|  |  |
| --- | --- |
| **Software Requirements Specification: Quebble** | **Academie IT en Mediadesign**    Course: OOAD  **Jaap Maaskant, 638742**  **Boris Otte, 640749**  Versie 1.0  Docent:  Herman Telman  AIM 2e leerjaar  1 april 2021  Nijmegen |

Inhoud

[1. Inleiding 3](#_Toc68179516)

[2. Use case model 4](#_Toc68179517)

[2.1 Brief format use cases 5](#_Toc68179518)

[2.2 Requirements 5](#_Toc68179523)

[2.3 Use case prioritering 6](#_Toc68179524)

[2.4 Fully dressed use cases 6](#_Toc68179525)

[2.4.1 Quiz spelen 6](#_Toc68179526)

[2.4.2 Quiz beheren 7](#_Toc68179527)

[2.4.3 Credits bijkopen 9](#_Toc68179528)

[3. Domeinmodel 10](#_Toc68179529)

[3.1 Domeinbeschrijving 11](#_Toc68179530)

[3.2 Ontwerpkeuzes 12](#_Toc68179531)

[4. Bronnenlijst 13](#_Toc68179535)

# Inleiding

Het bedrijf Solid Games wil een quiz-applicatie Quebble ontwikkelen. In dit document wordt de analyse van het domein beschreven aan de hand van een use case model en een domeinmodel. Het ontwerp van de applicatie is te vinden in de software design description.

# Use case model

Op basis van de beschrijving van het spel in de casus is het use case model opgesteld. Dit bestaat uit een use case diagram, usecases in brief format, requirements en fully dressed use cases.

Bij het gebruik van Quebble hebben de volgende actoren een rol:

* Medewerker

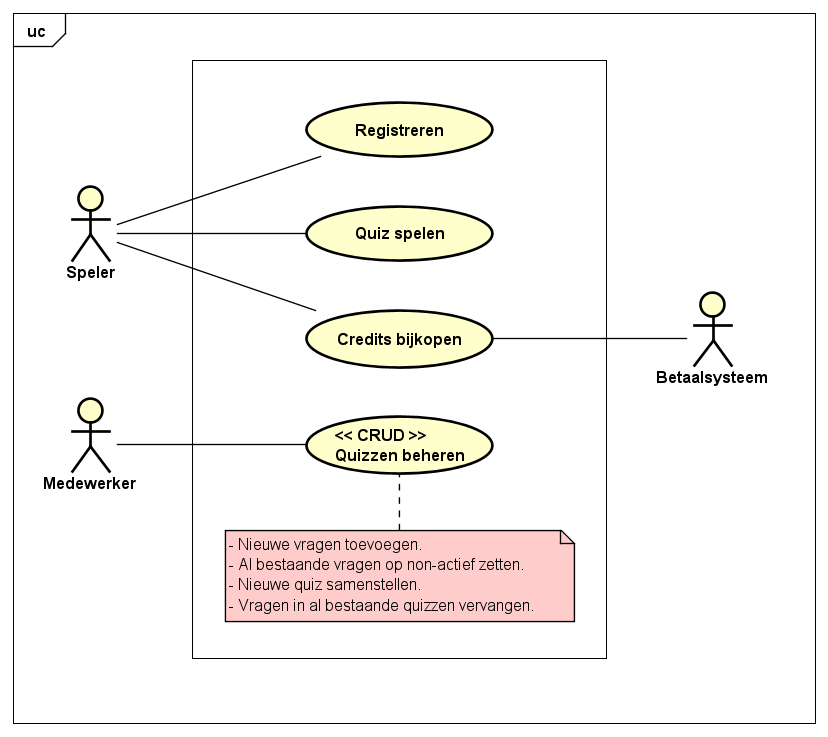
De medewerker beheert vragen en quizzen.

* Speler

De speler registreert zich, kan credits bijkopen, en speelt quizzen.

* Betaalsysteem

Om het kopen van credits af te handelen is een extern betaalsysteem nodig.



Figuur 1- Use case diagram

## Brief format use cases

### Quiz spelen

Een speler geeft aan een quiz te willen spelen. Het systeem controleert of de speler genoeg credits heeft en start de quiz wanneer dit het geval is. Na het invullen van de vragen krijgt de speler een letter voor elk juist beantwoorde vraag. Met deze letters probeert de speler een zo lang mogelijk woord te vormen. Het systeem controleert het woord en bepaald een score op basis van alle uitgevoerde acties.

