Analysedocument

Inhoudsopgave

Inleiding

Ik ga een applicatie maken voor een basisschool waarin leerlingen hun cijfers kunnen zien. De docenten hebben een overzicht van de cijfers van hun leerlingen en kunnen deze cijfers zelf invoeren en aanpassen. Ten slotte is er ook nog een Beheerder die alle leerlingen en docenten kan toevoegen aan de school.

Alle plaatjes die je hier voorbij ziet komen zijn scherper terug te vinden op de volgende website: https://drive.google.com/file/d/1myO5rFRrB2E-cWTtjxkCStxewHgBqApc/view?usp=sharing

Requirements

Ik heb alle functionele requirements in een tabel gezet en opgedeeld in Must, Should, Could en Won't. Verder heb ik er nog kwaliteitseisen en beperkingen bijgezet. Requirements die zijn gerealiseerd zijn groen gemarkeerd. Requirements die ik nog van plan ben te realiseren, maar niet heb gerealiseerd heb ik rood gemarkeerd.

Must have		
FR-01: Beheerder kan gegevens van een leerling toevoegen aan de school	K-01.1: Bij het toevoegen van een leerling moeten de volgende gegevens ingevoerd worden: Naam, geboortedatum, telefoonnummer en emailadres.	
FR-15: Docent moet toetsen kunnen aanmaken.	K-15.1: Bij de toets kan de docent aangeven wat de weging is	B-15.1: Weging mag niet minder dan 0 zijn en niet meer dan 100 zijn.

FR-14: Docenten kunnen cijfers van leerlingen invoeren.	K-14.1: Bij het cijfer kan de docent vermelden over welke toets het gaat. K-14.2: Als het cijfer wordt ingevoerd moet de datum van invoering zichtbaar zijn.	B-14.1: Een cijfer mag niet minder dan 1.0 zijn en niet meer dan 10.0 zijn.
FR-03: Leerling kan zijn eigen cijfers zien	K-03.1: leerling ziet meerdere cijfers per vak. K-03.2: Je moet je gemiddelde cijfer per vak kunnen zien	
FR-02: Beheerder kan een schoolklas aanmaken.	K-02.1: Elke schoolklas heeft een naam die eruitziet als "Groep " + getal + letter. K-02.2: Elke klas heeft ook een schooljaar. (Zoals eerste schooljaar, of Groep 3)	B-01.1: Als "Groep 3a" al bestaat kan die niet opnieuw aangemaakt worden
FR-11: Beheerder kan leerlingen in een schoolklas zetten.		B-11.1: De beheerder kan geen leerlingen toevoegen als er al 5 leerlingen in een klas zitten
FR-13: Beheerder kan een docent in een schoolklas zetten.		B-13.1: De beheerder kan geen docent in een schoolklas zetten, als er al een docent in een schoolklas zit.
FR-16: Beheerder kan gegevens van docent toevoegen aan school	K-01.1: Bij het toevoegen van een docent moeten de volgende gegevens ingevoerd worden: Naam, geboortedatum, telefoonnummer, emailadres en ID.	
Should have		
FR-12: Toetsen kunnen verschillende wegingen hebben.		B-12.1: Weging mag niet minder dan 1 zijn en niet meer dan 25 zijn. B-12.2: weging kan minimaal met een heel getal omhoog.
Could have		

FR-04: Roostermaker kan de roosters maken	K-04.1: Het lokaal wordt bij de les vermeld.	B-04.1: niet meerdere vakken op een moment.
		B-04.2: niet meer dan 8 lessen inplannen op een dag.
		B-04.3: zo min mogelijk tussenuren
FR-05: Roostermaker kan de roosters docentenrooster uploaden, deze moet gezien worden door docenten.		
FR-06: Roostermaker kan de roosters leerlingenrooster uploaden, deze moet gezien worden door leerlingen.		
FR-07: Docent moet een aantekening kunnen maken voor een leerling.		
FR-08: De roostermaker moet lessen kunnen laten uitvallen, dit moet dan visueel zichtbaar zijn voor docenten en leerlingen.		
FR-09: Docent moet kunnen aangeven op welke dag er een toets is, leerlingen moeten dit vervolgens kunnen zien.		
Won't have		
FR-10: Elkaar berichten kunnen sturen binnen de app.		

