Functioneel Ontwerp

Planning Poker

Versie 1.0

Timme Kingma s11  
2025

# Distributie

|  |  |  |  |
| --- | --- | --- | --- |
| Versie | Datum | Wijzigingen | Ontvangers |

# Inhoudsopgaven

[Distributie 2](#_Toc190956603)

[Inhoudsopgaven 3](#_Toc190956604)

[Inleiding 4](#_Toc190956605)

[1 Domeinanalyse 5](#_Toc190956606)

[1.1 Huidige situatie 5](#_Toc190956607)

[1.2 Gewenste situatie 5](#_Toc190956608)

[1.3 Entiteiten en relaties Domein model 6](#_Toc190956609)

[1.4 Use case diagram 6](#_Toc190956610)

[1.5 Entiteiten en relaties Use case diagram 7](#_Toc190956611)

[2 Users stories 8](#_Toc190956612)

[2.1 Als Scrum Master wil ik 8](#_Toc190956613)

[2.2 Als Developer wil ik 8](#_Toc190956614)

[3 Use cases 9](#_Toc190956615)

[3.1 Toelichting Betrokken requirements Asset/Type 9](#_Toc190956616)

[3.2 Toelichting Betrokken requirements MoSCoW 9](#_Toc190956617)

[3.3 Use Case: Bekijken Home pagina 10](#_Toc190956618)

[3.4 Use Case: Accountcreatie 12](#_Toc190956619)

[3.5 Use Case: Game starten 15](#_Toc190956620)

[3.6 Use Case: Game joinen 17](#_Toc190956621)

[3.7 Use Case: Game spelen 20](#_Toc190956622)

[3.8 Use Case: Aanmaken van Back log Item’s 23](#_Toc190956623)

[3.9 Use Case: naam 27](#_Toc190956624)

[Bijlage 1 Aanpak Functioneel Ontwerp 28](#_Toc190956625)

[3.10 Domein 28](#_Toc190956626)

[3.11 Use case diagram 29](#_Toc190956627)

[3.12 Use cases 29](#_Toc190956628)

# Inleiding

Dit document beschrijft het functionele ontwerp van de webapplicatie Planning Poker. Het ontwerp is gebaseerd op een uitgebreide requirementsanalyse en richt zich op het ontwikkelen van een veilige, intuïtieve en gebruiksvriendelijke applicatie. Het doel van dit project is om teams effectief te ondersteunen bij het schatten van backlog items door middel van een interactieve en gestructureerde Planning Poker-game.

# Domeinanalyse

In dit hoofdstuk wordt het domein beschreven op basis van eigen ervaring. Het domein sluit aan bij de opgestelde requirements. Om het domein inzichtelijk te maken, is gebruikgemaakt van een domeinmodel en een use-case diagram. Dit hoofdstuk begint met de huidige situatie, gevolgd door een beschrijving van de gewenste situatie. Daarna wordt een domein-/use-case diagram gepresenteerd met toelichting.

## Huidige situatie

In de huidige webapplicaties kunnen developers al planning poker spelen, maar er ontbreken enkele functionaliteiten. De betere functies zitten achter een paywall, en daarnaast ontbreken enkele gewenste features voor student Timme Kingma. Zo is er geen manier om inzicht te krijgen in eerder gepokerde items, en ontbreekt een verbeterde methode voor het toekennen van punten. Zie IV1 in de requirementsanalyse voor meer informatie.

## Gewenste situatie

In de gewenste situatie kunnen developers, zoals een werkend agile-team, een planning poker-spel spelen waarbij ze verschillende backlogitems storypoints kunnen toekennen. Het proces moet soepel en flexibel zijn, zodat developers hun eigen werkwijze kunnen hanteren. Ze moeten de reeks storypoints kunnen aanpassen aan hun voorkeur, een timer in- of uitschakelen en het spel starten of eraan deelnemen via een veilig authenticatieproces. Dit kan bijvoorbeeld met e-mailverificatie of door in te loggen met een bestaand account, zodat alleen geautoriseerde gebruikers toegang hebben. Hierdoor ontstaat een efficiëntere en gebruiksvriendelijkere applicatie die aansluit bij de behoeften van het team. Hieronder staat het domeinmodel van de applicatie in de gewenste situatie.