### Quiz beheren

Een medewerker geeft aan de quizzen te willen beheren. Het systeem geeft een overzicht van de reeds bestaande quizzen en vragen alsmede de mogelijkheid een nieuwe vraag of quiz toe te voegen. De medewerker geeft aan welke gegevens hij wilt wijzigen of toevoegen en voert deze door. Deze stappen worden doorlopen tot de medewerker tevreden is met de doorgevoerde wijzigingen en geeft aan dat deze opgeslagen moeten worden.

### Registreren

Een klant geeft aan zich te willen registreren. Het systeem vraagt om een gebruikersnaam en wachtwoord. Vervolgens maakt het systeem een account aan op basis van deze gegevens. De speler ontvangt een startsaldo van 1000 credits en kan voortaan inloggen met de opgegeven gegevens.

### Credits bijkopen

Een speler geeft aan credits te willen bijkopen. Het systeem toont een keuze menu voor het aantal credits en de daar bijhorende kosten. De speler maakt een keuze en wordt doorgestuurd naar de afhandeling van de betaling. Na het succesvol afhandelen van de betaling krijgt de klant een succesmelding en zijn de credits aan zijn account toegevoegd.

## Requirements

Aan de hand van de casus en de FURPS+ methode zijn de volgende requirements opgesteld:

**Functionality**

* Het systeem moet geregistreerde spelers de mogelijkheid bieden in te loggen op basis van hun emailadres en wachtwoord.
* Het systeem moet in staat zijn het aantal credits van een geregistreerde speler bij te houden.
* Het systeem moet bij kunnen houden welke quizzen een speler heeft gespeeld.

**Usability**

* De tekst binnen de applicatie moet in het Nederlands zijn.

**Performance**

* Het nakijken van de vragen mag niet langer dan 1 seconde duren.
* Het controleren van het gemaakte woorden mag niet langer dan 1 seconde duren.
* Het uitrekenen van de eindscore mag niet langer dan 1 seconde duren.

**Supportability**

* De functionaliteit voor het uitreken van de eindscore moet zo geïmplementeerd worden dat deze later door één persoon binnen één dag te gewijzigd kan worden zonder dat hier andere delen van de applicatie voor te moeten wijzigen.
* Het toevoegen van ondersteuningen voor meerdere talen moet door één persoon binnen één dag doorgevoerd kunnen worden mitst de benodigde tekst in de desbetreffende taal beschikbaar is.

**Plus**

* Het controleren van het woord moet m.b.v. een externe library gebeuren.

## Use case prioritering

De belangrijkste use case is Quiz spelen, hiervoor wordt de applicatie ontwikkeld. De andere drie use cases ondersteunen deze primaire use case. In de prioritering is er voor gekozen om registreren als laagste prioriteit te stellen omdat dit een erg algemene actie is die in veel applicaties voorkomt, en daarom waarschijnlijk geen unieke implementatie nodig zal hebben.

1. Quiz spelen.
2. Quiz beheren.
3. Credits bijkopen.
4. Registreren.

## Fully dressed use cases

### 2.4.1 Quiz spelen

|  |  |
| --- | --- |
| **Primary actor:** Speler | |
| **Stakeholders and Interests:** n.v.t. | |
| **Brief description:** Een speler geeft aan een quiz te willen spelen. Het systeem controleert of de speler genoeg credits heeft en start de quiz wanneer dit het geval is. Na het invullen van de vragen krijgt de speler een letter voor elk juist beantwoorde vraag. Met deze letters probeert de speler een zo lang mogelijk woord te vormen. Het systeem controleert het woord en bepaald een score op basis van alle uitgevoerde acties. | |
| **Preconditions:** Speler is ingelogd | |
| **Postconditions (Success Guarantee):** speler heeftquiz gespeeld, credits zijn van account gehaald. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. Speler geeft aan quiz te willen spelen.  7. Speler beantwoord vraag.  *Stap 6 en 7 worden herhaald tot alle vragen beantwoord zijn.*  9. Speler maakt woord met getoonde letters. | 2. Systeem controleert of speler voldoende credits heeft.  3. [speler heeft voldoende credits] Systeem schrijft kosten spelen quiz af van credits.  4. Systeem zoekt een quiz die de speler nog niet gespeeld heeft.  5. [nog niet gespeelde quiz bestaat] Systeem start quiz.  6. Systeem toont eerstvolgende vraag.  8. Systeem toont letters die bij juist beantwoorde vragen horen.  10. Systeem controleert woord.  11. Systeem berekent en toont eindscore. |
| **Extensions (Alternative Flow):** | |
|  | 3 A. [speler heeft onvoldoende credits] Systeem toont melding dat speler onvoldoende credits heeft om een quiz te spelen.  5 A. [speler heeft alle quizzen al eens gespeeld] Systeem start quiz die het langst geleden gespeeld is. *Gaat verder bij stap 6.* |

