

SOFTWARE REQUIREMENTS SPECIFICATION (SRS) - CASE STUDY: FINCH - OOAD 2022/2023

Martijn Gerritsen (1644170) en Dani Botland (617442)
DOCENT: MARCO ENGELBART Datum: 30-03-2023

Inhoud

1. Introduction	2
1.1 Overall description	2
1.2 User Classes and Characteristics	2
1.3 Operation Environment	2
1.4 Design and Implementation Constraints	2
1.5 Product Functions	2
2. Domain Model	4
2.1 Toelichting bij concepten	4
3. Use-case Descriptions	5
3.1 Registreren	5
3.2 Spelverloop quiz	6
3.3 Kopen-game-assets	8
4. Other functional requirements	9
5. Non-functional requirements	10
6. User interface sketches	11
7. Bronvermelding	12

1. Introduction

1.1 Overall description

In dit document wordt het SRS beschreven dat bij de case study 'Finch' hoort. Dit is een beroepsproduct van OOSE-OOAD. Het bedrijf PeeWee Games wil een kennisquiz-applicatie Finch ontwikkelen die beschikbaar komt op de meest gangbare devices. Dit is een spel waarbij vragenlijsten beantwoord worden door gebruikers.

1.2 User Classes and Characteristics

De aanwezige actoren en hun relatie tot het eindproduct zijn de volgende:

- Speler: De gebruiker van Finch. De speler kan Finch spelen door quizen te beantwoorden en munten te gebruiken. Via een winkel kan de speler munten bijkopen tegen tarieven. De speler heeft een gebruikersnaam, wachtwoord, saldo en munten tot zijn bezit.
- Medewerker: De gebruiker die de quizen en vragen beheerd/onderhoud. De medewerker moet quizen en vragen kunnen toevoegen, verwijderen en aanpassen.

1.3 Operation Environment

De applicatie is een Java-applicatie die in de console wordt gespeeld op een PC. Uiteindelijk zal de applicatie op de meest gangbare devices beschikbaar worden. Dit zijn de vier meest gebruikte besturingssystemen: Windows, MacOS, Linux en BSD.

1.4 Design and Implementation Constraints

Er zijn een aantal ontwerp- en implementatie beperkingen waaraan gehouden wordt:

- De applicatie moet geschreven worden in Java.
- De applicatie moet gebruikt kunnen worden door meerdere puntentellingsystemen.

1.5 Product Functions

Hieronder is het use case diagram te vinden. Verderop zijn de use case beschrijvingen te vinden.

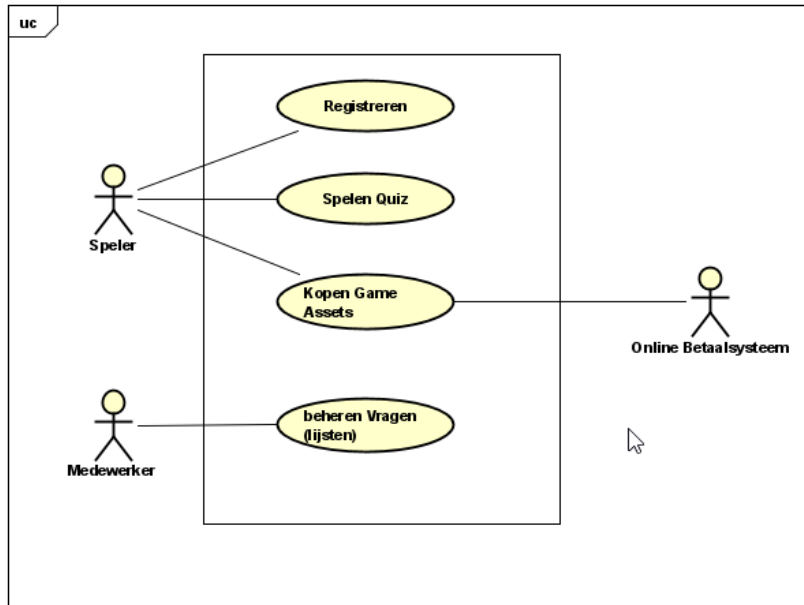


Figure 1: use case diagram

1. Registreren

Om Finch te kunnen spelen, moet de speler eerst registreren. De speler voert een gebruikersnaam en wachtwoord in. Finch slaat deze gegevens op, hierna wordt de speler 100 munten toegewezen en krijgt de speler twee vragenlijsten cadeau.

2. Spelen quiz

De speler kan een vragenlijst selecteren en een quiz spelen. De speler zal de vragen één voor één beantwoorden (waarbij de speler niet terug kan gaan). Per goed beantwoorde vraag verdient de speler een punt. Er zal een bonus worden gegeven, waarbij gekeken wordt naar de totale tijd die nodig was om de quiz te doorlopen. Indien alle vragen goed beantwoord zijn, wordt de speler 2 extra punten toegekend. Er zal voor elke speler per vragenlijst een 'lifetime best' bijgehouden worden.

