
RAPPORT

ANALYSE RAPPORT

23/11/2022

Het ontwikkelen van een exploratietool voor het dynamisch ontdekken van opleidings- en toekomstmogelijkheden

Auteurs: Valérie Becquart
Michal Davidse
Kevin Vandeputte
Stef Vanbroekhoven



Inhoudsopgave

Analyse en onderzoek	3
1 Bestaande tools op Vlaams niveau	3
2 Spelconcept	5
2.1 Verhaal	5
2.2 Opzet en verloop	5
Spelregels	5
2.3 Activiteiten Diagram	6
3 Functionele eisen	9
3.1 Usecase diagram	9
3.2 De student	10
3.3 De platformbeheerder	10
3.4 IT beheerder Thomas More	10
4 Niet functionele eisen	10
4.1 Implementatie	10
4.1.1 Hosting	10
4.1.2 Combell	11
4.1.3 Firebase	11
4.1.4 Back-end Framework	12
4.1.5 Database	14
4.1.6 Front End frameworks	16
4.2 Externe interface	18
4.3 Performantie	18
4.3.1 Antwoordtijden	18
4.3.2 Schaalbaarheid	18
4.4 Kwaliteitseisen	18
4.4.1 Beveiliging	18
4.4.2 Gebruiksvriendelijkheid	18
5 Game engines en frameworks	19
5.1 Gdevelop.io	20
5.2 Godot	21
5.3 Unity	22
5.4 GameMaker Studio 2	23
5.5 WRM van verschillende platformen:	24
6 Skills studenten Thomas More ifv project coordinatie	26
7 Analyse & onderzoek – conclusie	27
VOORONTWERP	28
8 Technisch plan van aanpak: realisatie	28
8.1 MoSCoW	28
8.2 Datamodel	28
8.3 Planning	29
9 Vertaling Thomas More opleidingen naar Metrolijnen	30
10 Prototype:	31
10.1 Prototype visualisatie	31

ANALYSE EN ONDERZOEK

1 Bestaande tools op Vlaams niveau

Aan de hand van de websites van de 16 hogeschool instellingen (<https://data-onderwijs.vlaanderen.be/onderwijsaanbod/lijst.aspx?n=3&hz=true&hs=411>) en 6 universitaire instellingen (<https://data-onderwijs.vlaanderen.be/onderwijsaanbod/lijst.aspx?n=3&hz=true&hs=511>) werden bestaande tools opgezocht en geanalyseerd hoe deze werken.

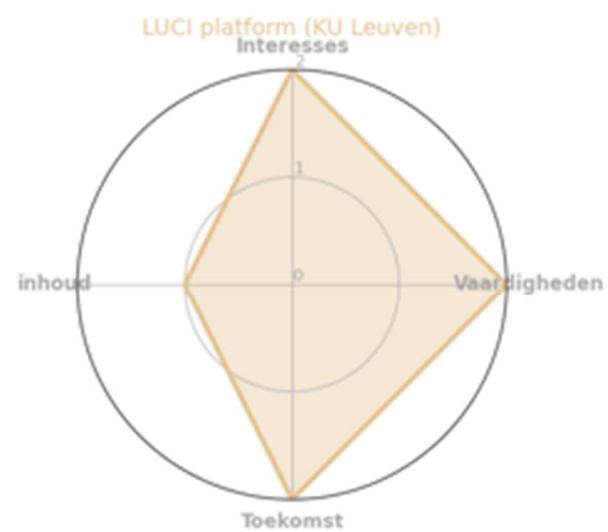
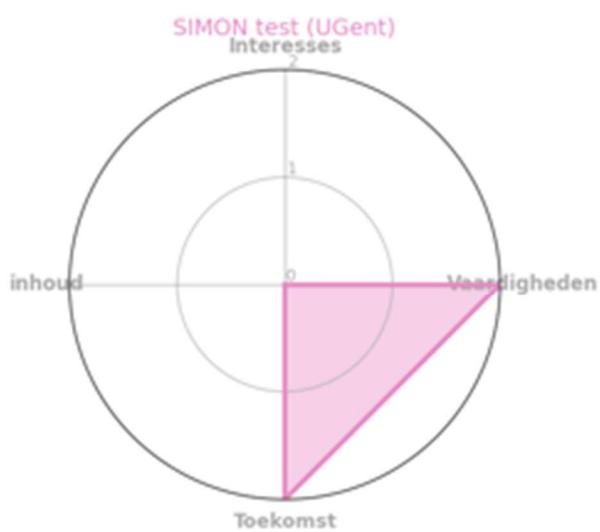
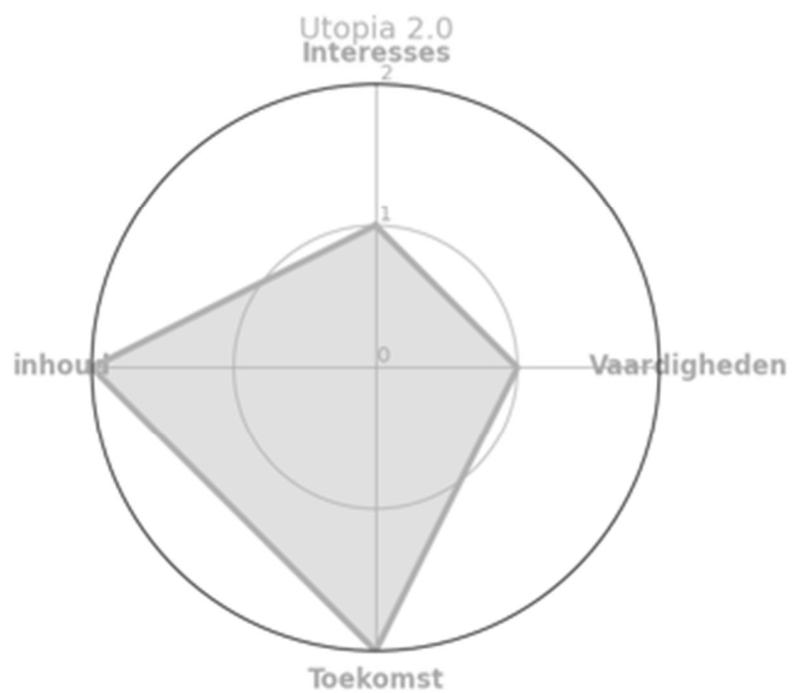
De belangrijkste tools werden verder geanalyseerd. We konden concluderen dat deze tools voornamelijk op enkele belangrijke pijlers werken in de begeleiding van de studiekeuze:

1. Interesses (Wie ben ik)
2. Vaardigheden (Wat kan ik)
3. Toekomst (Wat wil ik)
4. Inhoud (Wat is het)

De belangrijkste tools werden, aan de hand van hun omschrijving, ingeschaald op deze pijlers om zo na te gaan waar deze zich positioneren. Hierbij werden volgende scores gegeven:

- 0 = tool houdt er geen rekening mee
- 1 = tool houdt er enigszins rekening mee
- 2 = hoofdfocus

Daarnaast werd Utopia 2.0 tegen het licht gehouden om te vergelijken hoe de nieuwe tool zich zal verhouden tot reeds bestaande tools.



