikt om naar de koper een mail te sturen waarin hem verteld welke veiling hij gewonnen heeft.

### 2.8.4. Pseudo code

### 2.8.5. Activity Diagram



### 2.8.6. Testcases

Peiwand Ismaiel 619856

|  |  |  |
| --- | --- | --- |
| Test | Verwacht resultaat | Werkelijk resultaat |
| Veiling eindigt met een winnaar | De winnaar van de veiling en de eigenaar van de veiling krijgen een bericht met een melding over de veiling. | De winnaar van de veiling en de eigenaar van de veiling krijgen een bericht met een melding over de veiling. |
| Veiling eindigt zonder dat er op geboden is | De eigenaar van de veiling krijgt een bericht waarin staat dat er niet geboden is op zijn veiling. | De eigenaar van de veiling krijgt een bericht waarin staat dat er niet geboden is op zijn veiling. |

## 2.9. User story 9 - Als beheerder wil ik zien hoe de website op bedrijfsniveau presteert, zodat ik de strategie daarop kan aanpassen.

### 2.9.1. Wireframe’s

### 2.9.2. Functioneel ontwerp

### 2.9.3. Technisch ontwerp

### 2.9.4. Pseudo code

### 2.9.5. Activity Diagram

### 2.9.6. Testcases

Wesley Smeltink – 604792

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Count van het aantal gebruikersnamen die geactiveerd zijn. |  | Lijst van alle gebruikersnamen die geactiveerd zijn | Lijst van alle gebruikersnamen die geactiveerd zijn |
| Count van alle gebruikers die verkoper zijn |  | Lijst van alle gebruikersnamen die verkoper zijn | Lijst van alle gebruikersnamen die verkoper zijn |
| Count van het aantal veilingen in bepaalde rebrieken. |  | Van gekozen rubriek, veilingen laten zien | Van gekozen rubriek, veilingen laten zien |
| Count van de veilingen op looptijd |  | Lijst van alle veilingen, op looptijd gesorteerd | Lijst van alle veilingen, op looptijd gesorteerd |
| Cirkeldiagram van alle prijzen, opgedeeld in prijsgroepen. |  | Alle veilingen, opgedeeld in prijsgroepen | Alle veilingen, opgedeeld in prijsgroepen |

## 2.10. User story 10 - Als beheerder wil ik een overzichtelijke logging en presentatie daarvan zodat in de gaten kan houden of de site operationeel betrouwbaar is.

### 2.10.1. Wireframe’s

### 2.10.2. Functioneel ontwerp

### 2.10.3. Technisch ontwerp

### 2.10.4. Pseudo code

### 2.10.5. Activity Diagram

### 2.10.6. Testcases

## 2.11. User story 11 - Als Product Owner wil ik de opgekochte voorwerpen batchgewijs naar de site converteren, zodat ik kan starten met een gevulde website.

### 2.11.1. Wireframe’s

### 2.11.2. Functioneel ontwerp

### 2.11.3. Technisch ontwerp

### 2.11.4. Pseudo code

### 2.11.5. Activity Diagram

### 2.11.6. Testcases

Wesley Smeltink - 604792

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Conversie van Dataconversie database naar huidige database, iproject15 | Dataconversie script | De data van het database Dataconversie wordt succesvol naar onze huidige database ‘Iproject15’ geconverteerd, zonder foutmeldingen | De data van het database Dataconversie wordt succesvol naar onze huidige database ‘Iproject15’ geconverteerd, zonder foutmeldingen |
| Het maken van unieke data. | Gebruikersnaam + ‘@gmail.com’ | Iedere gebruiker heeft een zelfverzonnen uniek e-mail adres. | Iedere gebruiker heeft een zelfverzonnen uniek e-mail adres. |

|  |  |  |  |
| --- | --- | --- | --- |
| Bij het tabel ‘verkopers’ checken of de gebruiker een verkoper is of niet | Een update table regel die checkt op een bepaalde gebruiker een advertentie heeft. | Als de specifieke gebruiker een of meerdere advertenties heeft, dan wordt het kolom ‘Verkoper’ verandert van een ‘0’ naar ‘1’ | Als de specifieke gebruiker een of meerdere advertenties heeft, dan wordt het kolom ‘Verkoper’ verandert van een ‘0’ naar ‘1’ |
| Verwijdert gebruikers zonder advertenties bij de ‘Verkoper’ tabel. | Een delete script die kijkt naar of de waarde ‘verkoper’ in het tabel ‘0’ is. | Als de waarde ‘verkoper’ ‘0’ is, dan wordt die bepaalde gebruiker uit het tabel ‘Verkoper’ verwijdert. | Data wordt **niet** in de verkoper tabel toegevoegd en de gebruiker krijgt een foutmelding hierover |
| Conversie van Html beschrijving naar beschrijving zonder HTML of andere taal | Zelfgemaakte functie die HTML stript. | Alle verschillende tekens die te maken hebben met HTML of andere coderingstalen, worden verwijderd. | Beschrijving is HTML vrij. |
| Conversie van verschillende valuta’s naar euro. | Zelfgemaakte functie die Verschillende valuta’s naar euro converteerd | Er wordt aangegeven in de funtie welke valuta het is en wat het bedrag is, en dan wordt het gecoverteerd naar euro’s | Er wordt aangegeven in de functie welke valuta het is en wat het bedrag is, en dan wordt het geconverteerd naar euro’s |
| Testen van het constraint CK\_Controleoptie | Een Controle optie die niet in het reitje staat: Paypal | Constraint reageerd hierop, deze waarde mag niet worden ingevoerd | Constraint reageert hierop, deze waarde mag niet worden ingevoerd |

# Sprint 3

## 3.1. User story 1 - Als beheerder wil ik gebruikers kunnen blokkeren wanneer deze zich niet aan de voorschriften houden.