Figuur 2 – Fully dressed use case: Quiz spelen

### 2.4.2 Quiz beheren

|  |  |
| --- | --- |
| **Primary actor:** Medewerker | |
| **Stakeholders and Interests:** n.v.t. | |
| **Brief description:** Een medewerker geeft aan de quizzen te willen beheren. Het systeem geeft een overzicht van de reeds bestaande quizzen en vragen alsmede de mogelijkheid een nieuwe vraag of quiz toe te voegen. De medewerker geeft aan welke gegevens hij wilt wijzigen of toevoegen en voert deze door. Deze stappen worden doorlopen tot de medewerker tevreden is met de doorgevoerde wijzigingen en geeft aan dat deze opgeslagen moeten worden. | |
| **Preconditions:** Medewerker is ingelogd en heeft toegangsrechten tot admin-domein. | |
| **Postconditions (Success Guarantee):** Aangepaste gegevens zijn doorgevoerd en opgeslagen. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. Medewerker geeft aan quizzen te willen beheren.  3. Medewerker geeft aan een nieuwe vraag toe te willen voegen.  5. Medewerker vult gegevens in.  6. Medewerker geeft aan de ingevulde gegevens op te willen slaan.  *Stap 2 t/m 7 worden herhaald tot medewerker alle gewenste veranderingen heeft doorgevoerd.* | 2. Systeem toont keuzemenu met beheeropties.  4. Systeem geeft mogelijkheid gegevens op te geven.  7. Systeem slaat gegevens op. *Gaat terug naar stap 2.* |
| **Extensions (Alternative Flow):** | |
| 3 A. Medewerker geeft aan een al bestaande vraag op non-actief te willen zetten.  5 A. Medewerker selecteert vraag om op non-actief te zetten.  *Gaat verder bij stap 6.* | 4 A. Systeem geeft overzicht vragen. |
| 3 B. Medewerker geeft aan een nieuwe quiz te willen samenstellen.  5 B. Medewerker geeft basisgegevens quiz op.  7 B. Medewerker selecteert vragen die de nieuwe quiz gaan vormen.  *Gaat verder bij stap 6.* | 4 B. Systeem geeft mogelijkheid basisgegevens in te vullen.  6 B. Systeem geeft overzicht vragen. |
| 3 C. Medewerker geeft aan een vraag in een al bestaande quiz te willen vervangen.  5 C. Medewerker selecteert quiz om vraag van te vervangen.  7 C. Medewerker selecteert vraag om te vervangen.  *Gaat verder bij stap 6.*  9 C. Medewerker selecteert vraag.  *Gaat verder bij stap 6.* | 4 C. Systeem geeft overzicht quizzen.  6 C. Systeem geeft overzicht van vragen in quiz.  8 C. Systeem toont mogelijke vragen om de geselecteerde vraag mee te vervangen. |
| 3 D. Medewerker geeft aan alle gewenste veranderingen te hebben doorgevoerd. | 4 D. Systeem sluit beheermenu. |

