

Provo

Software Requirements Specification

v1.0

31 maart 2022

Alex Cheng (634967)

Rowan Paul Flynn (637588)

OOAD

Herman Telman

Docent

Inhoudsopgave

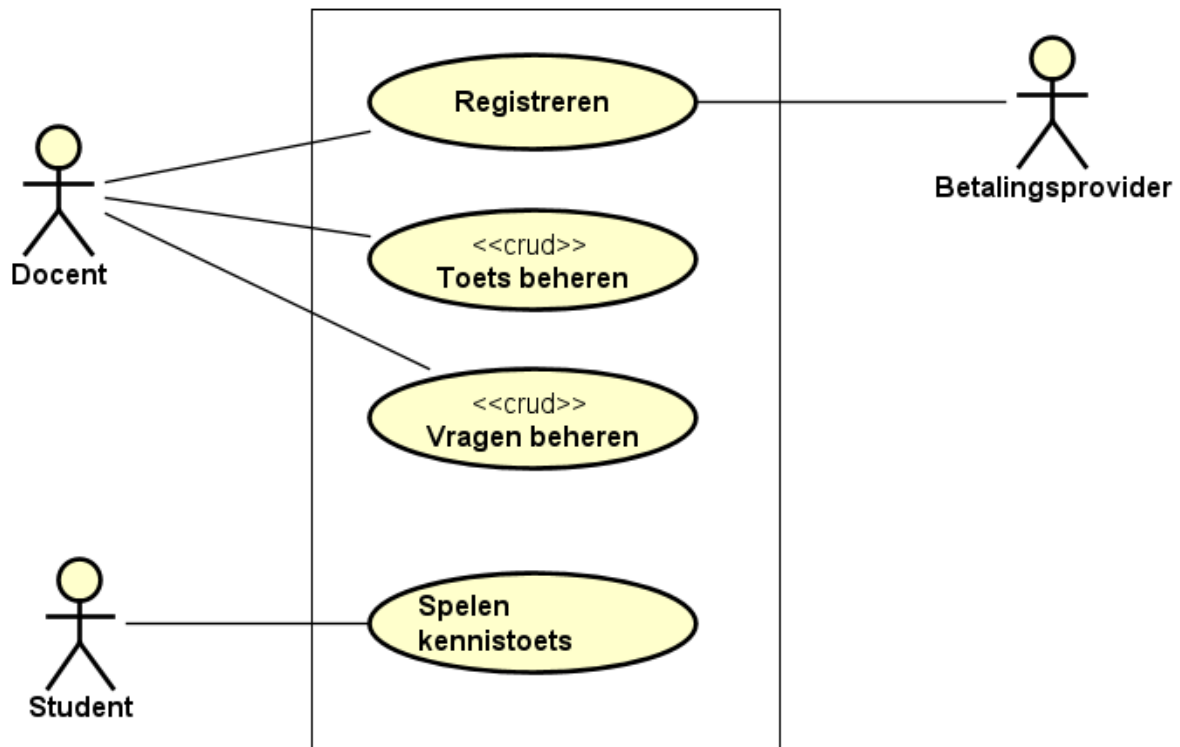
1	Introductie	2
1.1	Product functies	3
1.2.1	Registreren	3
1.2.2	Toets beheren	3
1.2.3	Vragen beheren	3
1.2.4	Spelen kennistoets	4
2	Domeinmodel	5
2.1	Glossary	6
3	Use Case beschrijvingen	7
3.1	Spelen kennistoets	7
3.2	Registreren	8
3.3	Toets beheren	9
4	Requirements	10
4.1	Functional requirements	10
4.2	Non functional requirements	10

1 Introductie

Dit document is het software requirements specification (SRS) van de applicatie Provo. Hierin staat de functionaliteit van de software beschreven en hoe de software zich moet gedragen. Dit gebeurt aan de hand van het beschrijven van use cases, een domeinmodel en het benoemen van requirements die niet benoemd zijn in de use cases.

De software die gemaakt moet worden is een applicatie genaamd Provo. In deze applicatie kunnen docenten kennistoetsen aanmaken en afnemen van studenten. Er zijn verschillende abonnementen voor de docent om uit te kiezen.