Afbeelding met tekst, schermopname, Lettertype, diagram

Automatisch gegenereerde beschrijving

Figure Domein model Planing poker

## Entiteiten en relaties Domein model

Hieronder een tabel met alle entiteiten binnen de applicatie kort toegelicht. De naam van de bijbehorende klasse, die terugkomt in het domeinmodel, wordt ook vermeld.

|  |  |  |
| --- | --- | --- |
| Entiteit | Klassenaam | Toelichting |
| **Developer/**  **Scrummaster** | **User** | Een developer of scrummaster van de applicatie. |
| **2 Authentication** | **2 Authentication** | Een beveiliging code die gebruikt word om als gebruiker zich te identificeren. |
| **BacklogItem** | **BacklogItem** | Een scrum artifact wat een taak voor beeld. |
| **Planning poker spel** | ***Spel*** | Het spel wat gespeeld gaat worden en bijbehorende settings. |

Tabel Tabel toelichting domein model

## Use case diagram

In dit hoofdstuk wordt het use-case diagram gepresenteerd. Het diagram geeft een overzicht van alle use cases en biedt inzicht in de verschillende actoren die betrokken zijn bij het systeem. Hieronder staat een figuur van het use-case diagram, gevolgd door tabellen met verdere toelichting.

Afbeelding met tekst, diagram, schermopname, lijn

Automatisch gegenereerde beschrijving

Figure 2 Use case diagram Planing poker

## Entiteiten en relaties Use case diagram

Hieronder twee tabellen met alle entiteiten binnen de applicatie kort toegelicht. De naam van de bijbehorende use case, die terugkomt in het use case diagram, wordt ook vermeld. De tabellen zijn opgesplitst in twee delen actoren en use cases.

|  |  |
| --- | --- |
| Actor | Toelichting |
| **Scrum master** | Een leider van een groep hij zorgt er voor dat een spel goed verloopt en ook ingesteld is volgens zijn wensen. |
| **Developer** | de developer speelt een spel onder leiding van de scrum master. |
| **Extern email systeem** | Een API systeem die de authenticatie code kan versturen naar developers. |

Tabel Toelichting actoren Use case diagram

|  |  |
| --- | --- |
| Use case | Toelichting |
| **Bekijken home pagina** | Een use case waarin de hoofdpagina bekeken gaat worden. |
| **Account maken** | Een use case waarin de developer een acount aan maakt. |
| **Game joinen** | Een use case waarin developers een game joined. |
| **Game starten** | Een use case waarin de game aangemaakt en gestart word. |
| **Aanmaken van Back log Item’s** | Een use case waarin de Scrum master backlog items aanmaakt zodat deze in het spel gespeelt word. |
| **Game spelen** | Een use case waarin het spel gespeeld word. |

Tabel 3 Toelcihting use cases Use case diagram

# Users stories

In dit hoofdstuk worden de user stories beschreven. Hierdoor wordt een duidelijk overzicht gegeven van de functionaliteiten die de applicatie moet bieden.

## Als Scrum Master wil ik

* Een game kunnen starten om poker te spelen.
* Mensen kunnen uitnodigen om poker te spelen.
* Eén of meerdere backlogitems kunnen verzamelen en in de juiste volgorde tonen.
* De game kunnen leiden.

## Als Developer wil ik

* Een game kunnen joinen.
* Punten kunnen geven aan een backlogitem.
* Kunnen inloggen.
* Me kunnen registreren.
* Informatie over de applicatie kunnen lezen.
* Een andere naam dan mijn eigen naam kunnen invoeren.
* Automatisch ingelogd kunnen blijven zonder steeds mijn naam in te voeren.

# Use cases

Dit hoofdstuk beschrijft de use cases vanuit functioneel perspectief.

## Toelichting Betrokken requirements Asset/Type

|  |  |
| --- | --- |
| Asset/Type | Toelichting |
| Functioneel | Beschrijft de gewenste functionaliteit van het systeem |
| Beperking | Geeft aan welke beperkingen of randvoorwaarden gelden |
| Security | Richt zich op beveiligingsmaatregelen. |
| Validatie | Bevat regels voor het valideren van invoervelden. |
| kwaliteit | Omvat technische en prestatienormen. |
| UX | UX staat voor user experience en richt op wat fijn werkt/voelt voor een bezoeker |