### 3.1.1. Wireframe’s

### 3.1.2. Functioneel ontwerp

### 3.1.3. Technisch ontwerp

### 3.1.4. Pseudo code

blokeren.php

<form handmatig(method POST)>

inputfile(POST = $username)

inputfile(POST = $reden)

inputfile(POST = $duur)

</form>

<form Blokeerbutton(method POST)>

button(value = $username)

inputfile(POST = $reden)

inputfile(POST = $duur)

</form>

if(!$duur){

$duur = NULL

}

$sql -> $username, $reden, $duur into Geblokeerde\_gebruikers

-------------------------------

deblokeren.php

<form handmatig(method POST)>

inputfile(POST = $username)

</form>

<form Deblokeerbutton(method POST)>

button(value = $username)

</form>

$sql -> Delte WHERE username = $username from Geblokeerde\_gebruikers

### 3.1.5. Activity Diagram

### 3.1.6. Testcases

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Gebruiker via blokkeer button blokkeren | Muisklik | Post naar Database | Post naar Database |
| Gebruiker blokkeren via naam | Naam en reden | Post naar Database | Post naar Database |
| Geen bestaande gebruiker blokeren via naam | Neppenaam en reden | geeft errorcode | geeft errorcode |
| Geen reden proberen mee te geven | Naam | geeft errorcode | geeft errorcode |
| Geen naam proberen mee te geven | Reden | geeft errorcode | geeft errorcode |
| Tijdsduur proberen mee te geven | naam reden tijdsduur | Post naar database | Post naar database |
| Extreem hoge tijdsduur proberen mee te geven | naam reden tijdsduur(9999) | Post naar database | Post naar database |
| Vul min 1 in bij duur | naam reden tijdsduur(-1) | Geeft errorcode | Geeft errorcode |
| Deblokkeren via button | muiskik | post naar database | post naar database |
| Deblokkeren via naam | naam | post naar database | post naar database |
| Deblokeren via nepnaam | niet geblokeerde naam | Geeft errorcode | Geeft errorcode |

### 

## 3.2. User story 2 - Als beheerder wil ik veilingen kunnen blokkeren wanneer deze zich niet aan de voorschriften houden.

### 3.2.1. Wireframe’s

### 3.2.2. Functioneel ontwerp

### 3.2.3. Technisch ontwerp

### 3.2.4. Pseudo code

blokeren2.php

<form handmatig(method POST)>

inputfile(POST = $advertentieID)

inputfile(POST = $reden)

</form>

<form Blokeerbutton(method POST)>

button(value = $advertentieID

inputfile(POST = $reden)

</form>

$sql -> $advertentieID $reden into Geblokeerde\_advertenties

-------------------------------

deblokeren.php

<form handmatig(method POST)>

inputfile(POST = $advertentieID)

</form>

<form Deblokeerbutton(method POST)>

button(value = $advertentieID)

</form>

$sql -> Delte WHERE advertentieID = $advertentieID from Geblokeerde\_Advertenties

### 3.2.5. Activity Diagram

### 3.2.6. Testcases

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Advertentie blokkeren via button blokkeren | Deblokkeren via nep Advertentie ID | Post naar Database | Post naar Database |
| Advertentie blokkeren via naam | Advertentie ID en reden | Post naar Database | Post naar Database |
| Geen bestaande Advertentie blokkeren via naam | Nep Advertentie ID en reden | geeft errorcode | geeft errorcode |
| Geen reden proberen mee te geven | Advertentie ID | geeft errorcode | geeft errorcode |
| Geen Advertentie ID proberen mee te geven | Reden | geeft errorcode | geeft errorcode |
| Deblokkeren via button | Muisklik | Post naar Database | Post naar Database |
| Deblokkeren via Advertentie ID | Naam | Post naar Database | Post naar Database |
| Deblokkeren via nep Advertentie ID | Niet geblokkeerde Advertentie ID | geeft errorcode | geeft errorcode |

## 3.3. User story 3 - Als gebruiker wil ik contact op kunnen nemen met de verkoper van een voorwerp, zodat ik mijn probleem kan voorleggen.

### 3.3.1. Wireframe’s

### 3.3.2. Functioneel ontwerp

### 3.3.3. Technisch ontwerp

Een wireframe is hier niet nodig omdat dit over een enkele knop gaat.

Om contact op te nemen met een verkoper gaat men naar de veilingpagina van een voorwerp van een verkoper. Hier staat de knop onder de naam van de verkoper. Als men is ingelogd kan er op deze knop worden gedrukt en kan een mail gestuurd worden.

### 3.3.4. Pseudo code

Bij laden van pagina:

If(gebruiker is ingelogd){

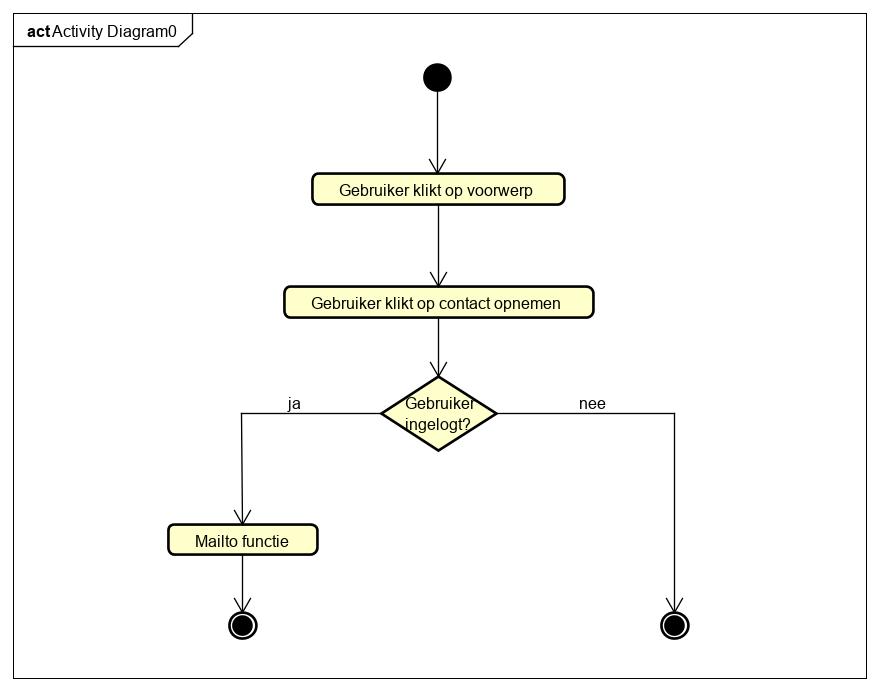
echo ‘HTML VOOR: Werkende contact opnemen knop met mailto functie en voorgeprogrammeerd mailadres’;

