

Software Requirements Specification

J. van Aalst, 639519
F.K.A. Soffers; 567780
OOAD
Finch Casus
20-10-2022

Inhoudsopgave

Introduction	3
Overall Description	3
User Classes and Characteristics	3
Operating Environment	3
Domain Model	3
Use case diagrams	4
Use case beschrijvingen	5
Brief Format	5
Registreren	5
Quiz spelen	5
Munten kopen	5
Vragenlijst kopen	6
Vragenlijst toevoegen	7
Thema toevoegen	7
Thema wijzigen	7
Vragen toevoegen	7
Vraag wijzigen	8
Vraag verwijderen	9
Fully Dressed Format	9
Use Case: Registreren	9
Use Case: Spelverloop quiz	11
Use Case: Finch-shop	13
Functionele eisen	15
Registreren	15
Vragenlijsten	15
Vragen	15
Toegang	15
Spelverloop quiz	15
Finch-shop	15
Non-functional Requirements	15

Introduction

Overall Description

In het SRS behandelen wij de software requirements voor de Finch casus. Voornamelijk gaan wij in op de use cases en basis requirements voor de software.

User Classes and Characteristics

Het spel kent 2 primaire actoren waaronder de speler van het spel en de medewerker. De speler kent de volgende acties:

- Quiz spelen
- Munten kopen
- Vragenlijsten kopen

De medewerker de volgende acties:

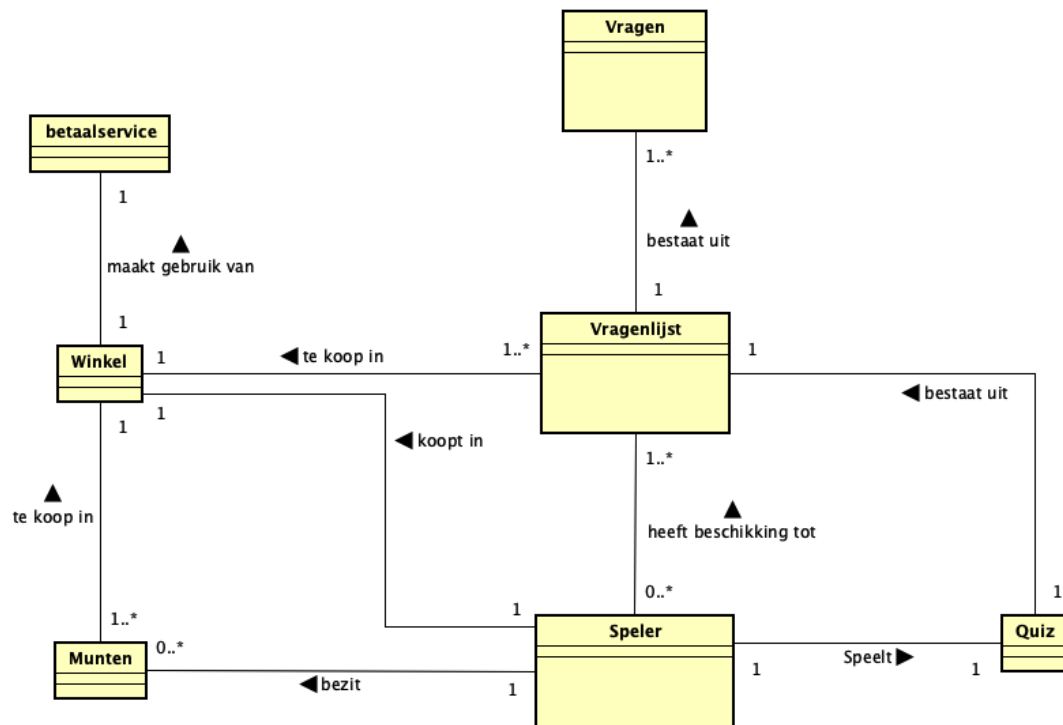
- Vragen toevoegen, verwijderen of bewerken
- Vragenlijsten toevoegen, verwijderen of bewerken
- Thema's toevoegen, verwijderen of bewerken

Operating Environment

In de opdracht staat de applicatie normaal gesproken speelbaar moet zijn op de meest gangbare devices.