2 Spelconcept

2.1 Verhaal

De speler krijgt een kaart van metrolijnen voor zich. Het startpunt is een Central Station en van hieruit moet de speler navigeren naar zijn proclamatie feestje. Het proclamatie feestje ligt op de lijn van zijn voorkeursgebied. Door vragen correct te beantwoorden in een metrostation, kan hij punten vergaren. Vanaf dat hij 180 studiepunten bijeengesprokkeld heeft krijgt hij toegang tot zijn proclamatie bestemming. De lengte en weg van de reis bepaalt hijzelf. Maar kies wijselijk, want de eerste 2 gekozen metrostations op een metrolijn zijn dubbele punten waard.

2.2 Opzet en verloop

De gebruiker start in het Centraal station(Punt A).

Nadat de metrokaart zichtbaar wordt, licht het eerste punt op. De gebruiker krijgt een korte inhoud van de richting en daarbij ook de keuze om te spelen of deze over te slaan.

- Keuze spelen:
 - Er wordt een mini wereld getoond. De huizen corresponderen met de studierichting. In deze wereld kan de speler rondlopen en mensen aanspreken. Deze mensen stellen vervolgens een vraag.
 - Antwoord juist: De gebruiker krijgt punten
 - Antwoord fout: De gebruiker ontvangt geen punten
 - Via een knop informatie kan informatie over de studierichting gevraagd worden. Deze popup sluiten kan door 2 knoppen.
 - Verder spelen
 - Reis verder zetten => terug naar de metrokaart. Zie eveneens knop 'overslaan'
 - Er is ook optie om email adres op te geven. Hierbij ontvangt de speler de informatie van de studierichting.
- Keuze overslaan: het volgende punt op de metrolijn wordt opgelicht en de vraag springt opnieuw op.

Wanneer het punt een kruispunt is, krijgt de gebruiker de mogelijkheid om

- Keuze spelen
- Keuze verder spelen lijn huidige lijn -> eerst volgende punt van de lijn licht op
- Keuze verder spelen lijn B -> eerst volgende punt op de gekozen lijn springt op.

Spelregels

Na het beantwoorden van een vraag, verdwijnt de vraag uit de wereld.

De gebruiker krijgt dubbele punten wanneer dit het eerste of tweede metropunt is dat hij speelt.

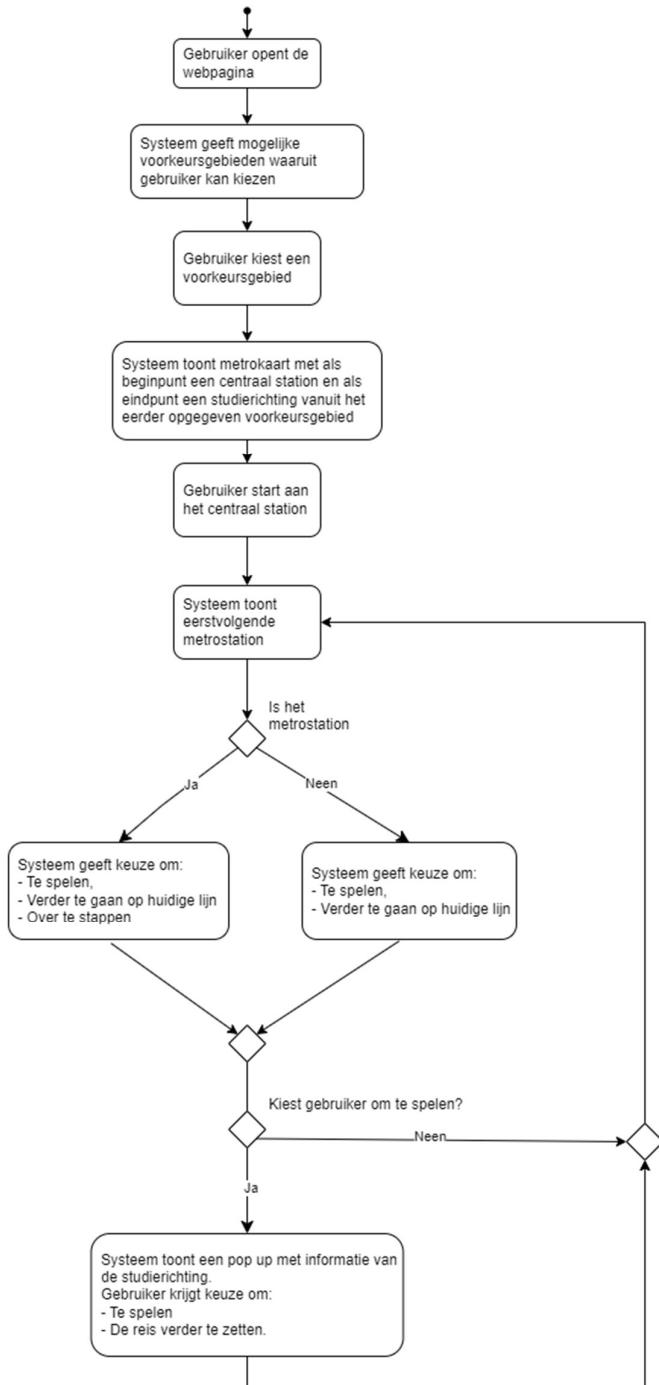
Na het vroegtijdig verlaten van een metrostation => dan krijgt de gebruiker geen punten.