1.1 Product functies



Figuur 1: Use case diagram

Hierboven staat het use case diagram van Provo. Er zijn in totaal 4 use cases te onderscheiden. In de volgende alinea's geven we per use case een brief use case description.

1.2.1 Registreren

Een docent geeft aan zich te willen registreren. Vervolgens geeft de docent een e-mail (gebruikersnaam), wachtwoord, voornaam, achternaam en school/organisatie waarbij hij werkzaam is. De docent kan kiezen uit een gelimiteerd basis account of een premium account dat meer functionaliteit biedt.

1.2.2 Toets beheren

Een docent kan in een lokaal een kennistoets aanmaken. Deze kennistoets bestaat uit meerkeuzevragen, juist/onjuist vragen en ten slotte kort-antwoord-vragen. Wanneer een toets gemaakt is dan kunnen studenten zich aanmelden voor de toets.

1.2.3 Vragen beheren

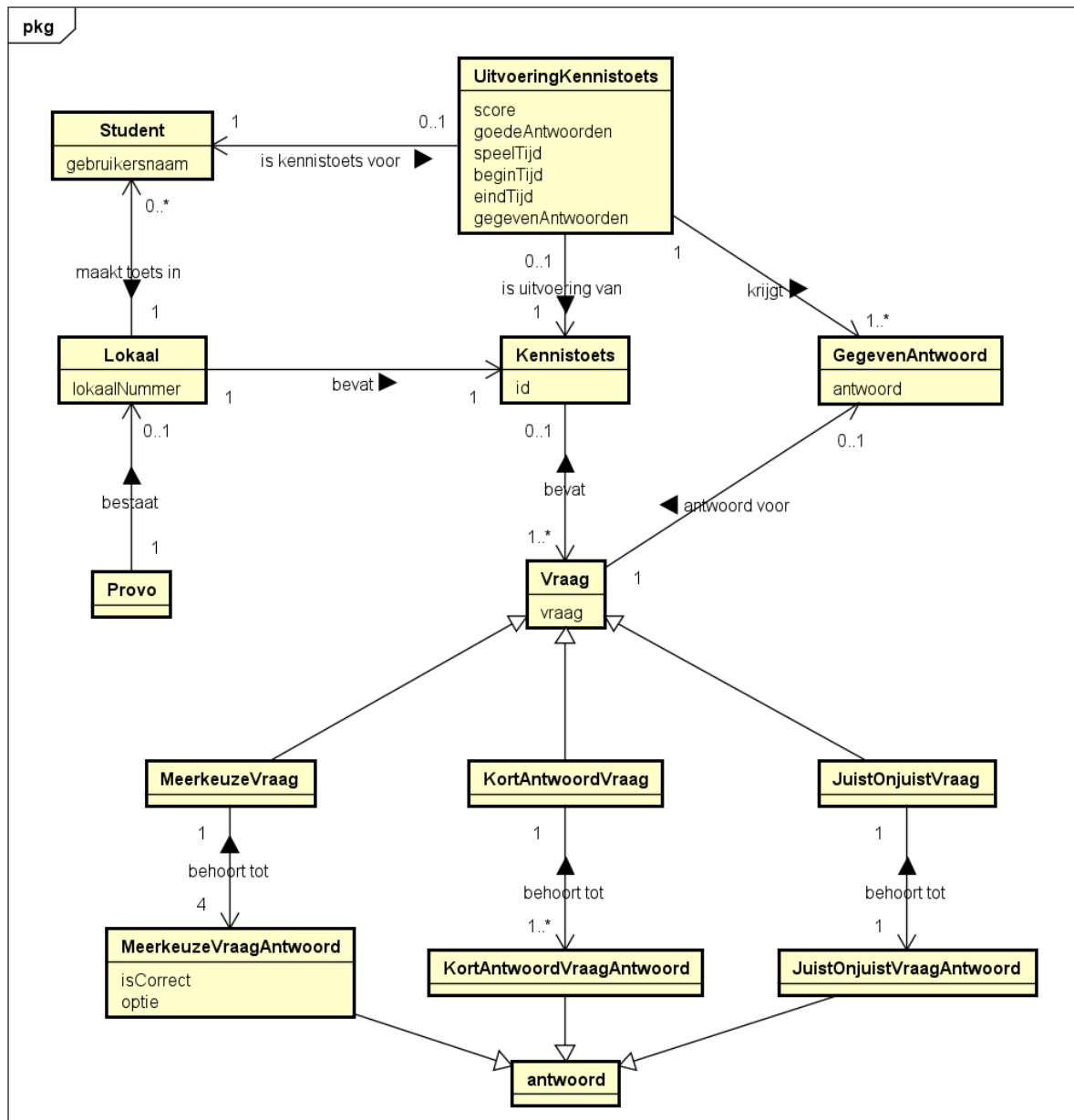
Een docent kan vragen aanmaken. Een vraag kan een meerkeuzevragen, juist/onjuist-vragen en kort-antwoord-vraag zijn. Bij een kort-antwoord-vraag moet de docent aangeven welke antwoorden correct zijn.