Tabel Toelichting assets/type

## Toelichting Betrokken requirements MoSCoW

|  |  |
| --- | --- |
| definitie | Toelichting |
| **M (Must)** | **Verplichte onderdelen. Zonder deze vereisten kan het project niet als succesvol worden beschouwd.** |
| **S (Should)** | **Belangrijke onderdelen die niet essentieel zijn, maar een aanzienlijke toegevoegde waarde bieden als ze worden geïmplementeerd.** |
| **C (Could)** | **Wenselijke onderdelen die leuk zijn om te hebben, maar kunnen worden uitgesteld zonder grote impact.** |
| **W (Won't)** | **Zaken die voorlopig niet worden geïmplementeerd, mogelijk wel in toekomstige iteraties.** |

Tabel Toelichting moscow

## Toelichting Actoren

|  |  |
| --- | --- |
| Actor | Toelichting |
| **Developer** | **Een developer is een bezoeker die de website bezoekt en veder geen specifieke rol heeft een developer kan een spel spelen en inloggen.** |
| **Scrum master** | **Een scrum master is een groepsleider de scrum master maakt een game lobby aan en ook backlog items hij kan ook de ronde beindigen.** |

## Use Case: Bekijken Home pagina

|  |  |
| --- | --- |
| Use case naam | Bekijken home pagina |
| Actoren | Scrum master, Developer |
| Doel | Een developer kan de hoofdpagina bekijken en zich oriënteren over de het planning poker. |
| Precondities | Leeg |
| Hoofdscenario | 1. De developer bezoekt de Home |
| Alternatieve scenario’s | 1.1 De website is niet geschikbaar |
| Postcondities | De website is inzichtbaar. |

Tabel UseCase bekijken home pagina

### Betrokken requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beschrijving | Asset/Type | MoSCoW | Testen |
| FR1 | De homepagina moet toegankelijk zijn voor iedereen. | Functioneel | Must |  |
| FR2 | De homepagina moet informatie tonen over het planning poker. | Functioneel | Must |  |
| FR3 | De gebruikte HTML-tags zijn semantisch waar dit mogelijk is. | Kwaliteit | Should |  |
| FR4 | Het design is Mobile First. | UX | Must |  |
| NFR1 | Het design is Responsive. | UX | Must |  |
| NFR2 | de profielpagina bevat een GDPR (ASVS V8.3 Sensitive Private Data). | Kwaliteit | Must |  |
| NFR3 | De GDPR-keuze wordt opgeslagen (cookie of localStorage). | Kwaliteit | Must |  |
| NFR4 | GDPR wordt alleen getoond als er geen consent is gegeven. | Kwaliteit | Must |  |
| NFR5 | Styling van GDPR past bij de rest van de pagina. | UX | Must |  |
| NFR8 | De gegevens op de pagina zijn niet via de webpagina te beheren. | Beperking | Must |  |
| NFR9 | De pagina wordt binnen 1 seconde geladen. | Kwaliteit | Must |  |

Tabel Betrokken requirments home pagina

### Schermontwerp

Afbeelding met tekst, schermopname, Lettertype, ontwerp

Automatisch gegenereerde beschrijving

Figure Shermontwerp hoofdpagina

### Evil user stories

|  |  |  |
| --- | --- | --- |
| # | Evil user story | Mitigatie |
| EUS1 | Als aanvaller wil ik toegang krijgen tot de cookievoorkeuren van gebruikers, zodat ik de sessie kan overnemen. | SM1,SM2 |
| EUS2 | Als aanvaller wil ik kwaadaardige scripts injecteren in de homepagina, zodat ik gebruikers kan misleiden of gegevens kan stelen. | SM3 |

Tabel 8 Evil user stories home pagina

## Use Case: Accountcreatie

|  |  |
| --- | --- |
| Use case naam | Accountcreatie |
| Actoren | Developer |
| Doel | Een nieuw account te maken en in te loggen |
| Precondities | leeg |
| Hoofdscenario | 1. De developer maakt een account aan  2. De developer verifieerd zijn account via email authenticatie.  3. De developer logt in met zijn nieuwe account  4 De sessie word opgeslagen |
| Alternatieve scenario’s | 2.1 Het acount word niet geauthentificeerd.  3.1 De developer heeft zijn acount vergeten of niet geverifieerd  De developer kan bij het inloggen alsnog authenticeren. |
| Postcondities | Er is een sessie/cookie opgeslagen waarin de developer niet weer hoeft inte loggen pas na 15 dagen.  De gebruiker heeft een acount  De gebruiker kan inloggen en word developer. |