3. Kopen game assets

De speler kan vragenlijsten en munten kopen in de Finch-shop. Dit kan hij doen met gekregen munten. Het aankoopbedrag van de aangeschafte lijst wordt dan in mindering gebracht op het saldo van de speler. De speler kan van een aangeschafte vragenlijst een jaar gebruik maken. Het afrekenen hiervan gebeurt via een online betaalservice.

4. Beheren vragen (lijsten)

Voor medewerkers van PeeWee Games is het mogelijk om vragen te beheren. De medewerker kan nieuwe vragen (meerkeuze of open) toevoegen, aanpassen en verwijderen met bijbehorende antwoorden. Ook thema's en bijbehorende vragenlijsten kunnen toegevoegd en gewijzigd worden.

2. Domain Model

Hieronder is het domeinmodel te zien voor Finch.

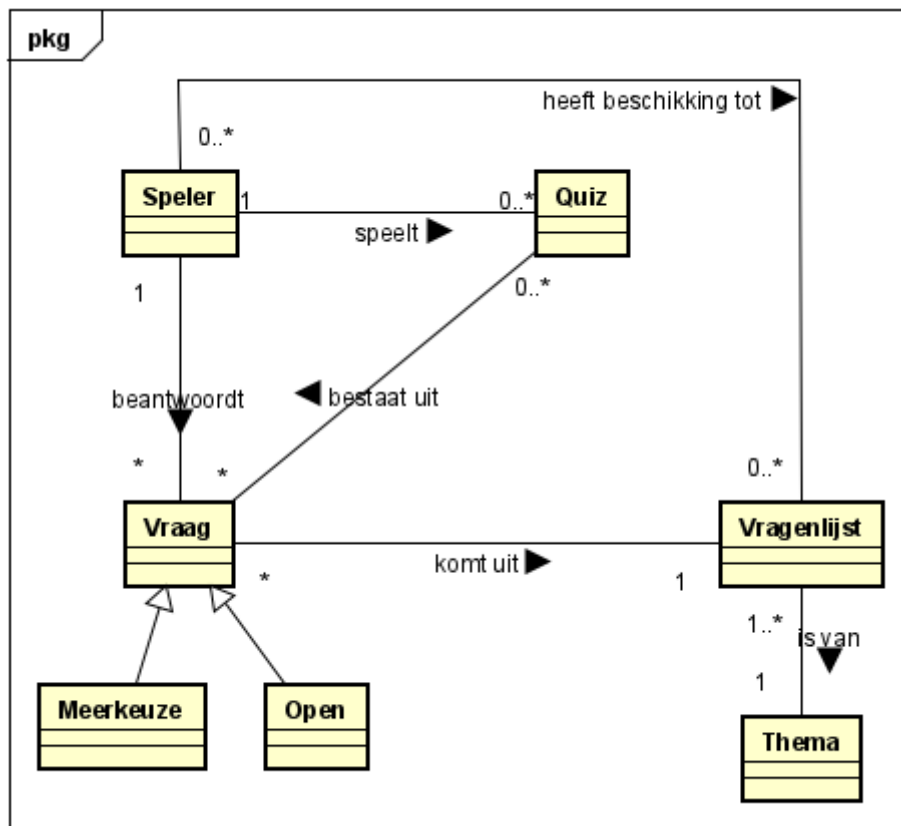


Figure 2: domain model

2.1
2.1

Toelichting bij concepten

Hieronder is een toelichting te zien van de concepten van bovenstaande domeinmodel.

- Speler: Speelt een quiz door vragen te beantwoorden. Heeft een gebruikersnaam, wachtwoord en muntensaldo.
- Vraag: Een vraag dat onderdeel uitmaakt van een vragenlijst. Kan meerkeuze of open zijn.
- Meerkeuzevraag: Een vraag die meerdere antwoorden bevat, hiervan is er maar één juist.
- Open vraag: Een vraag die één of meerdere juiste antwoorden heeft.
- Vragenlijst: Een lijst van vragen.
- Quiz: Een spel dat bestaat uit 10 vragen.
- Thema: Een thema heeft een aantal onderwerpen die verschillende vragenlijsten heeft.

3. Use-case Descriptions

In dit hoofdstuk zullen de uitgebreide uitwerkingen van de use cases plaatsvinden. Dit zijn fully-dressed use cases. Bijbehorende activity diagrams zijn te vinden in het Software Design Description bestand (SDD).