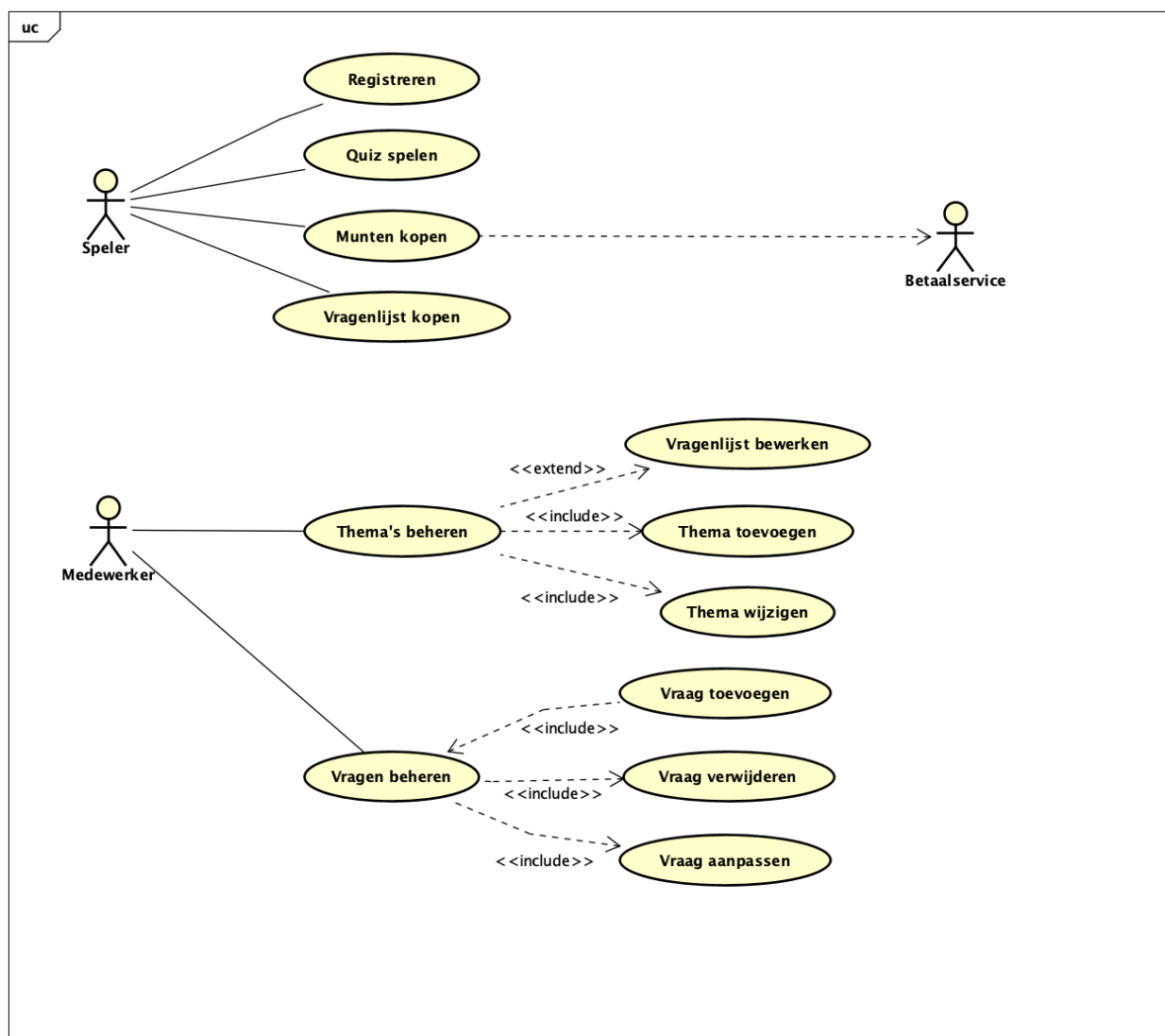
Alleen doordat de basis op het design ligt bij de opdracht hoeft er alleen een Java applicatie opgeleverd te worden.