Verder heb ik nog één non-functionele requirement.

NFR-01: interface moet gebruikersvriendelijk zijn. Foutmeldingen geven als er iets fout gaat.

Use-cases

Gebruiker is niet ingelogd. Docenten, leerlingen en beheerder zijn ingelogd.

Ik heb geprobeerd om de wat onduidelijkere requirements, duidelijk te maken met behulp van use-cases.

UC-01	FR-15: Docent moet toetsen kunnen aanmaken. FR-02: Docenten kunnen cijfers van leerlingen invoeren.
Samenvatting	Docent kan een toets aanmaken, per leerling kan hij voor deze toets een cijfer geven.
Actoren	Docent
Aannamen	Actor is op het scherm waarin hij een toets kan aanmaken
Scenario	1. Systeem vraagt naar de naam en de weging van de toets.

	2. Actor vult de gevraagde gegevens in en bevestigt. 3. Systeem controleert gegevens en slaat ze op [1] 4. Docent navigeert naar het scherm waar hij een toets kan kiezen waarvan hij de cijfers wil aanpassen 5. Systeem vraagt om welke toets het gaat. 6. Actor kiest een toets en bevestigt. 7. Systeem vraagt per leerling een cijfer. 8. Actor vult bij elke leerling een cijfer in en bevestigt. 9. Systeem controleert cijfers en slaat ze op [2] [3]
Uitzonderingen	1. Als waarde niet tussen 1.0 en 25.0 zit, laat foutmelding zien en ga terug naar stap 1. 2. Als er een veld niet is ingevuld, geef foutmelding en ga terug naar stap 7. 3. Als een cijfer een waarde buiten 1.0 en 10.0 heeft, geef foutmelding weer en ga terug naar stap 7.
Resultaat	Cijfers zijn nu opgeslagen en zijn klaar om getoond te worden aan de leerlingen.

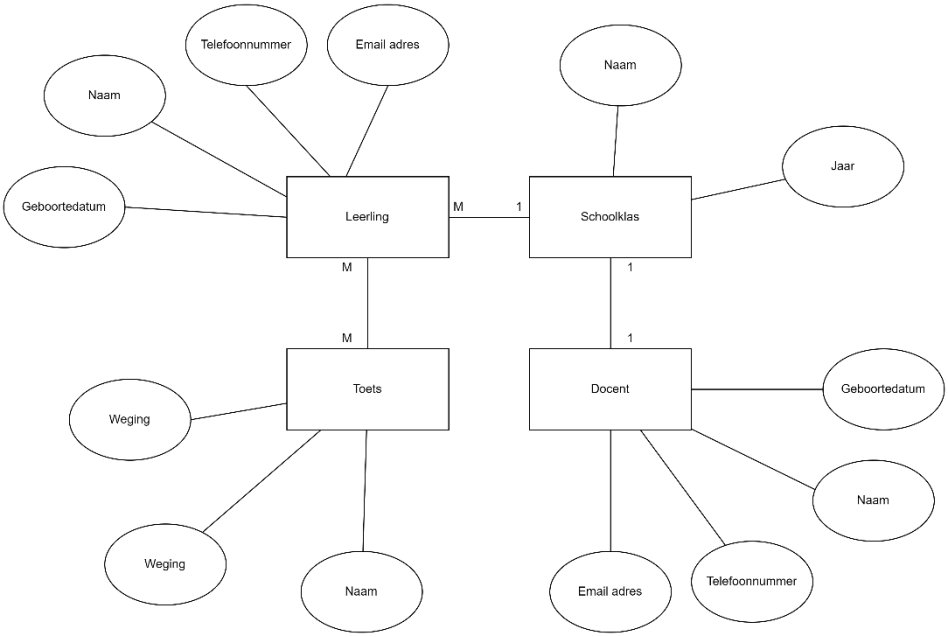
UC-02	FR-14: Beheerder kan een schoolklas aanmaken. FR-11: Beheerder kan leerlingen in een schoolklas zetten.
Samenvatting	Beheerder moet een klas maken waar hij vervolgens leerlingen in kan toevoegen.
Actoren	Beheerder
Aannamen	Actor is op het scherm waar hij schoolklassen kan aanmaken.
Scenario	1. Actor vult schoolklas naam in en bevestigt. 2. Systeem controleert de naam en voegt de klas toe. [1] 3. Actor navigeert naar het scherm waar je leerlingen kunt toevoegen aan de klas 4. Systeem vraagt om leerling 5. Actor kiest een leerling en bevestigt. 6. Systeem controleert hoeveel leerlingen er in de klas zitten en voegt toe. [2]
Uitzonderingen	1. De naam van de klas moet beginnen met Groep en dan een getal (Bijvoorbeeld 'Groep 6'). 2. Als er al 30 leerlingen in een klas zitten kan er niet nog een leerling toegevoegd worden. Laat foutmelding zien.
Resultaat	Er is nu een schoolklas met leerlingen en docenten.