}else{

echo ‘HTML VOOR: Uitgeschakelde contact opnemen knop met popover dat gebruiker ingelogd moet zijn’;

}

### 3.3.5. Activity Diagram



### 3.3.6. Testcases

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Contact opnemen knop indrukken wanneer ingelogd | Indrukken knop | Mailprogramma opent met juiste emailadres ingevuld | Mailprogramma opent met juiste emailadres ingevuld |
| Contact opnemen knop indrukken wanneer niet ingelogd | Indrukken knop | Knop is disabled dus indrukken doet niks | Knop is disabled dus indrukken doet niks |
| Inspect html om te kijken of mailadres er staat wanneer niet ingelogd | F12 | Als er niet inglogd is staat de mailto niet in de knop | Als er niet inglogd is staat de mailto niet in de knop |

## 3.4. User story 4 - Als beheerder wil ik dat de website veilig genoeg is zodat bindende financiële transacties kunnen worden gedaan.

### 3.4.1. Wireframe’s

### 3.4.2. Functioneel ontwerp

### 3.4.3. Technisch ontwerp

### 3.4.4. Pseudo code

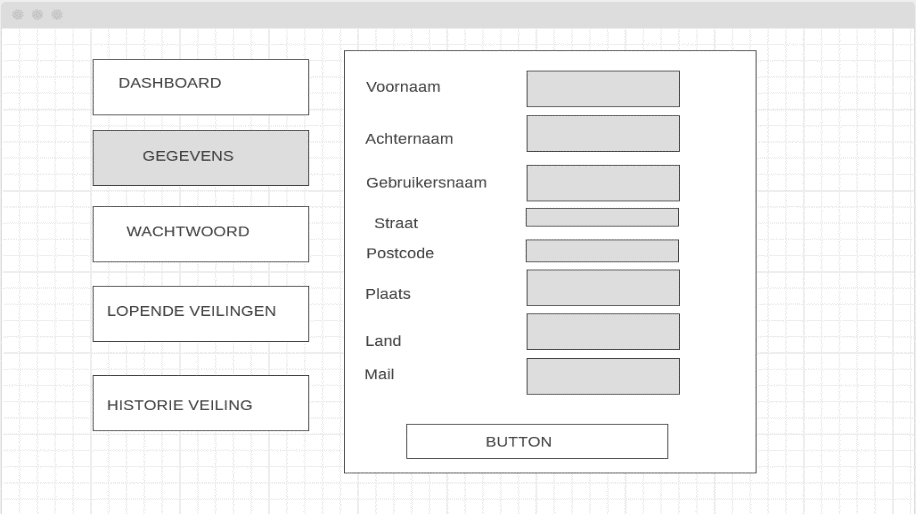
### 3.4.5. Activity Diagram

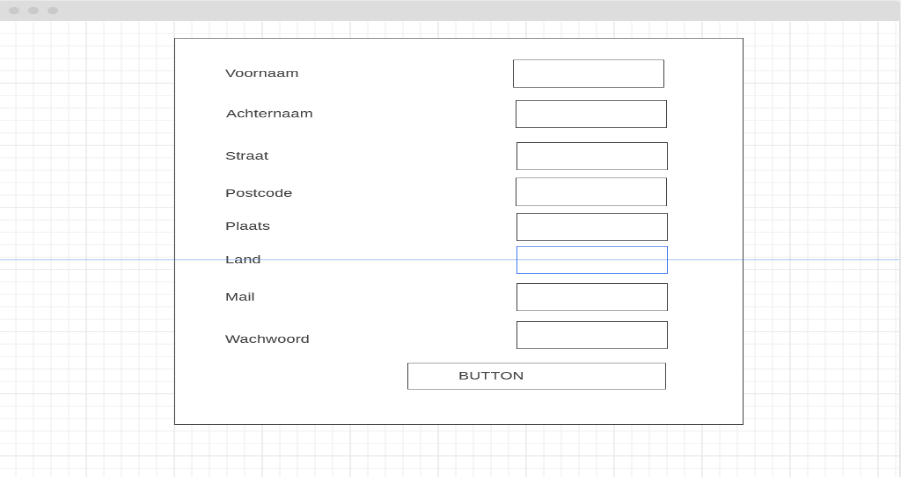
### 3.4.6. Testcases

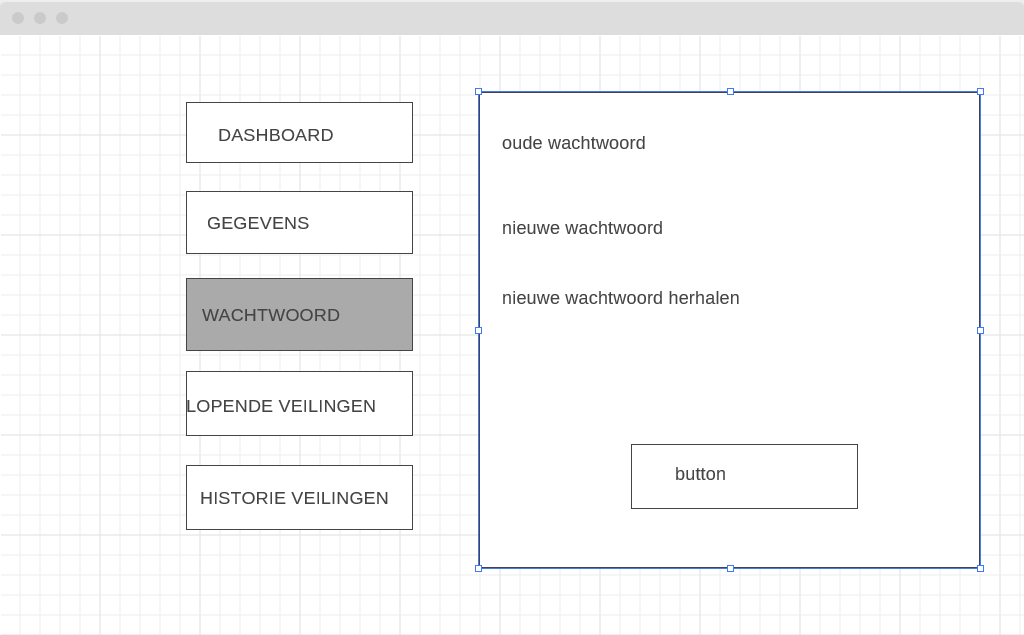
|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| XSS-aanval testen doormiddel van GET | <http://iproject15.icasites.nl/zoeken.php?zoeken=%27%3E>  <script>alert(1)</script> | Er moet geen javascript alert komen met het bericht ‘1’ | Er moet geen javascript alert komen met het bericht ‘1’ |
| XSS-aanval testen doormiddel van POST | [%27%3E%3Cscript%3Evar%20link%20](http://localhost/Website%20EenmaalAndermaal/bootstrap-4.3.1-dist/zoeken.php?zoeken=%27%3E%3Cscript%3Evar%20link%20)=  %20document.createElement(%27a%27);link.href%20=%20%27  <http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe%27>;  link.download%20=%20%27%27;document.body.appendChild(link);link.click();%3C/script%3E | Bij het openen van een veiling, moet er geen javascript worden uitgevoerd | Bij het openen van een veiling, wordter geen javascript worden uitgevoerd |
| Testen van HTML-injectie | <h2>Testtekst</h2> bij beschrijving veiling | De HTML-tags moeten negeert worden | HTML-tags worden negeert |
| Testen van SQL-injectie | SELECT \* FROM Gebruiker bij de beschrijving van een veiling | De SQL code wordt genegeerd | De SQL code wordt genegeerd |

## 3.5. User story 5 - Als gebruiker wil ik mijn profiel kunnen aanpassen zodat ik mijn gegevens up-to-date kan houden.

### 3.5.1. Wireframe’s







### 3.5.2. Functioneel ontwerp

Hierboven is de wireframe van de gegevenspagina. Hier zie je dat de gegevens allemaal disabled zijn. Je kan er dus niks in aanpassen. Om het te willen aanpassen komt er een pop up, deze is hieronder te zien. Als er gedrukt wordt op de button zal de gebruiker een pop up krijgen en deze ziet eruit als hieronder.