Tabel Use case Acountcreatie

### Betrokken requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beschrijving | Asset/Type | MoSCoW | Testen |
| FR1 | Een developer kan een account aanmaken. | Functionaliteit | Must |  |
| FR2 | De developer moet zijn account verifiëren via e-mailauthenticatie. | Functionaliteit | Must |  |
| FR3 | De developer kan inloggen met zijn nieuwe account. | Functionaliteit | Must |  |
| NFR1 | Verificatie via e-mail is vereist voor accountactivatie. | Security | Must |  |
| NFR2 | De sessie/cookie wordt opgeslagen voor 15 dagen. | Security | Should have |  |
| NFR3 | De authenticatie voldoet aan best practices zoals hashing en beveiligde sessies. | Security | Must |  |
| NFR4 | Alleen veilige wachtwoorden worden geaccepteerd (min. 8 tekens, cijfers, speciale tekens). | Validatie | Must |  |
| NFR5 | Foutmeldingen mogen geen gevoelige informatie lekken. | Security | Must |  |
| NFR6 | Email word gevalideerd op een geldig adress. | Validatie | Must |  |

Tabel Betrokken requirments Acountcreatie

### Schermontwerp

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Figure Registratie Schermontwerp Figure Login Schermontwerp

### Evil user stories

|  |  |  |
| --- | --- | --- |
| # | Evil user story | Mitigatie |
| EUS1 | Een aanvaller probeert een account aan te maken met een bestaand e-mailadres | SM4 |
| EUS2 | Een aanvaller probeert brute-force logins | SM3 |
| EUS3 | Een aanvaller probeert de sessie te kapen | SM2 |
| EUS4 | Een aanvaller probeert met bots de servers te overflowen door meerdere malen te registreren | SM3 |
| EUS5 | Een aanvaller probeert wachtwoorden van gebruikers te achterhalen via databaselekken. | SM1 |

Tabel 11 Evil user stories Acountcreatie

### Activity diagram

Hieronder een activity diagram om de functionele flow duidelijk te maken.

Afbeelding met tekst, schermopname, lijn, diagram

Automatisch gegenereerde beschrijving

Figure Activity diagram Acountcreatie

## Use Case: Game starten

|  |  |
| --- | --- |
| Use case naam | Game starten |
| Actoren | Scrum Master |
| Doel | Een game starten om planning poker te spelen |
| Precondities | Scrum master is ingelogd |
| Hoofdscenario | 1. Scrum Master navigeert naar de game-startpagina. 2. Scrum Master selecteert de optie om een nieuwe game te starten. 3. Scrum Master vult de game-instellingen in (zoals naam van de game, timerinstellingen, en type stemmen). 4. Scrum Master bevestigt de instellingen en start de game. 5. De game wordt gecreëerd en zichtbaar voor uitgenodigde deelnemers. 6. Na het joinen van deelnemers start de Scrum master de game. |
| Alternatieve scenario’s | 2.1 Scrum Master besluit halverwege het instellen van de game om te annuleren.  2.2 Het systeem sluit het instellingsscherm zonder een game te starten. |
| Postcondities | * De game is succesvol gestart en klaar voor deelname. * Spelers kunnen deelnemen op basis van uitnodigingen of een gamecode. * Game is gestart en kan gespeeld worden. |

Tabel Use case Game starten

### Betrokken requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beschrijving | Asset/Type | MoSCoW | Testen |
| FR1 | De spel leider moet een game kunnen starten. | Functionaliteit | Must |  |
| FR2 | De spel leider moet de game-instellingen kunnen configureren (zoals naam, timer en type stemmen). | Functionaliteit | Should have |  |
| Fr3 | De game moet een unieke ID genereren bij het aanmaken. | Functionaliteit | Must |  |
| FR4 | Spelers moeten kunnen deelnemen via een unieke link en gamecode. | Functionaliteit | Must |  |
| FR5 | De spel leider moet de game kunnen annuleren voordat deze wordt gestart. | Functionaliteit | Should have |  |
| NFR1 | De game moet real-time updates ondersteunen voor deelnemers. | Functionaliteit | Must |  |
| NFR2 | De gegenereerde game-ID’s moeten veilig en niet voorspelbaar zijn. | Security | Should have |  |
| NFR3 | De game moet schaalbaar zijn voor meerdere gelijktijdige games. | Kwaliteit | Should have |  |

Tabel Betrokken requirements Game starten

### Schermontwerp

Afbeelding met tekst, schermopname, software, Website

Automatisch gegenereerde beschrijvingAfbeelding met tekst, schermopname, Rechthoek, ontwerp