UC-03	FR-01: Beheerder kan gegevens van een leerling toevoegen aan de school
Samenvatting	Beheerder kan gegevens van een leerling toevoegen aan de school
Actoren	Beheerder
Aannamen	Actor is op het scherm waarin hij een leerling kan toevoegen aan de school
Scenario	1. Systeem vraagt naar gegevens van de leerling 2. Actor vult gegevens in van de leerling en bevestigt. [K-01.1] 3. Systeem slaat gegevens op [1] [2]
Uitzonderingen	1. Als niet alle velden zijn ingevuld dan moet er een foutmelding komen ga terug naar stap 1. 2. Als er in een veld de verkeerde informatie in ingevuld, geef foutmelding en ga naar stap 1.
Resultaat	Er is een leerling toegevoegd aan de school

UC-04	FR-03: Leerlingen kunnen hun eigen cijfers opvragen.
Samenvatting	Een leerling kan zijn eigen cijfer zien
Actoren	Leerling
Aannamen	Actor is ingelogd met zijn eigen account
Scenario	1. Leerling navigeert naar het overzicht van zijn cijfers. 2. Systeem laat alle tot nu toe behaalde cijfers zien.
Uitzonderingen	
Resultaat	Leerling heeft zijn eigen cijfers bekeken

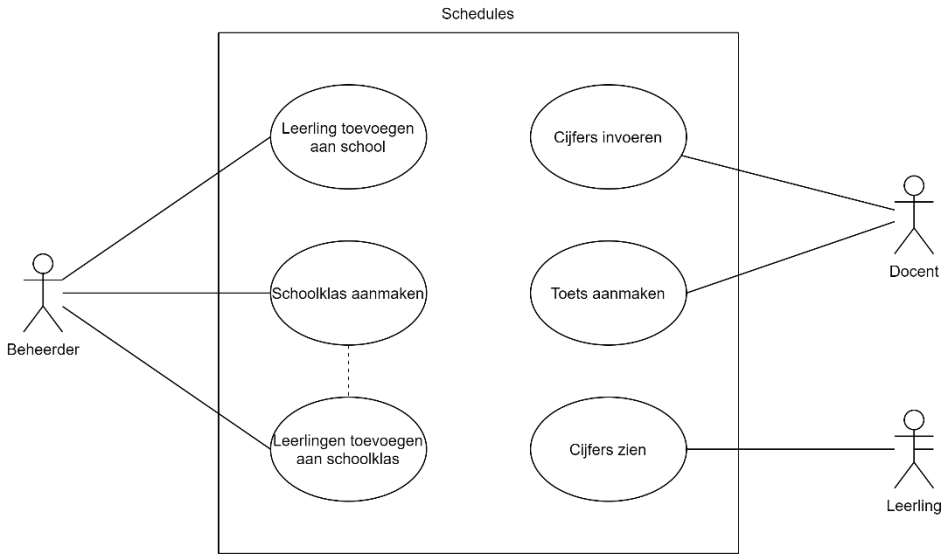
Conceptueel Model

Erg vanzelfsprekend.