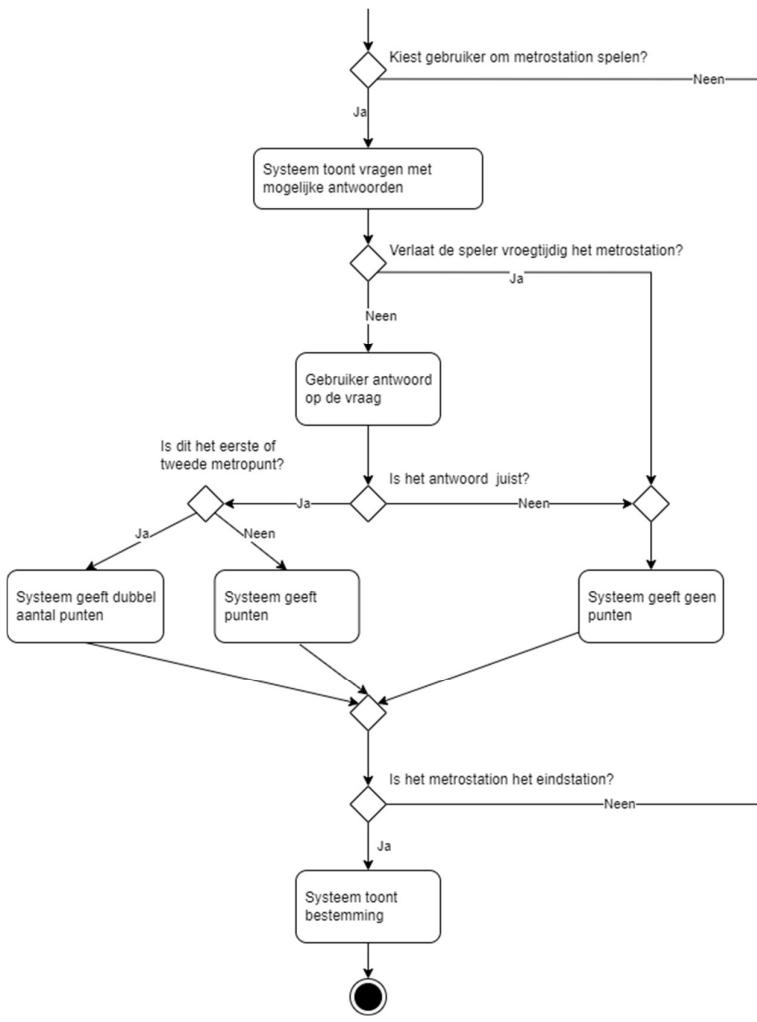
2.3 Activiteiten Diagram



Dit schema vertaald de verschillende use cases naar de flow van het systeem. Van start tot einde worden de handelingen en keuzes gemaakt, welke invloed hebben op de werking kunnen beïnvloeden.

Hieronder het spelconcept en de nodige acties gegoten in een activiteiten diagram.





3 Functionele eisen

In de functionele eisen leggen we de nadruk op wat de mogelijkheden, functies/kenmerken van ons systeem moeten zijn.

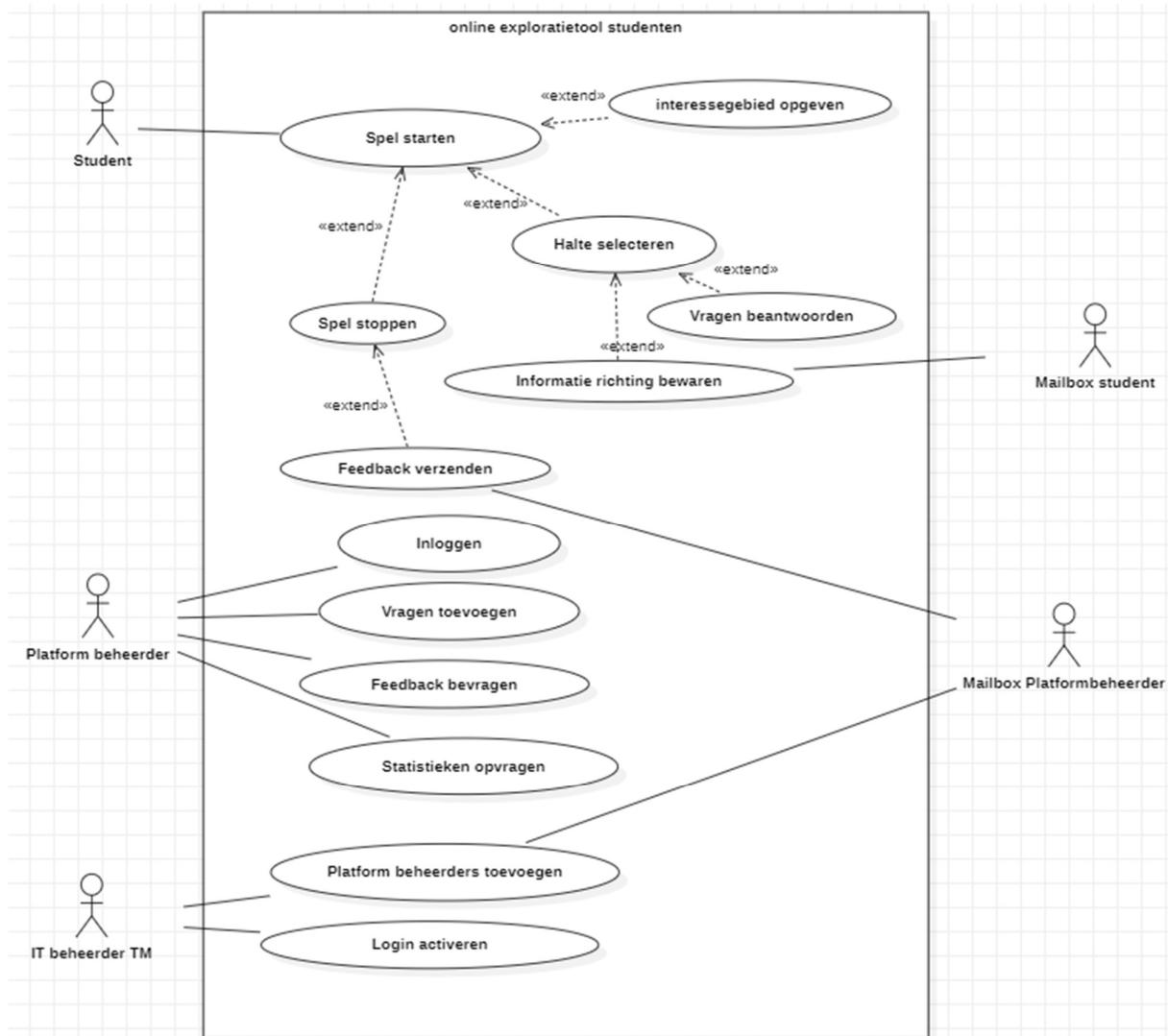
Het systeem kunnen we omschrijven als een online exploratietool voor studenten.

Als gebruikers kunnen we drie actieve gebruikers onderscheiden: de student, platformbeheerder en de IT beheerder van Thomas More.

3.1 Usecase diagram



Dit diagram legt de focus om schematisch de verschillende interacties van de actoren met het systeem in kaart te brengen.



3.2 De student

De student kan het spel starten en op elk moment stoppen.