Automatisch gegenereerde beschrijving

Figure Game starten schermontwerp Figure Game join schermontwerp

### Evil user stories

|  |  |  |
| --- | --- | --- |
| # | Evil user story | Mitigatie |
| EUS1 | Een niet-ingelogde gebruiker probeert een game te starten. | SM1 |
| EUS2 | Een kwaadwillende gebruiker start meerdere games om het systeem te overbelasten | SM3 |
| EUS3 | Een aanvaller probeert gamecodes te raden om deel te nemen aan privé-games. | SM2 |
| EUS4 | Een aanvaller onderschept of manipuleert gamecodes of tokens in de URL. | SM2 |

Tabel Evil user stories Game starten

### Activity diagram

Hieronder een activity diagram om de functionele flow duidelijk te maken.

Afbeelding met tekst, Lettertype, schermopname, lijn

Automatisch gegenereerde beschrijving

Figure 9 Activity diagram Game starten

## Use Case: Game joinen

|  |  |
| --- | --- |
| Use case naam | Game joinen |
| Actoren | Developer |
| Doel | Een planning poker game te kunnen joinen |
| Precondities | Er is een game aangemaakt en de developer heeft een code gekregen om zichzelf te kunnen authentiseren en ook een link om te kunnen joinen. |
| Hoofdscenario | 1. De developer gebruikt de link om op de juiste pagina te komen  2. De developer voert de een eigen naam in of logt in  3. De developer verifieert zichzelf met de bijgegeven code  4. de developer joined de game |
| Alternatieve scenario’s | 2.1 Er komt een error met het inloggen van de developer  De developer moet opnieuw proberen  4.1 de game bestaat niet meer  Er moet een nieuwe game gemaakt worden |
| Postcondities | De developer joind een game |

Tabel UseCase Game joinen

### Betrokken requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beschrijving | Asset/Type | MoSCoW | Testen |
| FR1 | Een speler kan via een link op de juiste gamepagina komen. | Functionaliteit | Must |  |
| FR2 | Een speler kan een eigen naam invoeren of inloggen. | Functionaliteit | Should have |  |
| FR3 | Een speler developer moet een geldige code invoeren om zichzelf te verifiëren. | Security | Must |  |
| FR4 | Een speler kan succesvol een game joinen. | Functionaliteit | Must |  |
| FR5 | Er wordt een foutmelding weergegeven als de game niet meer bestaat. | UX | Should have |  |
| NFR1 | De verificatiecode moet uniek en moeilijk te raden zijn. | Security | Must |  |
| NFR2 | de gamepagina moet binnen 2 seconden laden.(delay mag door realtime sync) | Kwaliteit | Could have |  |
| NFR3 | De UI moet duidelijke foutmeldingen tonen bij een mislukte login of ongeldig spel. | UX | Should have |  |
| NFR4 | De game moet real-time updates ontvangen zodra de developer succesvol joined. | Functionaliteit | Must |  |
| NFR5 | Er mogen geen gevoelige gegevens in de URL worden opgeslagen. | Security | Must |  |
| NFR6 | Als er twee mensen met de zelfde naam zijn moeten ze in de UI verschillend zijn door kleur of icon. | UX | Would have |  |

Tabel Betrokken requirments Game joinen

### Schermontwerp

Afbeelding met tekst, schermopname, Rechthoek, ontwerp

Automatisch gegenereerde beschrijving

Figure Game join schermontwerp

### Evil user stories

|  |  |  |
| --- | --- | --- |
| # | Evil user story | Mitigatie |
| UES1 | Een aanvaller probeert zonder geldige code een game te joinen | SM1,SM5 |
| UES2 | Een kwaadwillende gebruiker probeert meerdere keren een foute code in te voeren | SM6 |
| UES3 | Een kwaadwillende gebruiker probeert het systeem te overbelasten door herhaaldelijk te joinen met scripts | SM7 |
| UES4 | Een aanvaller probeert gamecodes te onderscheppen of manipuleren via onbeveiligde verbindingen. | SM2 |
| UES5 | Een aanvaller probeert SignalR-verbindingen te kapen om real-time gamegegevens te manipuleren. | SM4 |

Tabel 17 Evil user stories Game joinen

### Activity diagram

Hieronder een activity diagram om de functionele flow duidelijk te maken.