Domain Model



Use case diagrams

De Finch casus is opgedeeld in verschillende use cases die zijn afgeleid uit de voorbeelden. In totaal zijn er 10 verschillende use cases, 2 verschillende actoren waaronder de speler die het spel speelt en de medewerker die het spel onderhoudt. Daarnaast is er nog een externe actor: de betaalservice die betrokken is bij betalingen binnen de applicatie.



Use case beschrijvingen

Dit hoofdstuk bevat de brief format use case beschrijvingen voor alle opgestelde use cases.

Brief Format

Registreren

Use case	Registreren
Stakeholders & Interests	Speler
Brief description	Om het spel te kunnen spelen moet de speler zich registreren door een gebruikersnaam en wachtwoord op te geven. Na het registreren ontvangt de speler 100 munten en 2 vragenlijsten cadeau.
Preconditions	De speler heeft nog geen account.
Postconditions	Het account wordt geregistreerd in het systeem en de speler ontvangt 100 munten en 2 vragenlijsten cadeau.

Quiz spelen

Use case	Quiz Spelen
Stakeholders & Interests	Speler
Brief description	<p>De speler kiest een vragenlijst die hij wil spelen en krijgt vervolgens 10 vragen uit die lijst. De vragen bestaan uit zowel meerkeuze als open vragen.</p> <p>De vragen worden een voor een beantwoord en de speler mag niet teruggaan naar de vorige vraag.</p> <p>Per goed beantwoorde vraag krijgt de speler 1 punt, daarnaast krijgt hij ook nog bonuspunten op basis van de totale tijd en eventueel 2 extra punten als alle vragen correct zijn beantwoord.</p>
Preconditions	De speler is ingelogd en heeft beschikking tot de geselecteerde vragenlijst.
Postconditions	De score van de speler wordt opgeslagen.

Munten kopen

Use case	Munten kopen
----------	--------------

Stakeholders & Interests	Speler & Betaalservice
Brief description	De speler koopt in de Finch winkel extra munten en wordt voor het afrekenen doorgestuurd naar de betaalservice.
Preconditions	De speler is ingelogd.
Postconditions	De munten zijn toegevoegd aan het account van de speler.

Vragenlijst kopen

Use case	Vragenlijst kopen
Stakeholders & Interests	Speler
Brief description	<p>De speler koopt in de winkel een vragenlijst voor een bepaald aantal munten. De prijzen kunnen verschillen per vragenlijst.</p> <p>Vervolgens heeft de speler 1 jaar toegang tot de vragenlijst.</p>
Preconditions	De speler is ingelogd en heeft voldoende munten.
Postconditions	De speler krijgt toegang tot de vragenlijst en de munten worden in mindering gebracht.

Vragenlijst toevoegen

Use case	Vragenlijst toevoegen
Stakeholders & Interests	Medewerker Finch
Brief description	De medewerker voegt via de beheeromgeving een nieuwe vragenlijst toe aan een thema.
Preconditions	De medewerker heeft toegang tot de beheer omgeving. Het bijbehorende thema moest bestaan.
Postconditions	De vragenlijst is toegevoegd aan het systeem.

Thema toevoegen

Use case	Thema toevoegen
Stakeholders & Interests	Medewerker Finch
Brief description	De medewerker voegt via de beheeromgeving een nieuwe thema toe aan het systeem.
Preconditions	De medewerker heeft toegang tot de beheer omgeving. Het thema bestaat nog niet.
Postconditions	Het thema is toegevoegd aan het systeem.