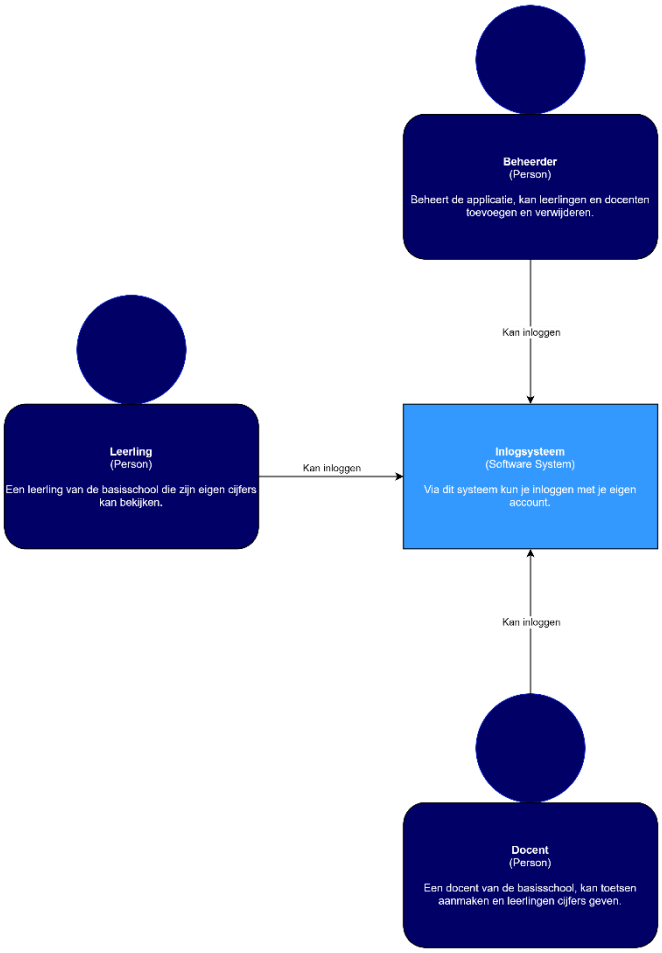
Use-Case Diagram

Ik heb de use case diagram opgesteld met behulp van mijn use cases. Je ziet een stippellijn tussen schoolklas aanmaken en leerlingen aan schoolklas toevoegen, dat is omdat ze beide in dezelfde use case zitten.