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Figure 11 Activity diagram Game joinen

## Use Case: Game spelen

|  |  |
| --- | --- |
| Use case naam | Game spelen |
| Actoren | Developer,Scrum master |
| Doel | Als developers moet er een game gespeeld kunnen worden en backlog item/s storypoints gegeven kunnen worden. |
| Precondities | Er bestaat een game met spelers er in. |
| Hoofdscenario | 1. De scrum master begint een game en voert of pakt uit een bestaande lijst een backlog item.  2. Het geselecteerde backlog item word weergegeven.  3. spelers en de scrum master kunnen storypoints opgeven via de vooraf ingestelde hoeveelheden.  4. De applicatie OF de scrummaster beëindigt de ronde.  5. gaat terug naar stap 1 |
| Alternatieve scenario’s | 2.1. Er kunnen syncing errors ontstaan.  Start een nieuwe game |
| Postcondities | De spelers kunnen input geven.  De scrum master kan input voor een taak/en opgeven. |

Tabel Use case Game spelen

### Betrokken requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beschrijving | Asset/Type | MoSCoW | Testen |
| FR1 | De Game leider kan een game starten en een backlog item selecteren. | Functionaliteit | Must |  |
| FR2 | Het geselecteerde backlog item wordt correct weergegeven met realtime communicatie. | Functionaliteit | Must |  |
| FR3 | Spelers en de leider kunnen storypoints toekennen. | Functionaliteit | Must |  |
| FR4 | De applicatie of de leider kan een ronde beëindigen. | Functionaliteit | Must |  |
| FR5 | Er moet foutafhandeling zijn bij synchronisatiefouten. | Security | Should have |  |
| NFR1 | De game moet realtime communicatie ondersteunen voor storypoint-input. | Functionaliteit | Must |  |
| NFR2 | De game moet betrouwbaar zijn bij het toekennen van storypoints zonder vertraging. | Functionaliteit | Must |  |
| NFR3 | De UI moet foutmeldingen tonen wanneer een synchronisatiefout optreedt. | UX | Should have |  |
| NFR4 | De applicatie moet binnen 1 seconde reageren op een actie van de speler. | Kwaliteit | Should have |  |
| NFR5 | Er mogen geen gevoelige gegevens via de applicatie worden opgeslagen zonder encryptie. | Security | Must |  |

Tabel betrokken requirments Game spelen

### Schermontwerp

Afbeelding met tekst, schermopname, ontwerp

Automatisch gegenereerde beschrijving

Figure Game spelen schermontwerp

### Evil user stories

|  |  |  |
| --- | --- | --- |
| # | Evil user story | Mitigatie |
| UES1 | Een aanvaller probeert ongeautoriseerd een game te starten | SM1 |
| UES2 | Een kwaadwillende gebruiker probeert ongeldige storypoints in te voeren | SM2 |
| UES3 | Een speler probeert het stemmen van anderen te manipuleren | SM3 |
| UES4 | Een aanvaller probeert een game vast te laten lopen door overmatige requests te sturen | SM4 |
| UES5 | Een aanvaller probeert de gamecode te raden en toegang te krijgen tot een privé-game. | SM1 |
| UES6 | Een speler probeert de integriteit van het spel te ondermijnen door vals te spelen met scripts of bots om de uitslag van de game te beïnvloeden. | SM4 |
| UES7 | Een aanvaller probeert een DDOS-aanval uit te voeren door een groot aantal ongeldige verbindingen naar de game te sturen. | SM6 |

Tabel Evil user stories Game spelen

### Activity diagram

Hieronder een activity diagram om de functionele flow duidelijk te maken.