Thema wijzigen

Use case	Thema wijzigen
Stakeholders & Interests	Medewerker Finch
Brief description	De medewerker wijzigt via de beheeromgeving een bestaand thema in het systeem.
Preconditions	De medewerker heeft toegang tot de beheer omgeving. Het thema bestaat.
Postconditions	Het thema is gewijzigd in het systeem.

Vragen toevoegen

Use case	Vragen toevoegen
Stakeholders & Interests	Medewerker Finch
Brief description	De medewerker voegt via de beheeromgeving één of meerdere vragen toe aan een vragenlijst inclusief het antwoord. De vraag moet een open of meerkeuzevraag zijn.
Preconditions	De medewerker heeft toegang tot de beheer omgeving. De vragenlijst bestaat.
Postconditions	De vragen zijn toegevoegd aan de vragenlijst.

Vraag wijzigen

Use case	Vraag wijzigen
Stakeholders & Interests	Medewerker Finch
Brief description	De medewerker wijzigt via de beheeromgeving een vraag en/of het antwoord. De vraag moet een open of meerkeuzevraag zijn.
Preconditions	De medewerker heeft toegang tot de beheer omgeving. De vragenlijst bestaat.
Postconditions	De vraag is aangepast in de vragenlijst.

Vraag verwijderen

Use case	Vraag verwijderen
Stakeholders & Interests	Medewerker Finch
Brief description	De medewerker verwijdert een vraag uit een vragenlijst.
Preconditions	De medewerker heeft toegang tot de beheer omgeving.
Postconditions	De vraag is verwijderd uit de vragenlijst.

Fully Dressed Format

Use Case: Registreren

Primary Actor	- Speler
Stakeholders and Interests	- ?
Brief description	
<i>Om de kennisquiz Finch te kunnen spelen moet je je registreren door een gebruikersnaam en wachtwoord op te geven. Nadat je je hebt geregistreerd ontvang je een startsaldo van 100 munten en krijg je twee vragenlijsten cadeau.</i>	
Preconditions	- Er is de mogelijkheid voor een gebruiker om zich te registreren
Postconditions (Success Guarantee)	<ul style="list-style-type: none"> - De gebruiker heeft zich geregistreerd met de opgegeven gebruikersnaam en wachtwoord combinatie - De nieuw geregistreerde gebruiker heeft een startsaldo ontvangen van 100 munten - De nieuw geregistreerde gebruiker heeft 2 vragenlijsten* ontvangen
Main Success Scenario (Basic Flow)	
Actor Action	System Responsibility
1. Gebruiker geeft aan zich te willen registreren	2. Het systeem toont de gebruiker een formulier waar deze zijn gewenste gebruikersnaam en wachtwoord in kan vullen
3. De gebruiker vult de gewenste gebruikersnaam en wachtwoord in	4. Het systeem verstuurt deze gegevens naar de database
	5. Het systeem slaat de gebruiker op

	6. Het systeem vult het saldo van de zoiust geregistreeerde gebruiker aan met 100 munten
	7. Het systeem geeft aan de zoiust geregistreeerde gebruiker twee cadeaulijsten cadeau
Extensions (Alternative Flow): “Er bestaat al een gebruiker met deze gebruikersnaam”	
Actor Action	System Responsibility
1. <i>Gebruiker geeft aan zich te willen registreren</i>	2. <i>Het systeem toont de gebruiker een formulier waar deze zijn gewenste gebruikersnaam en wachtwoord in kan vullen</i>
3. <i>De gebruiker vult de gewenste gebruikersnaam en wachtwoord in</i>	4. <i>Het systeem verstuurt deze gegevens naar de database</i>
→ Start alternative flow at 5	5. Het systeem merkt op dat er al een gebruiker met deze gebruikersnaam is
	6. Het systeem toont een melding aan de gebruiker dat de naam al bezet is
	7. Het systeem vraagt aan de gebruiker een nieuwe gebruikersnaam op te geven
8. De gebruiker vult een nieuwe gebruikersnaam in	9. <i>Het systeem verstuurt de gegevens naar de database</i>
—Continue success scenario—	