De student kan het spel spelen zonder in te loggen.

Hij/zij kan bij de start een voorkeursgebied ingeven.

De student kan een metrostation selecteren. In dit metrostation kan hij/zij vragen beantwoorden of kiezen om het station te verlaten.

Als de student het spel heeft uitgespeeld kan hij de spelresultaten laten opsturen naar zijn e-mail adres.

De student kan ook feedback geven op het platform.

3.3 De platformbeheerder

De beheerder kan inloggen in het platform.

De beheerder kan nadat hij/zij vragen toevoegen aan een bepaalde studierichting.

Verder kan hij/zij feedback bevragen aan de student, en data/statistieken vanuit het systeem opvolgen/interpreteren.

Als de beheerder wordt toegevoegd aan het platform zal hij hiervan melding krijgen in zijn /haar mailbox. Als de student feedback geeft zal de platformbeheerder hiervan een melding krijgen in zijn/haar mailbox.

3.4 IT beheerder Thomas More

De IT beheerder is diegene die platformbeheerders kan toevoegen aan het platform en logins kan activeren.

4 Niet functionele eisen



Niet functionele eisen leggen we de nadruk op wat ons systeem moet zijn.

- *Hosting: Wie de tool voorziet.* Waar de gebruiker heen moet om de tool te gebruiken.
- *Back-end: Wat de gebruiker niet ziet.* Zoals functionele programmeerlogica...
- *Front-end: wat gebruiker ziet.* Zoals de website in zijn browser...
- *Database: Waar alles wordt opgeslaan.* Zoals een tabel met de vraag en antwoorden...
- *Framework: Hulpmiddelen om tijd te sparen.* Als het ware kant en klare bouwstenen...

4.1 Implementatie

4.1.1 Hosting

Aan de start van de conceptfase mochten we contact opnemen met Jochen Mariën (Thomas More) voor de opzet van de IT architectuur en hosting mogelijkheden. Van hem kregen we onderstaande feedback:

"Er is een complicatie: de school wil het de website alleen hosten als de dienst marketing eerst heeft gezien of de opmaak conform de regels van thomas more is. Je testomgeving hosten via de school is dus niet haalbaar, de productie alleen na controle door hen (dus niet op korte termijn).

Daarom begin je best te ontwikkeling op SIN. Als je nog geen account hebt vraag je dat best na via sin@thomasmore.be.

Als je wil dat de website ooit wel via thomas more gehost wordt, en of dat moet en hoever je daarin mag gaan vraag je best eerst na bij de opdrachtgever, controleer je bij webmaster@thomasmore.be aan welke regels je moet voldoen."

De focus van onze realisatiefase zal liggen op het hosten enerzijds voor testing /ontwerpen anderzijds om feedback te krijgen van middelbare school studenten. Daarom onderzochten wij hosting op zowel Combell als Firebase.

4.1.2 Combell

Via het academic software kunnen wij 1 Combell hosting per jaar activeren.

De mogelijkheden en eigenschappen van Combell

Hosting:

- 50 GB aan SSD web space
- 5x 250MB MySQM databases
- 50x25GB mailboxes + ongelimiteerd aantal e-mail adressen.
- We kunnen de domeinnaam voor 3 jaar gebruiken
- Makkelijk en intuïtief content management systeem via one-click installaties

Security & performance

- SSL certificaat zit mee in alle hosting packages
- Combell voorziet antivirus software en firewalls voor de hosting
- Combell garandeert dat de hosting 99.99% van de tijd online zal zijn
- Performantiedata kan opgevolgd door de beheerder

Restricties:

- Enkel BE-domeinnamen kunnen gekozen worden
- SSD web space is minder snel als SSD web hosting.
- Geen ondersteuning voor niet-relationele databases

4.1.3 Firebase

Firebase is een platform van Google, en biedt hostingservices aan voor verschillende soorten toepassingen. Firebase biedt een gratis pakket aan met

- 50 000 gebruikers/maand
- 10 000 login authenticaties per maand
- 1Gb aan data worden opgeslagen in een NoSQL database.
- SSL certificaat aanwezig
- Er kunnen 100 connecties tegelijk met de database gelegd worden
- Domeinnaam is subdomein van web.app and firebaseapp.com
-

Firebase wordt aangeraden als hosting voor web en mobiele apps, alsook voor games. Ook tijdens de ontwikkelingsfase van een project wordt firebase vaak gebruikt.

Daarom kiezen we voor Firebase.

Indien in een latere fase het spel gehost zal worden voor gebruik zal men eerst een onderzoek moeten doen naar de regels/eisen waaraan het spel moet voldoen. Hiervoor verwijzen wij (naast de hierboven vermelde webmaster@thomasmore.be) ook graag door naar de studie marketing (zie ook punt 7 in dit document).

Bronnen

<https://www.combell.com/en/>

<https://firebase.google.com>

4.1.4 Back-end Framework

De keuze welk back-end framework gebruikt zal worden maken we op basis van de 'Weighted Ranking Method'. We stellen een aantal criteria op waaraan het framework moet voldoen en geven een score van 1-100. Voor elk criteria bepalen we daarnaast een gewicht, dat aangeeft hoe belangrijk het criterium is voor ons project.

We wegen vier frameworks tegen elkaar af. Deze frameworks zijn deel van het curriculum van de bachelor toegepaste informatica aan de Thomas More hogeschool. Deze keuze werd bewust gemaakt omdat het project na onze bijdrage zal verdergezet worden door studenten. Zo hoeft er niet veel tijd verloren te gaan om het framework onder de knie te krijgen.

De evaluatiecriteria zijn terug te vinden in de tabel. Naast de standaard vereisten is het voor ons belangrijk dat de back-end een API kan opzetten waarmee de front-end kan communiceren. Daarnaast gezien de grootte van het project moet het schaalbaar zijn en eenvoudig in gebruik. Als laatste nemen we de kostprijs ook in rekening als een belangrijke factor.