1.2.4 Spelen kennistoets

Een student meldt zich aan bij een lokaal en geeft een naam op. De student beantwoordt de vragen één voor één. Als de student wil kan hij terug naar een vorige vraag. Wanneer de student alle vragen heeft beantwoord kan hij aangeven dat hij klaar is. Op basis van de tijd krijgt de student punten per goed beantwoorde vraag.

2 Domeinmodel

In het domein model zijn de verschillende concepten van de applicatie te vinden. In de onderstaande afbeelding hebben wij alle belangrijke concepten benoemd van Provo. Ook zijn de onderlinge relaties en hun multipliciteit hierin verwerkt.



Figuur 2: Domeinmodel

2.1 Glossary

Concept	Beschrijving
Student	De student die de kennistoets maakt.
Lokaal	Een lokaal bestaat uit studenten en een docent, hierin wordt de kennistoets uitgevoerd.
Kennistoets	Kennistoets is de toets die uit vragen bestaat.
Provo	Provo is de applicatie die alle lokalen, studenten en toetsen bevat.
UitvoeringKennistoets	UitvoeringKennistoets is de specifieke uitvoering van een toets voor een student.
GegevenAntwoord	GegevenAntwoord is het antwoord voor een vraag dat de speler geeft.
Vraag	Een Vraag is een vraag die aan de speler wordt gesteld en die de speler moet beantwoorden.
MeerkeuzeVraag	Een MeerkeuzeVraag is een vraag waarbij vier opties van antwoorden horen waarvan er één goed is.
KortAntwoordVraag	Een KortAntwoordVraag is een open vraag.
Antwoord	Een Antwoord is een antwoord dat bij een vraag hoort.
MeerkeuzeVraagAntwoord	MeerkeuzeVraagAntwoord is een optie voor een MeerkeuzeVraag. Een MeerkeuzeVraag heeft vier MeerkeuzeVraagAntwoorden waarvan één correct is.
KortAntwoordVraagAntwoord	KortAntwoordVraagAntwoord is een antwoord dat goed wordt gerekend voor een KortAntwoordVraag.
JuistOnjuistVraag	JuistOnjuistVraag is een vraag die goed of fout is
JuistOnjuistAntwoord	JuistOnjuistAntwoord is het juiste antwoord van een JuistOnjuistVraag

3 Use Case beschrijvingen

3.1 Spelen kennistoets

Primary actor: Student	
Stakeholders and Interests: Educom	
Brief description: Een student meldt zich aan bij een lokaal en geeft een naam op. De student beantwoordt de vragen één voor één. Als de student wil kan hij terug naar een vorige vraag. Wanneer de student alle vragen heeft beantwoord kan hij aangeven dat hij klaar is. Op basis van de tijd krijgt de student punten per goed beantwoorde vraag.	
Preconditions: Er moet een lokaal met kennistoets beschikbaar zijn gesteld door de docent.	
Postconditions (Success Guarantee): De kennistoets is ingevuld en afgerond.	
Main Success Scenario (Basic Flow):	
Actor Action	System Responsibility
1. Student meldt zich aan bij een lokaal 2. Student geeft een naam op 4. Student beantwoordt vraag 5. Student gaat naar volgende vraag [Stappen 3-5 worden herhaald totdat alle vragen zijn ingevuld.] 6. Student geeft aan dat hij klaar is	3. Systeem weergeeft een vraag 7. Systeem controleert de antwoorden 8. Systeem berekent het aantal verdiende punten gebaseerd op de correcte antwoorden 9. Systeem berekent nog een bonus daarbovenop gebaseerd op de genomen tijd 10. Systeem toont de behaalde punten en beëindigt de kennistoets
Extensions (Alternative Flow):	
[Alternative Flow A: Student wilt naar een vorige vraag] 5A. Student gaat terug naar vorige vraag 4A. Student beantwoordt de vraag opnieuw	3A. Systeem weergeeft vorige vraag