Use Case: Spelverloop quiz

Primary Actor	<ul style="list-style-type: none"> - Gebruiker(s) - Speler (dit is een gebruiker, zodra het spel begint)
Stakeholders and Interests	
Brief description	
<p><i>Spelers selecteren een vragenlijst en krijgen uit deze vragenlijst 10 vragen voorgelegd. Deze vragen zijn een mix van meerkeuzevragen en open vragen. De speler beantwoordt deze vragen een voor een. Spelers kunnen de antwoorden van vorige vragen niet meer aanpassen. Per goed beantwoorde vraag verdienen de spelers punten. Er kunnen bonuspunten verdiend worden en voor spelers wordt per vragenlijst een 'lifetime best' score bijgehouden.</i></p>	
Preconditions	<ul style="list-style-type: none"> - Een speler heeft zich als gebruiker geregistreerd - Er is een vragenlijst beschikbaar
Postconditions (Success Guarantee)	<ul style="list-style-type: none"> - De speler heeft antwoord gegeven op 10 vragen en hiermee de quiz afgerond - De speler heeft een score gekregen - De 'lifetime best' score van de speler is opgeslagen / bijgewerkt
Main Success Scenario (Basic Flow)	
Actor Action	System Responsibility
1. Een gebruiker selecteert een vragenlijst	2. Het systeem selecteert 10 vragen uit de geselecteerde vragenlijst (bestaande uit open en meerkeuze vragen)
3. De speler geeft aan te willen starten met het spel	4. Het systeem start de timer
	5. Het systeem toont de eerste vraag aan de speler
	6. Het systeem biedt de speler de mogelijkheid een antwoord in te vullen
– Start loop (10x, het max. Aantal vragen) –	
7. De speler geeft antwoord op de getoonde vraag	8. Het systeem slaat dit antwoord op
9. De speler geeft aan naar de volgende vraag te willen gaan	10. Het systeem toont de volgende vraag
— Einde loop —	
11. De speler geeft aan de quiz te willen	12. Het systeem stopt de timer

afroonden	
	13. Het systeem controleert de vragen
	14. Het systeem berekent de bonuspunten
	15. Het systeem werkt bij / registreert de 'lifetime best' van de speler
	16. Het systeem toont de speler zijn score
	17. Het systeem toont de speler de optie om terug te gaan naar de hoofdpagina
Extensions (Alternative Flow): "vroegtijdig beëindigen spel"	
Actor Action	System Responsibility
1. Speler geeft aan het spel voortijdig te willen beëindigen	2. Het systeem stopt de timer
	3. Het systeem verwijdert de gegeven antwoorden van de speler
	4. Het systeem toont de speler de hoofdpagina

Use Case: Finch-shop

Primary Actor	- Speler
Stakeholders and Interests	- ?
Brief description	
<p><i>Een speler kan vragenlijsten kopen in de shop. Hiervoor moet een speler over genoeg saldo beschikken. Er zijn verschillende prijzen mogelijk voor de vragenlijsten. Het aankoopbedrag wordt verrekend met het saldo van de speler. Sommige vragenlijsten kunnen een jaar gebruikt worden. Een speler kan ook extra munten kopen, hier kan worden betaald via een online betaalservice.</i></p>	
Preconditions	<ul style="list-style-type: none"> - Er zijn vragenlijsten beschikbaar in de shop - Een gebruiker heeft zich geregistreerd - Een gebruiker is ingelogd
Postconditions (Success Guarantee)	<ul style="list-style-type: none"> - Een speler heeft een item gekocht - Het saldo van de speler is aangepast
Main Success Scenario (Basic Flow): “Kopen vragenlijst”	
Actor Action	System Responsibility
1. Een speler geeft aan iets in de shop te willen kopen	2. Het systeem toont de beschikbare items in de shop
	3. Het systeem toont het saldo van de speler
4. De speler selecteert een vragenlijst	5. Het systeem toont de prijs
	6. Het systeem toont de optie om de vragenlijst te kopen
7. De speler bevestigt de vragenlijst te willen kopen	8. Het systeem controleert of het saldo van de speler voldoende is
	9. Het systeem verrekent het aankoopbedrag met het saldo van de speler
	10. Het systeem registreert dat de speler toegang heeft tot de gekochte vragenlijst
	11. Het systeem legt vast dat de gebruiksdatum 1 jaar na aankoop is voor de betreffende vragenlijst
	12. Het systeem toont de shop interface weer