Hierboven is de pop up en hier zien we ook de Gebruikersnaam niet, dit omdat deze niet aan te passen is. Voordat alles wordt doorgevoerd wordt er nog een wachtwoord check gedaan.

Hierboven zie je dat de wachtwoord aangepast kan worden. Wachtwoord moet worden ingevuld. Als er op de button wordt gedrukt en alles komt overeen dan zal de wachtwoord gewijzigd worden en zal er een mail gestuurd worden.

### 3.5.3. Technisch ontwerp

**Niet aanpasbare gegevens**

Niet alle gegevens kunnen worden gewijzigd. Op de eerste pagina zien we alle gegevens van de gebruiker die is ingelogd. Deze gegevens zijn niet aan te passen hier. Om ze aan te passen zal de gebruiker op de Wijzigen knop moeten drukken. Als hierop gedrukt is zal de gebruiker een pop up zien. Alle gegevens worden uit de database gehaald van de betreffende gebruiker en laten zien.

**Aanpasbare gegevens**

Na het drukken op de Wijzigen knop zal de gebruiker naar de pop up gestuurd worden, zoals hiervoor al besproken. Hier zal de gebruiker alle mogelijke aanpasbare gegevens zien. Ze zijn hier niet meer disabled. De gebruiker ziet zijn eerder ingevulde (uit de database gehaalde gegevens) daar staan maar deze zijn aan te passen. Om te kiezen welke er gepakt moet worden wordt er een if statement per input field gemaakt. We stoppen eerst de oude gegevens in een nieuwe variabele en als de input field niet wordt aangepast dan blijft deze, wordt de input wel aangepast dan zal door die if statement de nieuwe variabele aangepast worden naar de nieuwe waarden. Omdat de gebruikersnaam niet aan te passen is zal deze ook niet hier laten zien worden. Om het door te kunnen voeren wordt er een check uitgevoerd met een wachtwoord, de wachtwoord die ingevuld is moet overeen komen met de bekende wachtwoord uit de Database.

Als laatst krijg de gebruiker nog een mail dat zijn gegevens zijn aangepast met een link erin mocht hij/zij dit niet hebben gedaan. Deze link stuurt ze naar het contactformulier.

**Wachtwoord wijzigen**

Om je wachtwoord te kunnen wijzigen moet je drie dingen invullen :

* Oude wachtwoord 🡪 deze moet overeen komen met je eerder ingevulde wachtwoord, is dit niet het geval dan zal je een foutcode krijgen.
* Nieuwe wachtwoord 🡪 dit is je nieuwe wachtwoord, deze moet overeen komen met de volgende punt
* Nieuwe wachtwoord herhalen 🡪 dit is je nieuwe wachtwoord maar hier wordt een check uitgevoerd om te kijken of die overeen komt. Als die niet overeen komt zal je weer een foutmelding krijgen. Klopt het wel dan krijg je een mail.