Figuur 3 – Fully dressed use case: Quiz beheren

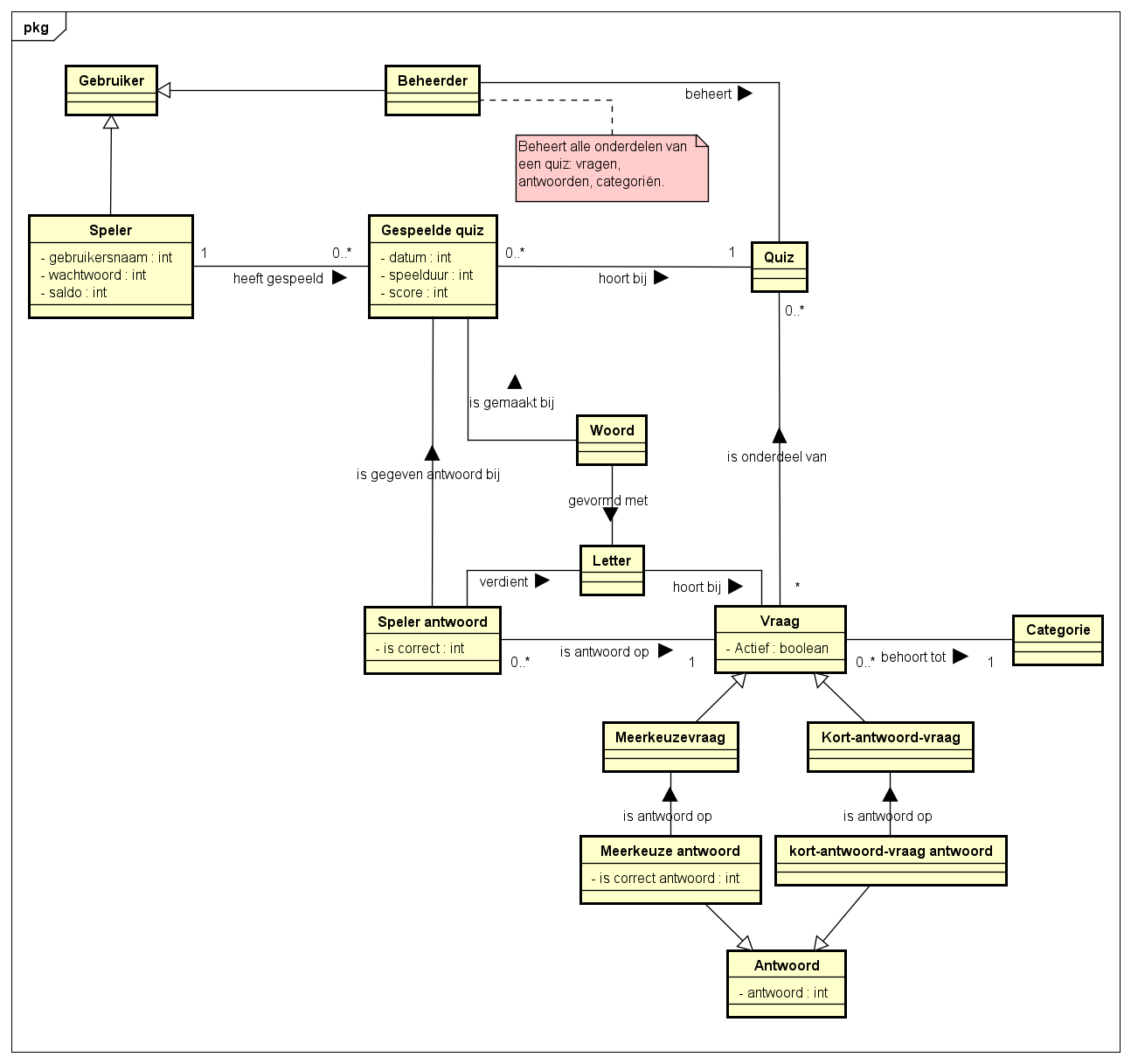
### 2.4.3 Credits bijkopen

|  |  |
| --- | --- |
| **Primary actor:** Speler | |
| **Stakeholders and Interests:** n.v.t. | |
| **Brief description:** Een speler geeft aan credits te willen bijkopen. Het systeem toont een keuzemenu voor het aantal credits en de daar bijhorende kosten. De speler maakt een keuze en wordt doorgestuurd naar de afhandeling van de betaling. Na het succesvol afhandelen van de betaling krijgt de klant een succesmelding en zijn de credits aan zijn account toegevoegd. | |
| **Preconditions:** Speler is ingelogd | |
| **Postconditions (Success Guarantee):** Betaling is voltooid, en credits zijn toegevoegd aan account. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. Speler geeft aan credits te willen bijkopen.  3. De speler selecteert het gewenste aantal credits.  5. De speler betaalt d.m.v. de betaalservice. | 2. Het systeem toont een keuzemenu.  4. Het systeem verwijst de speler door naar een betaalservice.  6. [Betaling voltooid] Het systeem voegt credits toe aan het account.  7. Het systeem toont een succesmelding. |
| **Extensions (Alternative Flow):** | |
|  | 6 A. [Betaling niet voltooid] Het systeem voegt geen credits toe aan het account.  7 A. Het systeem toont een foutmelding. |

Figuur 4 – Fully dressed use case: Credits bijkopen

# Domeinmodel

Om een beter overzicht te krijgen van het domein Quebble, is er een domeinmodel opgesteld. Dit model is opgesteld op basis van de use cases en de beschrijving van het spel in de casus. Hierbij is vooral aandacht besteed aan het in kaart brengen van concepten voor de use case Quiz spelen.