Extensions (Alternative Flow): <i>Speler heeft niet genoeg saldo</i>	
Actor Action	System Responsibility
1. Een speler geeft aan iets in de shop te willen kopen	2. Het systeem toont de beschikbare items in de shop
3. De speler selecteert een vragenlijst	4. Het systeem toont de prijs
	5. Het systeem toont de optie om de vragenlijst te kopen
6. De speler bevestigt de vragenlijst te willen kopen	7. Het systeem controleert of het saldo van de speler voldoende is
	8. Het systeem toont de speler een melding dat het saldo van de speler onvoldoende is
	9. Het systeem keert terug naar de shop interface
Extensions (Alternative Flow): <i>Speler wilt munten bijkopen</i>	
Actor Action	System Responsibility
1. Een speler geeft aan iets in de shop te willen kopen	2. Het systeem toont de beschikbare items in de shop
3. De speler selecteert het bijkopen van munten	4. Het systeem toont de prijs
	5. Het systeem toont de optie om de munten te kopen
6. De speler bevestigt de munten te willen kopen	7. Het systeem stuurt de speler door naar de online betaalservice
ZIE → Use Case: “Online Betaalservice”	
VERVOLG NA SUCCESVOL AFRONDEN USE CASE “ONLINE BETAALSERVICE”	
	8. Het systeem toont de speler dat de transactie is geslaagd
	9. Het systeem voegt het aantal gekochte munten toe aan het saldo van de speler
	10. Het systeem keert terug naar de shop interface

Functionele eisen

Registreren

1. Een gebruiker moet een unieke gebruikersnaam hebben.

Vragenlijsten

2. Een vragenlijst moet bestaan uit minimaal 10 vragen.
3. Een vragenlijst moet bestaan uit zowel open als meerkeuze vragen.
4. Een vragenlijst moet gekoppeld zijn aan een thema.
5. Er zijn ten minste twee vragenlijsten beschikbaar

Vragen

6. Een meerkeuzevraag moet bestaan uit 4 antwoorden.
7. Een vraag mag maar deel uitmaken van 1 vragenlijst.
8. Een open antwoord kan meerdere juiste antwoorden hebben.
9. Een speler mag maar 1 vraag tegelijkertijd beantwoorden.
10. Het systeem moet voor elke speler zijn beste score bijhouden per vragenlijst.

Toegang

11. Alleen medewerkers mogen toegang hebben tot het beheren van vragen(lijsten) en thema's.

Spelverloop quiz

12. Een speler kan een vraag niet overslaan.
13. Wanneer er geen antwoord wordt gegeven op een vraag wordt deze automatisch als "fout" beoordeeld.
14. Wanneer de tijd op is en er vragen zijn die niet beantwoord zijn, worden deze automatisch als "fout" beoordeeld.\
15. Een speler kan zijn antwoord op vorige vragen niet aanpassen.

Finch-shop

16. Vragenlijsten hebben een prijs.
17. Een speler kan niet over een negatief saldo beschikken.
18. De Finch-shop heeft niks te maken met de online betaalservice en ontvangt enkel of de transactie geslaagd is.

Non-functional Requirements

- 1 Het systeem moet later omzet kunnen worden naar een andere taal. (Replaceability?)
- 2 Het systeem mag geen vertraging hebben tijdens het spelen van een quiz van meer dan 0.1 seconde. (Time behaviour)
- 3 Het puntentel systeem moet makkelijk aanpasbaar zijn. (Replaceability?)