Afbeelding met schermopname, tekst, lijn, Lettertype

Automatisch gegenereerde beschrijving

Figure Activity diagram Game spelen

## Use Case: Aanmaken van Back log Item’s

|  |  |
| --- | --- |
| Use case naam | Aanmaken en beheren van backlog items |
| Actoren | Scrum Master |
| Doel | De Scrum Master kan één of meerdere backlog items aanmaken, beheren en in de juiste volgorde zetten voordat de game begint. |
| Precondities | De Scrum Master is geauthenticeerd.  De Scrum Master heeft toegang tot de backlogbeheerpagina.  Er is nog geen game gestart. |
| Hoofdscenario | 1. De Scrum Master heeft een game aangemaakt. 2. De Scrum Master maakt een nieuw backlog item aan door een naam en beschrijving in te voeren. 3. De Scrum Master kan meerdere backlog items aanmaken en deze lokaal opslaan in cookies of de sessie. 4. De Scrum Master kan de volgorde van de backlog items wijzigen via een drag-and-drop systeem. 5. Wanneer de Scrum Master tevreden is met de backlog, selecteert hij/zij welke item/s er word laten zien in de game. 6. De Scrum Master start de game en de geselecteerde backlog items worden naar de spelers gestuurd. |
| Alternatieve scenario’s | 2.1 De Scrum Master annuleert het aanmaken van een backlog item.  **Het systeem slaat de wijzigingen niet op.**  4.1 De Scrum Master verliest de sessie of verlaat de pagina.  **De backlog blijft behouden in cookies of de sessie.**  5.1 De Scrum Master selecteert geen backlog items om te delen.  **Het systeem geeft een melding dat er één backlog item nodig is om de game te starten.** |
| Postcondities | De backlog items zijn correct opgeslagen en zichtbaar voor de Scrum Master.  De geselecteerde backlog items zijn gedeeld met andere spelers en beschikbaar voor de game.  De Scrum Master kan na afloop van een game de backlog opnieuw beheren en aanpassen. |

Tabel Use case Aanmaken van Back log Item’s

### Betrokken requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beschrijving | Asset/Type | MoSCoW | Testen |
| FR1 | De groepsleider kan één of meerdere backlog items aanmaken voordat de game start. | Functionaliteit | Must |  |
| FR2 | De groepsleider kan een lijst van backlog items beheren en aanpassen. | Functionaliteit | Should have |  |
| FR3 | De groepsleider kan de volgorde van backlog items wijzigen via een drag-and-drop systeem. | Functionaliteit | Could have |  |
| FR4 | De groepsleider kan de backlog items lokaal opslaan in de sessie of cookies. | Functionaliteit | Must |  |
| FR5 | Na het afronden van een game krijgt de groepsleider opnieuw toegang tot de backlog en kan de volgorde aanpassen. | Functionaliteit | Should have |  |
| FR6 | De groepsleider kan bevestigen welke backlog items worden gedeeld met SignalR voor gebruik in de game. | Functionaliteit | Must |  |
| NFR1 | De backlog moet opgeslagen worden in cookies of de sessie van de groepsleider voordat deze gedeeld wordt via SignalR. | Functionaliteit | Must |  |
| NFR2 | Het slepen en wijzigen van de volgorde moet intuïtief en responsief werken. | UX | Must |  |
| NFR3 | De UI moet een duidelijke indicatie geven van de nog niet gespeelde en gespeelde items. | UX | Must |  |
| NFR4 | Backlog items mogen niet verloren gaan bij een herstart van de applicatie zolang deze niet gedeeld zijn. | Functionaliteit | Must |  |
| NFR5 | De backlog moet binnen 1 seconde geladen worden bij het openen van het scherm. | Kwaliteit | Must |  |
| NFR6 | De gegevensoverdracht via SignalR moet veilig en betrouwbaar zijn. | Security | Must |  |
| NFR7 | Er mogen geen gevoelige gegevens in de URL of onversleuteld via SignalR verstuurd worden. | Security | Must |  |

Tabel Betrokken requirments Aanmaken van Back log Item’s

### Schermontwerp

Afbeelding met tekst, schermopname, software, Multimediasoftware

Automatisch gegenereerde beschrijving

Figure Schermontwerp backlog items aanmaken

### Evil user stories

|  |  |  |
| --- | --- | --- |
| # | Evil user story | Mitigatie |
| EUS1 | Een gebruiker manipuleert lokaal opgeslagen backlog items in de browserconsole. | SM2 |
| EUS2 | Een ongeautoriseerde gebruiker probeert backlog items via SignalR te onderscheppen of wijzigen. | SM3 |
| EUS3 | Een kwaadwillende gebruiker verstuurt een extreem lange backlog itemnaam om het systeem te crashen. | SM4 |
| EUS4 | Een gebruiker wijzigt de backlog items van een andere sessie via een manipulatie van de opgeslagen cookies. | SM2 |
| EUS5 | Een gebruiker probeert toegang te krijgen tot privé-backlog items door de opslag te manipuleren in de browser. | SM1 |
| EUS6 | Een niet-geautoriseerde gebruiker probeert via een onjuist roltoewijzing toegang te krijgen tot de backlog items van andere gebruikers. | SM6 |

Tabel Evil user stories Aanmaken van Back log Item’s

### Activity diagram

Hieronder een activity diagram om de functionele flow duidelijk te maken.