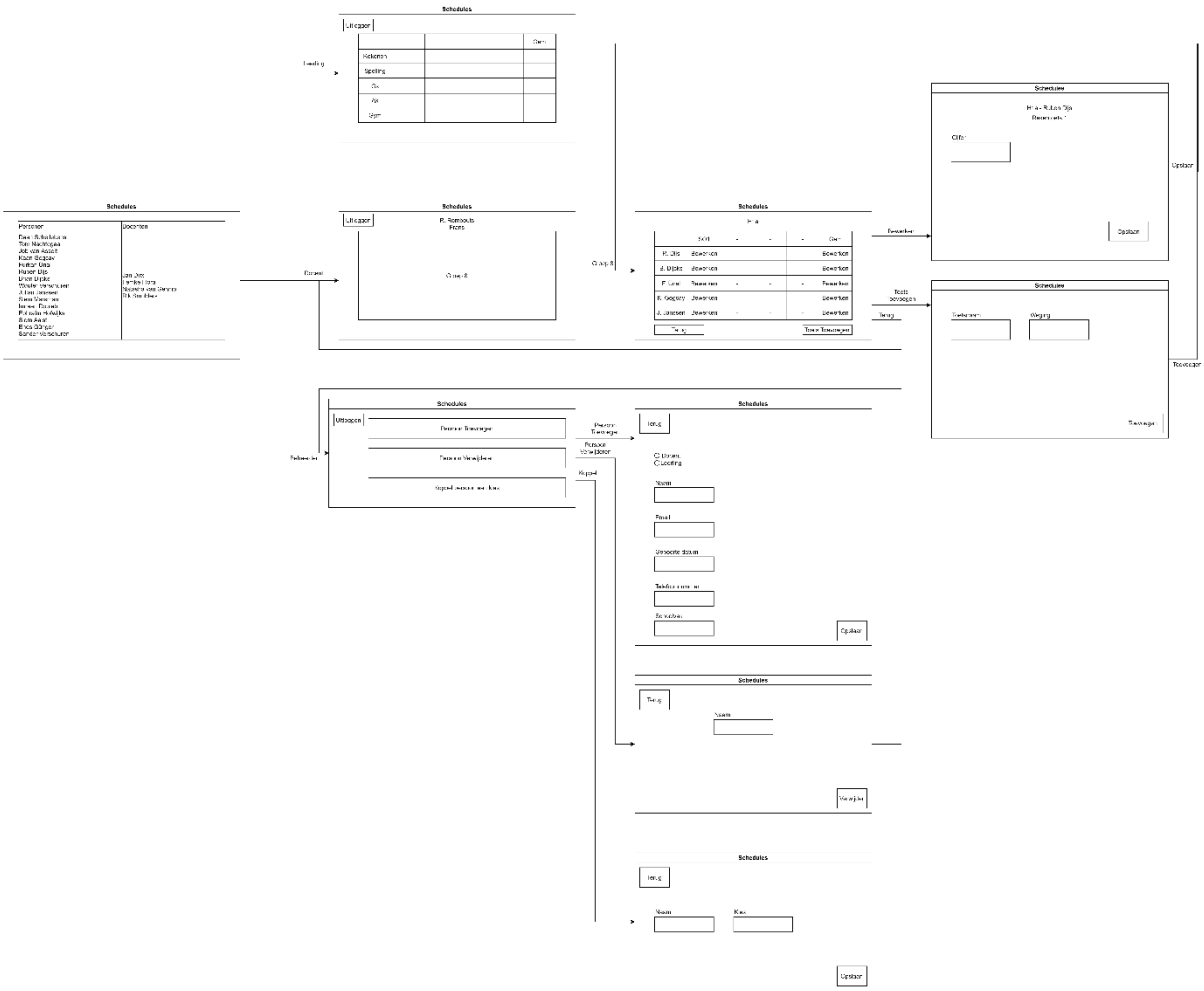
Context Diagram

Naast het inlogsysteem heb ik geen andere systemen in mijn applicatie. Ook gebruik ik geen externe API.



Wireframe

Ik heb wat wireframes gemaakt, dit geeft mij en de klant een duidelijker overzicht van hoe de applicatie eruit komt te zien. Echter zou ik wel adviseren om ze te bekijken via de bijlage.



Use Case	
Summary	Play a full round of Reverse Hangman Online
Actors	P1 (player 1), P2 (player 2)
Assumptions	Player 1 and Player 2 are sitting next to each other. Player 1 and Player 2 are in the menu screen of Reverse Hangman Online

Scenario	1. P1 presses on the multiplayer button 2. Frontend shows multiplayer screen 3. P1 types in a teamname 4. P2 types in a teamname 5. P2 confirms the teamnames 6. Frontend gives values to backend 7. Backend saves teamnames 8. Frontend shows game screen 9. P1 types in a word and confrims 10. Frontend gives values to backend 11. Backend calculates properties 12. Backend sends properties to frontend 13. Frontend shows properties on screen 14.
Uitzonderingen	4. Als waarde niet tussen 1.0 en 25.0 zit, laat foutmelding zien en ga terug naar stap 1. 5. Als er een veld niet is ingevuld, geef foutmelding en ga terug naar stap 7. 6. Als een cijfer een waarde buiten 1.0 en 10.0 heeft, geef foutmelding weer en ga terug naar stap 7.
Resultaat	Cijfers zijn nu opgeslagen en zijn klaar om getoond te worden aan de leerlingen.