### 3.5.4. Pseudo code

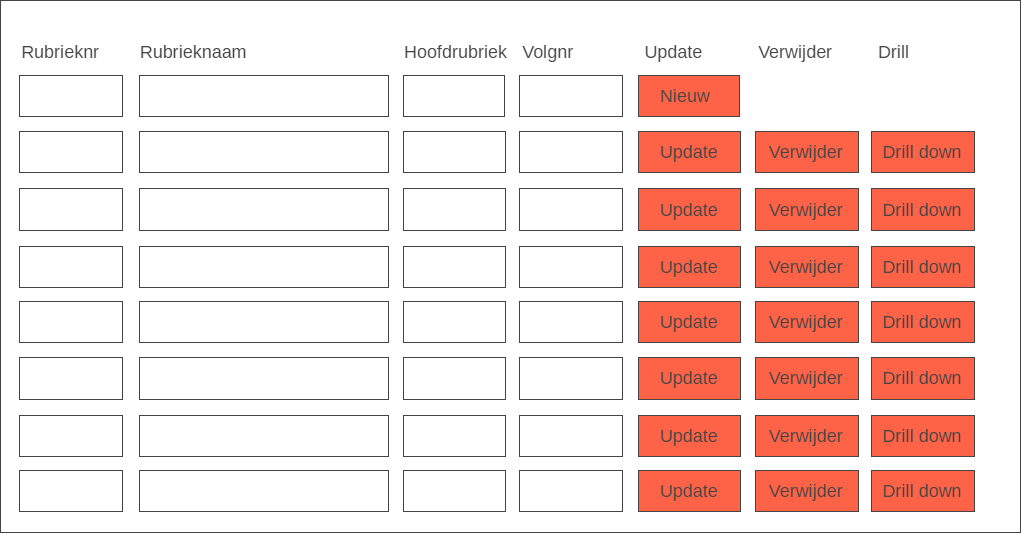
### 3.5.5. Activity Diagram

### 3.5.6. Testcases

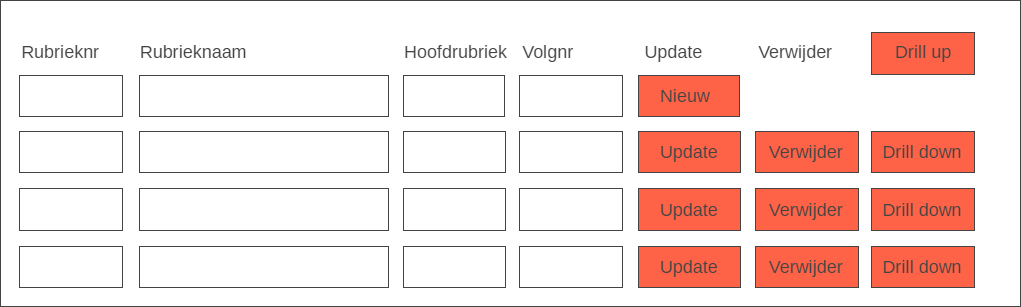
## 3.6. User story 6 - Als beheerder wil ik de rubriekenboom kunnen beheren door het toevoegen, verwijderen, sorteren, hernoemen en/of uitfaseren van rubriekenboom.

### 3.6.1. Wireframe’s

**Wireframe – pagina waar rubrieken toegevoegd en aangepast kunnen worden**Op deze pagina kunnen nieuwe rubrieken gemaakt worden en bestaande rubrieken geüpdatet worden. De eerste rij is voor het maken van een nieuwe rubriek en de rest van de rijen laten rubrieken zien.



**Wireframe – Drill down**Als op “Drill down” geklikt is kom je op dezelfde pagina terecht maar nu staan de subrubrieken van de gekozen rubriek in de lijst, ook staat er nu een “drill up” knop. Met deze knop word je weer terug naar de vorige pagina gebracht. Deze pagina functioneert verder precies hetzelfde als voorheen.



### 3.6.2. Functioneel ontwerp

Via deze pagina kan er een nieuwe rubriek gemaakt worden, een rubriek geüpdatet worden of een rubriek verwijderd worden.

Om een nieuwe rubriek te maken moet de bovenste rij ingevuld worden en moet er op de knop “Nieuw” geklikt worden.  
Alle rijen daar onder zijn automatisch opgevuld met de rubrieken uit de database, om deze aan te passen moet er iets anders ingetypt worden en op “Update” geklikt worden.  
Een rubriek kan verwijderd worden met de “Verwijder” knop.  
Met de “Drill down” knop worden de subrubrieken van de geselecteerde rubriek opgevraagd.

### 3.6.3. Technisch ontwerp

Bij alle acties de uitgevoerd worden wordt er eerst gekeken of alle velden ingevuld zijn, als dit niet zo is word je teruggestuurd.

Als op “Nieuw” wordt geklikt worden er een aantal checks uitgevoerd voordat de rubriek toegevoegd wordt aan de database. Eerst wordt gekeken of het rubrieknummer al bestaat, daarna of de naam al bestaat bij het gekozen hoofdrubriek en vervolgens of de verwijzing naar die hoofdrubriek wel bestaat. Als hij door alle checks heen komt wordt de rubriek toegevoegd.

Bij het updaten van rubrieken worden al deze checks ook uitgevoerd. Als de update door de checks heen komt, wordt de rubriek geüpdatet. Daarna worden alle subrubrieken hiervan geüpdatet zodat hun hoofdrubriek overeenkomt met het nieuwe rubrieknummer, bijvoorbeeld: als rubrieknummer 1 was maar naar 2 is veranderd, dan worden alle rubrieken waar een 1 in de kolom “hoofdrubriek” stond ook geüpdatet en wordt daar een 2 van gemaakt.

Bij verwijderen wordt eerst gekeken of de rubriek subrubrieken heeft, als dit zo is wordt hij niet verwijderd. Heeft hij geen subrubrieken dan wordt er gekeken of het een van de rubrieken op hoogste niveau is (deze hebben in de kolom hoofdrubriek -1 staan), deze rubrieken mogen geen producten bevatten als je ze wilt verwijderen. Als hij geen producten bevat kan hij verwijderd worden, bevat hij wel producten dan wordt er een foutmelding weergeven.   
Als een andere rubriek verwijderd wordt, worden alle producten die hierin zaten verplaatst naar de hoofdrubriek hiervan, vervolgens wordt de rubriek zelf verwijderd.