Criteria voor selectie van kandidaten:

- Ondersteuning van een front-end framework
- Scalability
- Bekend bij studenten aan de hogeschool
- Eenvoudig
- Kostprijs

Te onderzoeken frameworks:

- 1 Spring (Java)
- 2 FastApi (Python)
- 3 .NET Core (C#)
- 4 Laravel (PHP)

Requirements	Weight (/10)	Scores (%)				Verantwoording
		Spring (Java)	.NET Core	Laravel	FastAPI	
REST API	10	100%	100%	100%	100%	Al deze frameworks kunnen een REST API opzetten.
Schaalbaarheid	10	100%	100%	100%	70%	Doordat FastAPI nog vrij jong is, loopt het wat achter op gebied van schaalbaarheid. De andere, langer bestaande frameworks, zijn hier allemaal goed voorzien.
Performantie	10	100%	80%	60%	100%	FastAPI is erg snel en blinkt uit in performantie.
Gebruiksvriendelijkheid	5	90%	80%	70%	100%	FastAPI laat je erg snel een API opzetten. Bij de andere frameworks vraagt dit wat meer werk.
Kost	7	100%	100%	100%	100%	Alle frameworks zijn open source en dus gratis.
Documentatie en community support	7	100%	100%	100%	70%	FastAPI is redelijk recent en heeft daarom een kleinere community. De documentatie daarentegen is van erg hoog niveau. De andere frameworks bestaan al lange tijd en hebben daarom veel meer documentatie en community support.
Microservice architectuur	8	100%	100%	60%	100%	Laravel voorziet een microservice architectuur, maar heeft een hogere latency.
Container support	6	100%	100%	100%	100%	Alle frameworks kunnen opgezet worden in een container service.
Security	3	100%	100%	100%	100%	Alle frameworks voorzien in hun API de mogelijkheid voor beveiliging met verschillende protocollen.
Total	66	65.5	63	57.3	60.9	



API: Een API (Application Programming Interface) is een software-interface die het mogelijk maakt dat twee applicaties met elkaar kunnen communiceren. Je zou kunnen zeggen de boodschapper die een verzoek indient bij een provider en vervolgens het antwoord terugkoppelt.

Conclusie:

Uit de analyse komt Spring als beste eruit. Het framework is performant, schaalbaar en er is heel uitgebreide documentatie. Het voldoet dus aan alle belangrijkste vereisten. Daarnaast zit het framework in het curriculum van Thomas More, waardoor toekomstige projectteams geen leercurve moeten doorstaan om verder te werken aan het project.

Bronnen

- <https://www.slant.co/topics/1397/~best-web-frameworks-to-create-a-web-rest-api#19>
- <https://hrcak.srce.hr/219995>
- <https://www.snapt.net/blog/best-api-frameworks-for-fast-performance>
- <https://www.clariontech.com/blog/5-best-technologies-to-build-microservices-architecture>
- <https://blog.back4app.com/top-10-server-side-frameworks/>
- <https://www.codemotion.com/magazine/backend/microservices/develop-microservices-in-php-an-overview-from-enrico-zimuel-talk/>
- <https://www.pangea.ai/resources/web/>

4.1.5 Database

Welke data wordt opgeslagen

In ons spelconcept worden de volgende gegevens opgeslagen:

1. Inhoud

- a. Vragen
- b. Antwoorden
- c. Studierichtingen
- d. Faculteiten
- e. Metronetwerk
- f. Stations

2. Gebruikersdata

- a. Specifieke informatie van de gebruiker voor de start van het spel?

- i. Geboortedatum
 - ii. Naam
 - iii. Mail
 - iv. Studierichting middelbare school
 - v. Voornaamste interesses

- b. Behaalde score

- i. Per station

De inhoud van het spel heeft geen speciale vereisten op gebied van opslag. Het zijn telkens tekstvelden die worden opgeslagen, en het volume is controleerbaar. Het kan opgeslagen worden in zowel SQL als NoSQL formaat.

Gebruikersdata daarentegen kan erg groot worden in volume naargelang meer en meer gespeeld wordt. Naar schaalbaarheid toe kijken we dus best naar een DBMS dat horizontaal schaalbaar is. Hierdoor kunnen we enkel naar een NoSQL formaat kijken voor opslag van deze data.

Vereisten voor het DBMS:

- 1. Schaalbaarheid
- 2. Performantie
- 3. Complexiteit
- 4. Kost
- 5. Security
- 6. Documentatie
- 7. Geschiktheid voor game development

Vergelijking relationele DBMS

De opties die we gaan vergelijken zijn:

- 1. MySQL
- 2. MariaDB
- 3. PostgreSQL

In onderstaande tabel wordt een score gegeven voor elk van de vereisten.

Requirements	Weight (/10)	Scores (%)			Verantwoording	
		Relationale DBMS				
		MySQL	MariaDB	PostgreSQL		
Schaalbaarheid	8	60%	80%	100%	MySQL is eerder opgesteld voor kleine toepassingen. Zorgt voor een aantal uitdagingen op gebied van schaalbaarheid.	
Performantie	5	60%	80%	100%	MariaDb en PostgreSQL zijn lijken erg op MySQL, maar voorzien toevoegingen om de performantie te verhogen. Hierin scoort PostgreSQL het hoogste in.	
Gebruiksvriendelijkheid	5	90%	90%	70%	MySQL en Mariadb zijn beide erg gebruiksvriendelijk. PostgreSQL heeft een iets grotere leercurve. Het is meer ingesteld op complexe datastructuren.	
Kost	5	80%	100%	100%	Alle DBMS in deze lijst zijn open-source en dus gratis in gebruik. MySQL heeft echter betalende uitbreidingen.	
Security	3	70%	100%	100%	MariaDb en PostgreSQL voorzien heel wat toevoegingen op gebied van security tegenover MySQL.	
Community en documentatie	8	70%	100%	70%	MySQL is slechts gedeeltelijk open source. De community is daarom beperkt in het maken van uitbreidingen en verbeteringen	
Total	34	24	30.9	30.1		

Vergelijking non-relationele DBMS

De opties die we gaan vergelijken zijn:

1. MongoDB
2. Firebase

In onderstaande tabel wordt een score gegeven voor elk van de vereisten.

Requirements	Weight (/10)	Scores (%)		Verantwoording	
		Non-relationele DBMS			
		MongoDB	Firebase		
Schaalbaarheid	10	100%	100%	Beide horizontaal schaalbare DBMS.	
Performantie	10	60%	80%	MongoDb vraagt erg veel geheugen op bij het verwerken van queries.	
Gebruiksvriendelijkheid	6	80%	100%	Firebase blinkt uit in gebruiksvriendelijkheid.	
Kost	5	80%	80%	Beide open-source, met een betalende uitbreiding.	
Security	3	50%	70%	Verhoogde security enkel beschikbaar bij betalende uitbreiding voor MongoDB. NoSQL in het algemeen is minder secure dan SQL.	
Documentatie	8	70%	100%	Firebase heeft een erg uitgebreide en kwalitatieve documentatie.	
Geschikt voor game development	10	50%	100%	Firebase biedt een analytics tool aan.	
Total	52	36.9	48.1		

Conclusie:

De data in onze applicatie kan zowel in relationele als non-relationele databases opgeslagen worden. Daarom werd voor de beide een analyse uitgevoerd met als belangrijkste eigenschappen: schaalbaarheid, performantie en geschiktheid voor game development. Hieruit bleek dat voor ons project MariaDb (relationeel) en Firebase (non-relationeel) het meest geschikt zijn.