Afbeelding met tekst, diagram, schermopname, lijn

Automatisch gegenereerde beschrijving

Figure Activity diagram Aanmaken van Back log Item’s

## Use Case: naam

|  |  |
| --- | --- |
| Use case naam |  |
| Actoren |  |
| Doel |  |
| Precondities |  |
| Hoofdscenario |  |
| Alternatieve scenario’s |  |
| Postcondities |  |

### Betrokken requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Beschrijving | Asset/Type | MoSCoW | Testen |

### Schermontwerp

### Evil user stories

|  |  |  |
| --- | --- | --- |
| # | Evil user story | Mitigatie |

### Activity diagram

Hieronder een activity diagram om de functionele flow duidelijk te maken.

# Bijlage 1 Aanpak Functioneel Ontwerp

In deze bijlage een overzicht van de stappen die genomen zijn om te komen tot een functioneel ontwerp.

Eerst een schematische weergave van de stappen:

Afbeelding met tekst, schermopname, lijn, Lettertype

Automatisch gegenereerde beschrijving

Figuur Ontwikkelstappen Functioneel Ontwerp

In het SSDLC ziet dit er als volgt uit:

Afbeelding met tekst, schermopname, cirkel, diagram

Automatisch gegenereerde beschrijving

Figuur Threat Modeling in het Functioneel Ontwerp

## Domein

Het domein bepalen zorgt voor een gemeenschappelijk referentiekader tussen de stakeholders en het development team.  
Dit referentiekader is een vertrekpunt voor de vervolgstappen. Tijdens het ontwerpen kan het noodzakelijk zijn om het domein aan te passen of uit te breiden. Overleg met stakeholders moet daarvoor plaatsvinden.  
In use cases mogen geen nieuwe domein entiteiten of eigenschappen worden gebruikt, alleen wat in het domein model is opgenomen mag gebruikt worden.

## Use case diagram

Om een functioneel overzicht te bieden van de applicatie is een use case diagram opgenomen. Het zoeken naar de relatie tussen Domein en Use case diagram is een belangrijke stap voor stakeholders en developers.

## Use cases

De use cases moeten in lijn zijn met:

* Requirements analyse
* Domein
* Use case diagram
* Use Case beschrijving (optioneel te gebruiken, wel erg bruikbaar!)

### Principes

Een requirement kan gelden voor meer dan één use case. Bijvoorbeeld: wanneer een request wordt verstuurd, dan toont de pagina een spinner. Dit is een algemeen acceptatiecriterium dat als een algemeen principe kan worden opgenomen in de requirementsanalyse. Hieronder een voorbeeld:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Bron | Beschrijving | Asset/Type | MoSCoW | Testen |
| P1 | IV# | Wanneer een request wordt verstuurd, dan toont de pagina een spinner. | Principe |  |  |
| UC1 | IV1 | Als gebruiker wil ik de CV van een developer kunnen zien om te oriënteren voordat ik contact leg |  | Must | Functioneel |
| … |  | … | .. | .. |  |

Als identificatie is P1 gebruikt. Maar een onderverdeling had ook gekund: PF1 als identificatie voor **p**rincipes voor **f**ormulieren en gegevens versturen.

Een alternatief is om principes op te nemen in een hoofdstuk/bijlage van het Functioneel Ontwerp.

### Acceptatiecriteria

Bij de use case zijn acceptatiecriteria opgenomen, deze zijn overgenomen van de requirementanalyse. Zoals te zien is aan de markeringen zijn nieuwe requirements nodig en aanscherping van bestaande requirements.  
Vervolgens zijn Evil user stories opgesteld en gekeken of de Security Measurements (SM#) vastgelegd in de requirementsanalyse afdoende waren. De conclusie was: SM1 moet worden aangescherpt!

### Schermontwerpen

Om een goed beeld te krijgen van de functionele werking van een webpagina kun je niet zonder een schermontwerp. Aangeraden wordt om Figma te gebruiken. Met deze tool kun je wireframes en componenten maken (bijvoorbeeld een status aan een knop meegeven: hover, disabled).

### Controle requirements

Na het uitwerken van het Functioneel Ontwerp is een controle uitgevoerd of het ontwerp overeenkomstig de eisen de in de requirementsanalyse is opgesteld.