### 3.6.4. Pseudo code

<body>

//nieuwe rubriek

<form>

<input name='Rubrieknummer'>

<input name='Rubrieknaam'>

<input name='Hoofdrubriek'>

<input name='Volgnr'>

<input type='submit' value=’Nieuw’>

</form>

foreach (alle rubrieken) {

<form>

//waardes laten vullen uit de database

<input name='Rubrieknummer' value='$rubriek['Rubrieknummer']'>

<input name='Rubrieknaam' value='$rubriek['Rubrieknaam']'>

<input name='Hoofdrubriek' value='$rubriek['Hoofdrubriek']'>

<input name='Volgnr' value='$rubriek['Volgnr']'>

//leest de waarde die zijn ingevuld en voert een update uit

<input type='submit' value=’Update’>

//verwijdert de rubriek

<input type='submit' value=’Verwijder’>

//haalt de subrubrieken van de aangeklikte rubriek op

<input type='submit' value=’Drill down’>

</form>

}

</body>

<?php

if (een input veld is leeg) {

Doe niks;

} else if ($\_POST['action'] == 'Update') {

if (rubrieknr is niet in gebruik || rubrieknaam is niet gebruikt binnen deze hoofdrubriek) {

Voer update uit;

Update de subrubrieken;

} else {

Geef error;

}

} else if ($\_POST['action'] == 'Verwijder') {

if (rubriek heeft geen subrubrieken) {

Verplaats alle voorwerpen in de rubriek naar hun hoofdrubriek;

Verwijder rubriek;

} else {

Geef error;

}

} else if ($\_POST['action'] == 'Drill down') {

header("Location: ../CategorieUpdate.php?cat=$originalrub");

} else if ($\_POST['action'] == 'Nieuw') {

if (rubrieknr is niet in gebruik || rubrieknaam is niet gebruikt binnen deze hoofdrubriek) {

Maakt nieuwe rubriek aan;

} else {

Geef error;

}

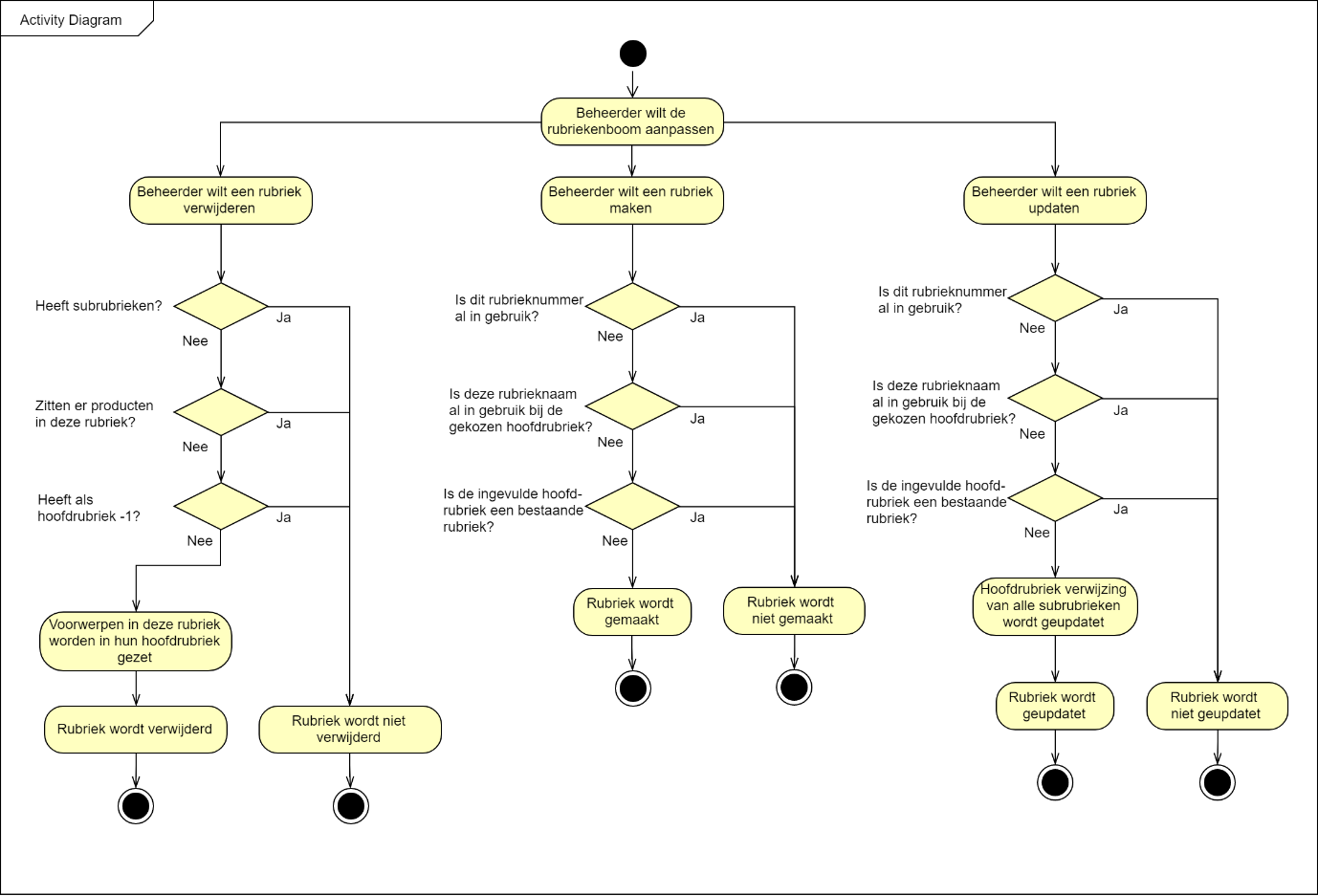
} else {

Doe niks;