Uit de analyse van de andere frameworks (back-end en front-end) zal blijken of we de beide systemen in gebruik gaan nemen, of dat we met enkel Firebase verder kunnen.

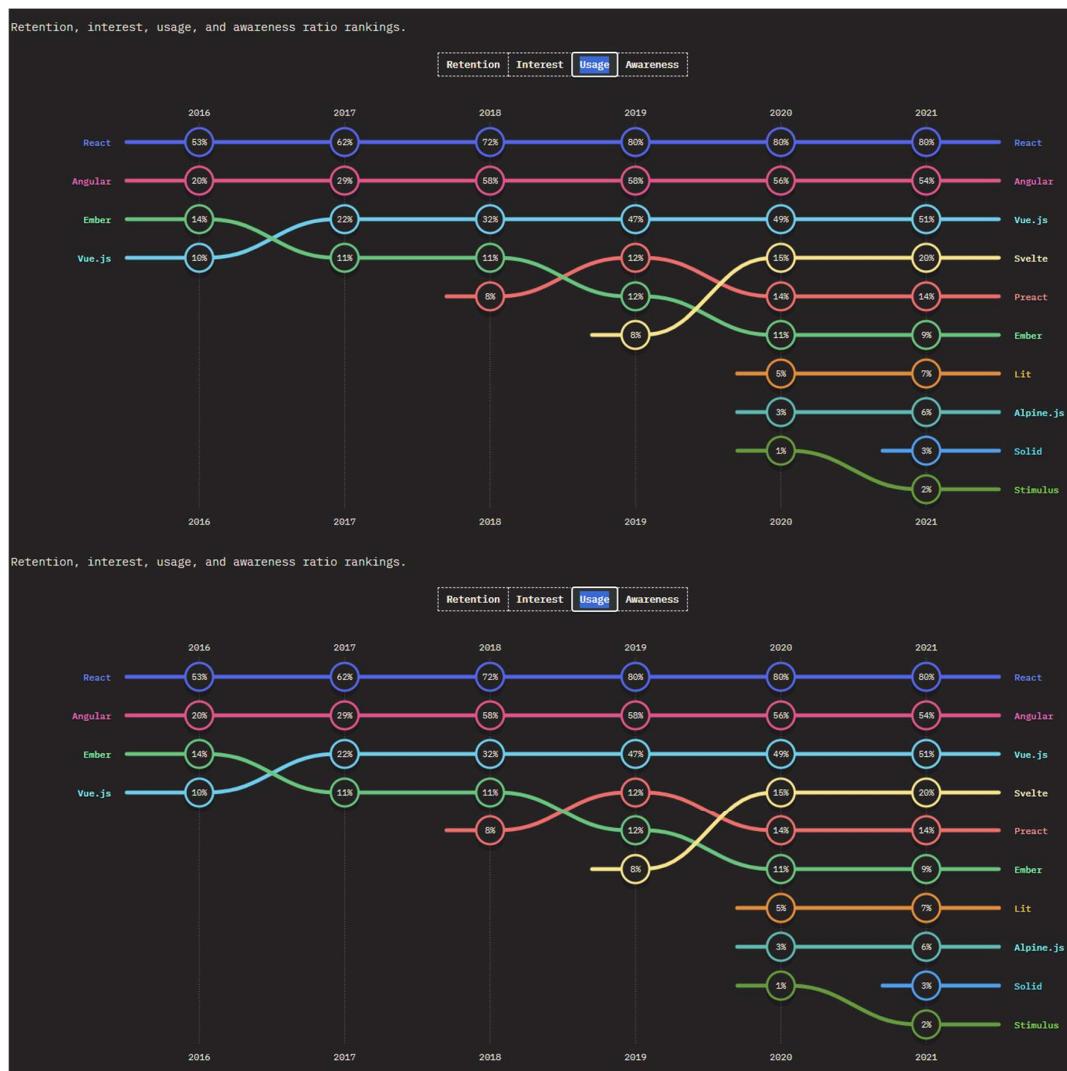
Bronnen:

<https://www.altexsoft.com/blog/business/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/>

4.1.6 Front End frameworks

In het geval we het spel custom zullen builden zal er een keuze gemaakt moeten worden in welk front-end javascript framework of library gebuikt zal worden. Deze keuze maken we op basis van de 'Weighted Ranking Method'. We stellen een aantal criteria op waaraan het framework moet voldoen en geven een score van 1-100. Voor elk criterium bepalen we daarnaast een gewicht, dat aangeeft hoe belangrijk het criterium is voor ons project.

We wegen 3 frameworks/libraries tegen elkaar af. Twee van deze frameworks zijn deel van het curriculum van de bachelor toegepaste informatica aan de Thomas More hogeschool. Deze keuze werd bewust gemaakt omdat het project na onze bijdrage zal verdergezet worden door studenten. Zo hoeft er niet veel tijd verloren te gaan om het framework onder de knie te krijgen. Daarnaast hebben we ook Vue.js mee opgenomen. Dit wordt mee opgenomen omwille van zijn populariteit en hun gebruikerstevredenheid. Zo staat het mee in de top 3 van de meest gebruikte frameworks volgens de 'state of Javascript' survey [bron: https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/#front_end_frameworks_experience_ranking]



De evaluatiecriteria zijn hieronder terug te vinden. Daarnaast is het belangrijk dat het framework er zich toe leent om snel tot een proof of concept te komen dat aan alle functionele en niet-functionele eisen voldoet.

Te onderzoeken frontend Frameworks

- React.js
- Angular
- Vue.js

Requirements	Weight (/10)	Scores (%)			Verantwoording
		React	Angular	Vue.js	
State management	8	80%	85%	70%	Hier scoort Angular beter dan React. In React dient gebruik gemaakt te worden van Redux wat code onnodig uitbreidt. In Angular kan een service gemaakt worden die door Dependency Injection overal kan worden geïntegreerd
Mobile Friendliness	10	85%	80%	80%	Dit is hetzelfde voor de verschillende frameworks.
Performantie	8	80%	80%	90%	Vue is het snelste framework. React het langzaamste.
Uitbreidbaarheid	7	70%	80%	60%	Angular is sneller uitbreidbaar omdat code duidelijk gestructureerd is terwijl dit in React en Vue meer vrij is
Flexibiliteit	9	90%	70%	90%	React en Vue zijn zeer flexibele frameworks. Angular legt een bepaald stramien op
Eigen kennis	9	80%	80%	0%	React en Angular zijn frameworks die we kennen. Vue kennen we niet.
Documentatie	8	65%	80%	70%	Officiële React documentatie is relatief beperkt. Voor angular en vue is deze uitgebreider
Support	8	80%	80%	70%	Omwille van de populariteit van de frameworks is er voldoende support
Geschiktheid	10	20%	20%	20%	Deze frameworks/libraries zijn vooral geschikt voor CRUD operaties. Game ontwikkeling hierin is mogelijk maar is niet waarvoor deze frameworks bestaan.
Total	77	55,1	55,1	46,3	