Figuur 5 - Domeinmodel

## Domeinbeschrijving

|  |  |
| --- | --- |
| Concept | Beschrijving |
| Antwoord | Elke quiz is een mix van vragen. Elke vraag behoort tot een categorie. Vragen kunnen toegevoegd worden of op niet-actief gezet worden door een beheerder. |
| Beheerder | Een beheerder is een medewerker van Solid Games. Deze medewerker kan vragen en quizzen beheren. |
| Categorie | Elke vraag behoort tot een categorie, bijv. Geografie, Muziek, Sport, Taal, et cetera. |
| Gebruiker | Een gebruiker is iemand die het systeem gebruikt. Medewerkers van Solid Games zijn beheerders, aangemelde klanten zijn spelers. |
| Gespeelde quiz | Een gespeelde quiz is een quiz die een speler op het moment speelt, of eerder heeft gespeeld. Bij het spelen van een quiz geeft de speler antwoorden op vragen. Als alle vragen beantwoord zijn ontvangt de speler voor elke goed beantwoorde vraag een letter. Met de letters dient de speler een zolang mogelijk woord te vormen. Aan het eind van het spel krijgt de speler zijn score te zien. |
| Kort-antwoord-vraag | Bij een kort-antwoord-vraag geeft de speler antwoord op een open vraag. |
| Kort-antwoord-vraag-antwoord | Bij een kort-antwoord-vraag worden één of meer antwoorden goed gerekend. |
| Letter | Bij het goed beantwoorden van vragen verdient de speler letters. Met deze letters dient de speler een zolang mogelijk woord te vormen. |
| Meerkeuze antwoord | Een meerkeuze antwoord is een antwoord dat aan de speler getoond wordt bij een meerkeuzevraag. Deze antwoorden kunnen goed of fout zijn. |
| Meerkeuzevraag | Bij een meerkeuzevraag worden meerder antwoorden getoond aan de speler, daarvan is er één het correcte antwoord. |
| Quiz | Een quiz is een mix van meerkeuzevragen en kort-antwoordvragen. |
| Speler | Een speler kan quizzen spelen. Een klant wordt een speler als hij zich heeft geregistreerd met een gebruikersnaam en wachtwoord. Bij registreren ontvangt de speler een startsaldo. Dit saldo kan gebruikt worden om quizzen mee te spelen. |
| Speler antwoord | Een door de speler gegeven antwoord op een vraag. |
| Vraag | Een vraag is deel van een quiz en komt in de vormen meerkeuzevraag en kort-antwoord-vraag voor. |
| Woord | Met verdiende letters dient de speler een zolang mogelijk woord te vormen. Dit woord wordt op geldigheid gecontroleerd. |

Figuur 6 – Domeinbeschrijving

## Ontwerpkeuzes

### Gespeelde Quiz

Het concept Gespeelde Quiz in ons ontwerp is bedoeld om concepten, met informatie die nodig is om een quiz te spelen, te verbinden. Dit concept stelt zowel een quiz voor die eerder gespeeld is, als een quiz die op het moment gespeeld wordt door de speler.

### Vraag en Antwoord

Zowel Vraag als Antwoord hebben wij verdeeld in een meerkeuze en een kort-antwoord variant. Meerkeuzevragen worden anders getoond dan kort-antwoord-vragen, namelijk met de mogelijke antwoorden er bij gegeven. Deze mogelijke antwoorden op de meerkeuze vraag dienen daarom ook goed OF fout te zijn, terwijl antwoorden op kort-antwoord-vragen altijd goed zijn.

### Antwoord en Speler antwoord

Wij hebben er voor gekozen om Speler antwoord niet over te laten erven van Antwoord. Wij vonden het Speler antwoord conceptueel te verschillend van de verschillende soorten antwoorden op verschillende soorten vragen. Het Speler antwoord is complexer dan andere antwoorden in de zin dat het niet alleen een String is, maar ook zelf een letter heeft (of ophaalt) en dat het Speler antwoord nagekeken moet worden waar de antwoorden op vragen al van tevoren als goed of fout bestempeld worden.

# Bronnenlijst

HAN AIM. Case Study Quebble OOSE-OOAD 2020-2021.

Gedownload via <https://onderwijsonline.han.nl/>

Ottinger, T., Langr, J. (2009). FURPS+.

Geraadpleegd via <http://agileinaflash.blogspot.com/2009/04/furps.html>