}

?>

### 3.6.5. Activity Diagram



### 3.6.6. Testcases

**Rubrieken updaten**

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Rubrieknummer veranderen | Rubrieknr: 2  Rubrieknaam: <laten staan>  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | Het rubrieknummer wordt geüpdatet, alle rubrieken die onder de rubriek stonden zijn ook geüpdatet aan hun Hoofdrubriek kolom. Alle voorwerpen in de tabel voorwerpen\_in\_rubriek worden geüpdatet naar het nieuwe nummer. | Het rubrieknummer wordt geüpdatet, alle rubrieken die onder de rubriek stonden zijn ook geüpdatet aan hun Hoofdrubriek kolom. Alle voorwerpen in de tabel voorwerpen\_in\_rubriek worden geüpdatet naar het nieuwe nummer. |
| Rubrieknummer veranderen in iets dat al bestaat | Rubrieknr: 1  Rubrieknaam: <laten staan>  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | De update wordt niet uitgevoerd en de fout wordt afgehandeld door een foutmelding te laten zien. | De update wordt niet uitgevoerd en de fout wordt afgehandeld door een foutmelding te laten zien. |
| String invullen in rubrieknummer | Rubrieknr: eaefsef  Rubrieknaam: <laten staan>  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | Textvak staat alleen nummers toe maar door de html te veranderen kan een string ingevuld worden, dit wordt ook niet in de database geüpdatet | Krijgt een PDO crash pagina en de update wordt niet uitgevoerd. |
| Groot getal invullen in rubrieknummer | Rubrieknr: 2147483660  Rubrieknaam: <laten staan>  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | Getal is groter dan in een int past en wordt dus niet toegestaan bij het invullen, het textvak laat standaard niet zulke grote getallen toe | Krijgt een PDO overflow error, de update wordt niet uitgevoerd. |
| Rubrieknaam veranderen | Rubrieknr: <laten staan>  Rubrieknaam: Verzamelen2  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | De naam wordt geüpdatet | De naam wordt geüpdatet |
| Rubrieknaam veranderen in iets dat al bestaat | Rubrieknr: <laten staan>  Rubrieknaam: Verzamelen  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | Wordt toegestaan als de naam gebruikt word onder een andere hoofdcategorie, anders komt er een foutmelding. | Wordt toegestaan als de naam gebruikt word onder een andere hoofdcategorie, anders komt er een foutmelding. |
| Rubrieknaam veranderen in iets dat te lang is | Rubrieknr: <laten staan>  Rubrieknaam: aaaaaaaaaaaaaaaaaaaaaaaaa aaaaaaaaaaaaaaaaaaaaaaaaaa  (51 symbolen)  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | Textvak staat een maximum van 50 characters toe (limiet van de kolom in de database), meer dan dit zal dus niet in de database | Krijgt een PDO crash pagina en de update wordt niet uitgevoerd. |
| Hoofdrubriek veranderen | Rubrieknr: <laten staan>  Rubrieknaam: <laten staan>  Hoofdrubriek: 160  Volgnr: <laten staan> | Rubriek wordt geüpdatet en verplaatst naar zijn nieuwe hoofdrubriek | Rubriek wordt Geüpdatet en verplaatst naar zijn nieuwe hoofdrubriek |
| Hoofdrubriek veranderen in een rubrieknummer dat niet bestaat | Rubrieknr: <laten staan>  Rubrieknaam: <laten staan>  Hoofdrubriek: 4  Volgnr: <laten staan> | Rubriek wordt niet geüpdatet en je krijgt een foutmelding | Rubriek wordt niet geüpdatet en je krijgt een foutmelding |
| String invullen in hoofdrubriek | Rubrieknr: <laten staan>  Rubrieknaam: <laten staan>  Hoofdrubriek: fasdfj  Volgnr: <laten staan> | Textvak staat alleen nummers toe maar door de html te veranderen kan een string ingevuld worden, dit wordt ook niet in de database geüpdatet | Krijgt een PDO crash pagina en de update wordt niet uitgevoerd. |
| Volgnr veranderen | Rubrieknr: <laten staan>  Rubrieknaam: <laten staan>  Hoofdrubriek: <laten staan>  Volgnr: 3 | Volgnr verandert | Volgnr verandert |
| String invullen in hoofdrubriek | Rubrieknr: <laten staan>  Rubrieknaam: <laten staan>  Hoofdrubriek: <laten staan>  Volgnr: kjhlakhf | Textvak staat alleen nummers toe maar door de html te veranderen kan een string ingevuld worden, dit wordt ook niet in de database geüpdatet | Krijgt een PDO crash pagina en de update wordt niet uitgevoerd. |
| Alles tegelijk veranderen | Rubrieknr: 3  Rubrieknaam: Postzegels1  Hoofdrubriek: 1  Volgnr: 2 | De update veranderd succesvol alle waardes | De update veranderd succesvol alle waardes |
| Niks veranderen | Rubrieknr: <laten staan>  Rubrieknaam: <laten staan>  Hoofdrubriek: <laten staan>  Volgnr: <laten staan> | Er gaat niks fout en er verandert niks | Er gaat niks fout en er verandert niks |
| Velden leeg laten | Rubrieknr: <laten staan>  Rubrieknaam:  Hoofdrubriek: <laten staan>  Volgnr: | Er wordt gekeken of alles is ingevuld, zo niet krijg je hier een foutmelding over en verandert er niks | Er wordt gekeken of alles is ingevuld, zo niet krijg je hier een foutmelding over en verandert er niks |

**Rubrieken verwijderen**

|  |  |  |
| --- | --- | --- |
| Test | Verwacht resultaat | Werkelijk resultaat |
| Rubriek dat geen subrubrieken heeft verwijderen | De rubriek wordt verwijderd | De rubriek wordt verwijderd |
| Rubriek dat subrubrieken heeft verwijderen | De rubriek wordt niet verwijderd en er komt een foutmelding | De rubriek wordt niet verwijderd en er komt een foutmelding |
| Rubriek verwijderen die geen subrubrieken heeft maar wel producten in die rubriek | De rubriek wordt verwijderd en alle producten in die rubriek worden in hun hoofdcategorie gezet | De rubriek wordt verwijderd en alle producten in die rubriek worden in hun hoofdcategorie gezet |
| Rubriek verwijderen die wel subrubrieken heeft en producten in die rubriek | De rubriek wordt niet verwijderd en er komt een foutmelding | De rubriek wordt niet verwijderd en er komt een foutmelding |
| Rubriek met als hoofdrubriek -1 verwijderen als er geen producten in zitten en geen subrubrieken | De rubriek wordt verwijderd | De rubriek wordt verwijderd |
| Rubriek met als hoofdrubriek -1 verwijderen als er producten in zitten maar geen subrubrieken | De rubriek wordt niet verwijderd om te voorkomen dat de vorwerpen in de categorie ROOT terecht komen | De rubriek wordt niet verwijderd om te voorkomen dat de vorwerpen in de categorie ROOT terecht komen |