Conclusie:

De frameworks die worden aangeleerd in de opleiding zijn zeer flexibel en breed inzetbaar. Echter zijn ze niet optimaal voor het ontwikkelen van games. Het is mogelijk om in deze 'kleine' frameworks spelletjes te maken zoals schaken, dammen, en zelfs iets uitgebreidere dingen. Maar het vergt heel wat code en manipulatie van het framework om een vloeiende en sprekende game te maken.

Bronnen:

1. https://2021.stateofjs.com/en-US/libraries/front-end-frameworks/#front_end_frameworks_experience_ranking
2. https://kruschecompany.com/ember-jquery-angular-react-vue-what-to-choose/#Angular_VuejsReact_Preact_EMBER_Svelte_and_SolidJS_which_JavaScript_framework_is_most_popular_in_2022_and_why
3. <https://dev.to/hb/react-vs-vue-vs-angular-vs-svelte-1fdm>
4. <https://javascript.plainenglish.io/angular-vs-vue-vs-react-in-2022-what-do-the-numbers-say-7ae38a549d8d>
5. <https://stateofjs.com/en-us/>
6. <https://serverless.page/blog/stack-choices-react-vs-vue-vs-angular-vs-svelte>
7. <https://www.psd2html.com/blog/vue-vs-react-vs-angular-pros-cons-use-cases-and-comparison.html>
8. <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>
9. <https://athemes.com/guides/angular-vs-react-vs-vue/>
10. <https://krausest.github.io/js-framework-benchmark/current.html>

4.2 Externe interface

Onze game zal communiceren met een zelf ontwikkelde API. Via deze API wordt input opgevraagd rond vragen, antwoorden, studierichtingen, enzovoort. Nadat het spel is voltooid wordt daarnaast ook output gecommuniceerd via deze API om de spelronde te analyseren.

De communicatie gebeurt in een JSON-formaat.

4.3 Performantie

De performantie van de applicatie wordt gemeten op basis van gemiddelde antwoordtijden en schaalbaarheid.

4.3.1 Antwoordtijden

Langdurig wachten tot de antwoorden zijn verwerkt heeft een negatieve invloed op de ervaring van de gebruiker. Het is daarom belangrijk dat er een snelle respons is bij het invullen van de vragen. Dit mag niet belemmerd worden door het aantal users dat gelijktijdig de app aan het gebruiken is.

De verwachte antwoordtijden zijn:

- Normaal: 1s
- Maximum: 2s

4.3.2 Schaalbaarheid

Vaak zal in klassen de demo gegeven worden. In de beginfase is dit klas per klas en hebben we een minimumvereiste van minstens 25 personen die tegelijk het spel kunnen spelen. Bij stijgende populariteit kan dit verder uitgebreid worden.

Vereisten rond de schaalbaarheid:

- Normaal: 25
- Maximum: Onbekend. Horizontale schaalbaarheid voorzien.

4.4 Kwaliteitseisen

Kwaliteitseisen voor ons project bevatten eisen rond beveiliging en gebruiksvriendelijkheid.

4.4.1 Beveiliging

Onze app bevat geen authenticatie, en daarom ook geen gevoelige persoonsgebonden gegevens die sterk beveiligd moeten worden. De data die opgeslagen wordt vermeld in hoofdstuk 5.1.2 zijn eveneens niet kritiek en vereisen geen speciale beveiliging.

De eisen voor beveiliging zijn daarom laag en is geen make-or-break eigenschap.

4.4.2 Gebruiksvriendelijkheid

Gebruiksvriendelijkheid van de applicatie is erg belangrijk. Studenten moeten makkelijk hun weg kunnen vinden in het spel. Het doel van een game is daarnaast dat gebruikers het een aangename ervaring vinden en er een bepaald plezier-aspect aanwezig is. Gebruiksvriendelijkheid speelt hier een belangrijke rol in.

De opdrachtgever gaf ook aan dat het niet noodzakelijk rekening te houden met bvb slechtziende gebruikers.

5 Game engines en frameworks

Volgend op het front end onderzoek zijn we op zoek gegaan naar game engines die ons kunnen helpen het spel te ontwikkelen naar de noden en wensen van de opdrachtgever.

We zijn op zoek gegaan naar platformen/tools die reeds veel van de functionaliteit bevatten die nodig is om ons project uit te werken. Het project is een spelconcept dat studenten op een interactievere manier de studierichtingen laat ontdekken. Dit houdt in dat we een game gaan ontwikkelen. Het zoekgebied voor bestaande platformen betreft zich daarom vooral op game-ontwikkelingsplatformen en software. Eerst lijsten we een aantal basis technische vereisten op waaraan voldaan moet worden.

Technische vereisten – basis

Hieronder zijn de vereisten waaraan het platform moet voldoen opgeliist:

1. Punten bijhouden
2. Tijd bijhouden
3. Interactie met de omgeving
4. Metronetwerk en stations
5. Informatie studierichtingen opslaan
6. Schaalbaar concept
7. Grafische uitwerking
8. Opvolging van statistics van de game
9. Highscores voor iedereen raadpleegbaar
10. Mobile friendly
11. Implementatie in externe hosting

Bij de evaluatie van het platform wordt bij elke vereiste geplaatst in welke mate deze hieraan voldoet.

Technische vereisten analyse op platformniveau

Hieronder worden de onderzochte platformen afzonderlijk afgetoetst aan bovenstaande vereisten.

5.1 Gdevelop.io

'Gdevelop.io' is een 2D development platform waar je op een gebruiksvriendelijke manier games kan ontwikkelen. In onderstaande tabel worden de mogelijkheden van het platform afgewogen tegenover de vereisten die in hoofdstuk 10.1 werden opgesteld.