3.1 Registreren

Use case name: Registreren	
Primary actor: Speler	
Stakeholders and Interests: Speler: Wil na het registreren meteen kunnen beginnen met spelen.	
Brief description: Om Finch te kunnen spelen, moet de speler eerst registreren. De speler voert een gebruikersnaam en wachtwoord in. Finch slaat deze gegevens op, hierna wordt de speler 100 munten toegewezen en krijgt de speler twee vragenlijsten cadeau.	
Preconditions: n.v.t.	
Postconditions: De speler is geregistreerd. De speler heeft twee vragenlijsten en 100 munten tot zijn beschikking.	
Main Success Scenario (Basic Flow):	
Actor Action	System Responsibility
1. De speler kiest ervoor om zich te registreren. 2. De speler vult een gebruikersnaam en wachtwoord in.	3. Systeem registreert de speler en voegt 100 munten toe aan het startsaldo van de speler.
Extensions (Alternative Flow):	
.	3a. De gebruikersnaam en/of wachtwoord is leeg en/of foutief ingevuld 3b. Systeem toont foutmelding en keert terug naar stap 2.

3.2 Spelverloop quiz

Use case name: Spelverloop quiz	
Primary actor: Speler	
Stakeholders and Interests: Speler: Speelt de quiz en beantwoord een vragenlijst.	
Brief description: De speler kan een vragenlijst selecteren en een quiz spelen. De speler zal de vragen één voor één beantwoorden (waarbij de speler niet terug kan gaan). Per goed beantwoorde vraag verdient de speler een punt. Er zal een bonus worden gegeven, waarbij gekeken wordt naar de totale tijd die nodig was om de quiz te doorlopen. Indien alle vragen goed beantwoord zijn, wordt de speler 2 extra punten toegekend. Er zal voor elke speler per vragenlijst een 'lifetime best' bijgehouden worden.	
Preconditions: De speler is geregistreerd. De speler heeft genoeg startsaldo om een quiz te kunnen spelen. De speler heeft een vragenlijst ter beschikking.	
Postconditions: De speler heeft alle quizvragen beantwoord. De speler heeft een bonus gekregen aan de hand van totale tijd die nodig was om de quiz te doorlopen. Het systeem heeft (de eventuele) 'lifetime best' bijgehouden.	
Main Success Scenario (Basic Flow):	
Actor Action	System Responsibility
1. De speler start een quiz.	2. Het systeem laat een nieuwe quizvraag zien.
3. De speler beantwoordt de quizvraag. <i>Stap 2 en 3 worden herhaald totdat alle quizvragen zijn beantwoord.</i>	4. Het systeem controleert het antwoord. 5. Het systeem keert een bonus uit aan de speler aan de hand van de totale tijd die nodig was om de quiz te doorlopen. 6. Het systeem houdt (de eventuele) 'lifetime best' bij.
Extensions (Alternative Flow):	
.	4a. systeem constateert een invalide antwoord. 4b. Het systeem toont een foutmelding en herhaalt stap 3.

Hieronder is een activity diagram te zien van de use case: Spelverloop Quiz.