3.2 Registreren

Primary actor: Docent	
Stakeholders and Interests: Educom	
Brief description: Een docent geeft aan zich te willen registreren. Vervolgens geeft de docent een e-mail (gebruikersnaam), wachtwoord, voornaam, achternaam en school/organisatie waarbij hij werkzaam is. De docent kan kiezen uit een gelimiteerd basis account of een premium account dat meer functionaliteit biedt.	
Preconditions: N.v.t	
Postconditions (Success Guarantee): De gegevens van de docent worden opgeslagen bij Provo.	
Main Success Scenario (Basic Flow):	
Actor Action	System Responsibility
1. Docent geeft op dat hij zich wilt registreren 2. Docent geeft e-mail, wachtwoord, voor- en achternaam en school/organisatie op 3. Docent selecteert basis account (gratis)	4. Systeem controleert of e-mail uniek is 5. Systeem slaat gegevens op
Extensions (Alternative Flow):	
[Alternative Flow B: Docent wilt premium account aanschaffen] 3B. Docent selecteert premium account 5B. Docent betaalt de factuur [Alternative Flow C: Betaling voor premium account mislukt] 5C. Docent betaalt de factuur	[Alternative Flow A: E-mail is al in gebruik] 4A. E-mail is niet uniek 5A. Systeem geeft terug dat het e-mail al in gebruik is 4B. Systeem stuurt docent door naar betalingsprovider 6B. Systeem toont aan dat de betaling gelukt is 6C. Systeem toont foutmelding dat de betaling mislukt is

3.3 Toets beheren

Primary actor: Docent	
Stakeholders and Interests: Educom	
Brief description: Een docent kan in een lokaal een kennistoets aanmaken. Deze kennistoets bestaat uit meerkeuzevragen, juist/onjuist vragen en ten slotte kort-antwoord-vragen. Wanneer een toets gemaakt is dan kunnen studenten zich aanmelden voor de toets.	
Preconditions: De docent heeft een bestaand account. De docent heeft nog een beschikbaar lokaal. Er bestaan al vragen voor de toets	
Postconditions (Success Guarantee): Lokaal bevat nu een toets.	
Main Success Scenario (Basic Flow):	
Actor Action	System Responsibility
1. Docent geeft aan een toets te willen maken in een lokaal 3. Docent voegt een of meerdere vragen toe 4. Docent geeft aan dat hij de kennistoets afgerond heeft	2. Systeem toont de beschikbare soorten vragen 5. Systeem slaat de betreffende kennistoets op in het lokaal
Extensions (Alternative Flow):	

4 Requirements

In dit hoofdstuk worden de functionele requirements besproken die niet voorkwamen in de use cases. De requirements zijn beschreven in FURPS+ omdat dit een goede “checklist” geeft van belangrijke zaken.

4.1 Functional requirements

Nummer	Requirement
1	Een docent kan zijn/haar profielgegevens aanpassen na de registratie
2	Er kunnen niet meer dan 50 studenten deelnemen aan een toets

4.2 Non functional requirements

Usability

Nummer	Requirement
1	Een docent kan 8 lokalen op hetzelfde moment gebruiken zonder problemen

Performance

Nummer	Requirement
1	Het systeem moet 99% van de systeem operaties uitvoeren binnen 2 seconden

Supportability

Nummer	Requirement
1	De applicatie moet binnen een week kunnen worden uitgebreid met een extra taal
2	De puntentelling kan vervangen worden binnen een week
3	De applicatie werkt op desktop en mobile