**Rubrieken toevoegen**

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Rubriek maken | Rubrieknr: 2  Rubrieknaam: test  Hoofdrubriek: -1  Volgnr: 1 | De rubriek wordt toegevoegd | De rubriek wordt toegevoegd |
| Rubriek maken met rubrieknummer dat al bestaat | Rubrieknr: 1  Rubrieknaam: test2  Hoofdrubriek: -1  Volgnr: 1 | De rubriek wordt niet gemaakt en de fout wordt afgehandeld door een foutmelding te laten zien. | De rubriek wordt niet gemaakt en de fout wordt afgehandeld door een foutmelding te laten zien. |
| Rubriek maken met rubrieknaam die al bestaat | Rubrieknr: 3  Rubrieknaam: Verzamelen  Hoofdrubriek: -1  Volgnr: 1 | Wordt toegestaan als de naam gebruikt word onder een andere hoofdcategorie, anders komt er een foutmelding. | Wordt toegestaan als de naam gebruikt word onder een andere hoofdcategorie, anders komt er een foutmelding. |
| Rubriek maken met hoofdrubriek die niet bestaat | Rubrieknr: 4  Rubrieknaam: t  Hoofdrubriek: 5  Volgnr: 1 | Rubriek wordt niet gemaakt en je krijgt een foutmelding | Rubriek wordt niet gemaakt en je krijgt een foutmelding |
| Velden leeg laten | Rubrieknr:  Rubrieknaam: asdf  Hoofdrubriek: -1  Volgnr: | Er wordt gekeken of alles is ingevuld, zo niet krijg je hier een foutmelding over en verandert er niks | Er wordt gekeken of alles is ingevuld, zo niet krijg je hier een foutmelding over en verandert er niks |

## 3.7. User story 7 - Als gebruiker wil ik op de hoogte worden gehouden van de status van een veiling, zodat ik eventueel naar vergelijkbare voorwerpen op zoek kan gaan.

### 3.7.1. Wireframe’s

Geen beschikbaar

### 3.7.2. Functioneel ontwerp

De gebruiker wordt via de mail op de hoogte gehouden, dit gebeurt elke keer dat er geboden wordt.

### 3.7.3. Technisch ontwerp

Zodra een bod geplaatst wordt, word er gekeken wie eerst het hoogste bod had. Deze persoon ontvangt een mail waarin hem verteld wordt dat zijn bod overschreden is. In deze mail staat de volgende informatie:

* De titel van de veiling
* Zijn bod
* Het nieuwe bod
* Een link naar de veiling

### 3.7.4. Pseudo code

if (Bod geplaatst) {

if (een na hoogste bod geplaatst door een andere gebruiker dan hoogste bod) {

Haal de een na hoogste bieder op;

Haal titel voorwerp op;

Haal zijn geboden bedrag op;

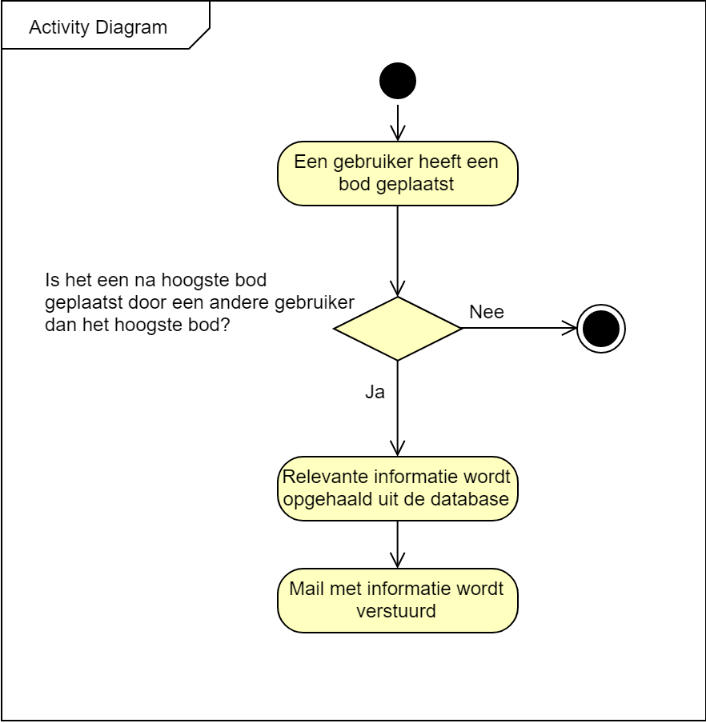
Haal het nieuwe bod op;

Verstuur een mail naar de een na hoogste bieder waarin deze informatie staat;

}

}

### 3.7.5. Activity Diagram



### 3.7.6. Testcases

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Input | Verwacht resultaat | Werkelijk resultaat |
| Over iemand bieden | Bij een product een bod plaatsen onder een account dat op dat moment **niet** het hoogste bod heeft. | Het bod wordt geplaatst en degene die daarvoor het hoogste bod had krijgt een mail waarin staat dat zijn bod overtroffen is. In de mail staat het product, zijn oude bod en het nieuwe bod met een link naar het product. | Het bod wordt geplaatst en degene die daarvoor het hoogste bod had krijgt een mail waarin staat dat zijn bod overtroffen is. In de mail staat het product, zijn oude bod en het nieuwe bod met een link naar het product. |
| Over jezelf heen bieden | Bij een product een bod plaatsen onder een account dat op dat moment **wel** het hoogste bod heeft. | Het bod wordt geplaatst maar er wordt geen mail verstuurd. | Het bod wordt geplaatst maar er wordt geen mail verstuurd. |

## 3.8. User story 8 - Als gebruiker wil ik een nieuw wachtwoord kunnen opgeven wanneer ik het wachtwoord ben vergeten.

### 3.8.1. Wireframe’s

### 3.8.2. Functioneel ontwerp

### 3.8.3. Technisch ontwerp

### 3.8.4. Pseudo code

### 3.8.5. Activity Diagram

### 3.8.6. Testcases