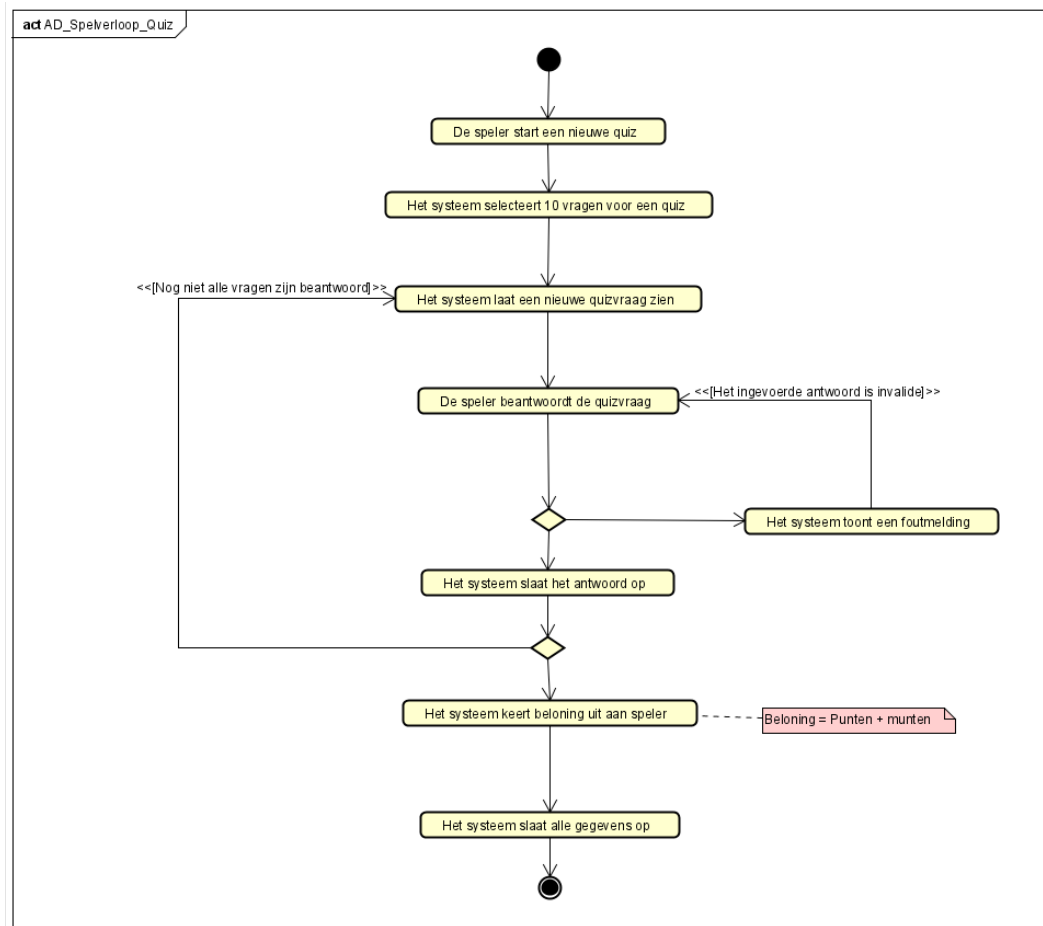


Figure 3: activity diagram spelen quiz

3.3 Kopen-game-assets

Use case name: Kopen-game-assets

Primary actor: Speler

Stakeholders and Interests: Speler: Koopt munten uit de Finch-shop.

Brief description: De speler kan vragenlijsten kopen in de Finch-shop. Dit kan hij doen met gekregen munten. Het aankoopbedrag van de aangeschafte lijst wordt dan in mindering gebracht op het saldo van de speler. De speler kan van een eenmaal aangeschafte vragenlijst een jaar gebruik maken. De speler kan in de Finch-shop ook munten kopen. Het afrekenen hiervan gebeurt via een online betaalservice.

Preconditions: De speler is geregistreerd. De speler heeft beschikbaar geld om vragenlijsten en/of munten te kunnen kopen.

Postconditions: De speler heeft gekochte vragenlijsten en/of munten toegevoegd gekregen aan Finch. De online betaalservices of het muntensaldo hebben het aankoopbedrag van de speler ontvangen.

Main Success Scenario (Basic Flow):

Actor Action	System Responsibility
--------------	-----------------------

<p>1a. De speler koopt een vragenlijst.</p> <p>1b. De speler koopt munten.</p>	<p>2a. Het systeem brengt het aankoopbedrag in mindering van de munten balans en voegt de vragenlijst toe aan de beschikbare vragenlijsten in Finch.</p> <p>2b. Het systeem controleert of de betaling via de online betaalservice is voltooid en keert de juiste hoeveelheid munten uit aan de munten balans van de speler.</p>
Extensions (Alternative Flow):	
	<p>2aa. De speler heeft niet voldoende muntensaldo om de vragenlijst te kunnen kopen en geeft een foutmelding.</p> <p>2bb. De speler zijn betaling is niet geslaagd en het systeem geeft een foutmelding.</p>

4. Other functional requirements

Er zijn geen overige functionele requirement van toepassing.

5. Non-functional requirements

Hieronder zijn de overige niet-functionele requirements te zien. Dit wordt opgesteld via de FURPS+ methodiek (zie bronvermelding).

FURPS+	Niet functionele eis
Functionality	X
Usability	NFE1. Het spel Finch wordt in het Nederlands weergegeven. NFE2. Het spel is beschikbaar op de 4 meest gebruikte besturingssystemen voor computers (Windows, Mac-OS, Linux & BSD).
Reliability	X
Performance	NFE3. Het systeem doet maximaal 1 sec over het laden van een vraag tijdens het spelen van een quiz.
Supportability	NFE4. Een developer moet binnen 1 werkdag een nieuwe taal aan het systeem kunnen toevoegen. NFE5. Een developer moet de puntentelling kunnen veranderen binnen 1 werkdag.
+ Implementation	NFE6. Afrekenen in de finch shop gebeurt via een online betaalservice van een derde partij.

6. User interface sketches

Hier komen de eventuele user interface sketches.

Deze zijn niet van toepassing.

7. Bronvermelding

Identifying Non-Functional Requirements. (z.d.).

<http://www.cs.sjsu.edu/faculty/pearce/modules/lectures/ooa/requirements/>

[IdentifyingURPS.htm](#)