Requirements	Gdevelop.io
Punten bijhouden	Built in met variables (globaal, scene en object)
Tijd bijhouden	Built in met frames
Interactie met de omgeving	Volledig ingebouwde event functionaliteit (collision, click, timed...)
Metronetwerk en stations	Vlot navigeren tussen scenes. Metronetwerk is upper layer, en kan navigeren naar sublayers (=stations).
Informatie studierichtingen opslaan	Opslaan in de scene variables. Kan communiceren met JSON op remote webserver en deze data vertonen.
Schaalbaar concept	Makkelijk nieuwe stations toevoegen in upper layer metronetwerk
Grafische uitwerking	Grafische elementen kunnen makkelijk geïmporteerd worden en object definitie aangegeven.
Opvolging van statistics van de game	Opslaan en versturen naar remote webserver.
Highscores voor iedereen raadpleegbaar	Raadpleegbaar.
Mobile friendly	Automatisch mobile-friendly. Bevat instellingen speciaal voor mobile, en kan gedeployed worden op Android, iOS...
Implementatie op externe hosting	

5.2 Godot

Ook 'godot' [<https://godotengine.org/>] werd geëxplorerd. Ook hier is het mogelijk om op een gebruiksvriendelijke manier via een visual scripting. Het is ook mogelijk om te programmeren in GDScript wat een eigen taal is. Maar gezien de korte doorlooptijd van het project is het gebruikt van een visual scripting tool makkelijker aan te leren.

Requirements	Godot
Punten bijhouden	Mogelijk om punten bij te houden in een label of in een afzonderlijk object te steken.
Tijd bijhouden	Er is een timer node aanwezig om tijd bij te houden of een aflopende timer te maken
Interactie met de omgeving	Level transitions, animaties op alles zijn mogelijk
Metronetwerk en stations	Scene transitions zijn mogelijk. Je kan makkelijk doorheen de node tree navigeren.
Informatie studierichtingen opslaan	Mogelijk om HTTP requests te doen.
Schaalbaar concept	Makkelijk om scenes en nodes bij te maken en te kopieren en te hergebruiken.
Grafische uitwerking	Het is mogelijk om afbeelding, geluiden, vertalingen, te importeren
Opvolging van statistics van de game	Via HTTP requests op te slaan in de database
Highscores voor iedereen raadpleegbaar	Highscores moeten bijgehouden worden in een database om deze dan te kunnen tonen.
Mobile friendly	Multi platform deployment mogelijk. Desktop, mobile, consoles, web
Implementatie op externe hosting	

Voor de backend is er een gratis godot plugin 'Silentwolf' [<https://silentwolf.com/>] die speciaal ontwikkeld is voor godot die de game en player data kan bijhouden.

5.3 Unity

Unity is een van de bekendste platformen in de game development wereld. Het kan gebruikt worden door kleine en grote projecten, voor 2D als 3D games. Maar ook voor AR games... Zeer uitgebreid platform en ondersteuning.

Mits er zeer uitgebreide informatie aanwezig is door Unity, maar ook op youtube, blogs,... lijkt het erop dat deze software best een stege leercurve heeft.

Requirements	Unity
Punten bijhouden	Is mogelijk
Tijd bijhouden	Is mogelijk
Interactie met de omgeving	Je kan aanklikken, lopen en verkennen van de omgeving.
Metronetwerk en stations	Je kan een wereld bouwen met Unity. Out of the box feature.
Informatie studierichtingen opslaan	Mogelijk om HTTP requests te doen.
Schaalbaar concept	Unity is platform dat gebruikt wordt voor kleine, beginnende developpers als grote games. De klinkende 3D namen gemaakt met dit platform zijn Pokemon Go en Fall guys. Maar er zijn ook verschillende mooie 2D games ontwikkeld. Daarnaast kun je Augmented reality toepassingen ontwikkelen. Wie weet bij verdere ontwikkeling van deze app of als uitbreiding?
Grafische uitwerking	Je kan grafisch uitwerken in 2D en zelfs 3D apps maken. Geluid, animatie kunnen allemaal verwerkt worden.
Opvolging van statistics van de game	Er kan functie geactiveerd worden waarbij statistieken zichtbaar worden. Zo kun je kijken spelersdata bekijken(hoeveel spelers bepaalde levels speelden, hoeveel de toepassing bezocht is), gebruiksdata (aantal keer gepseeld, tijd gespendeerd in spel,..)
Highscores voor iedereen raadpleegbaar	Highscores moeten bijgehouden worden in een database om deze dan te kunnen tonen.
Mobile friendly	Multi platform deployment mogelijk. Desktop, mobile, consoles, web
Implementatie op externe hosting	Is mogelijk. Je kan ook hosten op specifieke game hostings zoals simmer.io en itch.io.

Bronnen:

<https://medium.com/geekculture/how-to-host-unity-games-on-the-web-16be49449b10>

<https://www.juegostudio.com/blog/reasons-why-you-need-pick-unity-game-development-for-creating-a-2d-game>

<https://www.nuclino.com/solutions/game-design-software#unity>

<https://hub.packtpub.com/5-2d-game-engines-you-might-not-have-considered/>

<https://html5gameengine.com/>

<https://learn.unity.com/project/ruby-s-2d-rpg>

5.4 GameMaker Studio 2

GameMaker Studio is een ontwikkelingstool die gebruikt wordt om vooral 2D games (maar gelimiteerd ook voor 3D) te ontwerpen, door zowel ervaren professionele gebruikers, door studenten en beginnende gebruikers. GameMaker is makkelijk in gebruik en kennismaking als startende developer. Je kan je game maken aan de hand van codering in de eigen programmeertaal Game Maker Language of via visual scripting.

Requirements	Unity
Punten bijhouden	Punten kunnen worden bijgehouden
Tijd bijhouden	Timing kan worden bijgehouden, timer kan lopen tijdens het spel
Interactie met de omgeving	Je kan interageren met de omgeving
Metronetwerk en stations	Je kan een kaart maken, een 3D ruimte, ...
Informatie studierichtingen opslaan	Kan bijgehouden worden op server
Schaalbaar concept	Upscaling van het project is mogelijk
Grafische uitwerking	2D en 3D uitwerking is mogelijk, maar de focus ligt op 2D games. (Hiervoor is Unity meer aangewezen)
Opvolging van statistics van de game	Statistieken kunnen worden opgevraagd als dit wordt opgeslagen
Highscores voor iedereen raadpleegbaar	Highscores bijhouden is mogelijk
Mobile friendly	Mogelijk voor mobile, consoles, .. De gratis versie kan enkel gespeeld worden in de browser Opera GX. Wat deze versie niet mogelijk maakt voor deze toepassing
Implementatie op externe hosting	Externe hosting is mogelijk

Bronnen:

https://manual.yoyogames.com/Drag_And_Drop/Drag_And_Drop_Index.htm

<https://help.yoyogames.com/>

<https://gamemaker.io/en/tutorials/>

<https://forum.yoyogames.com/>

<https://www.quora.com/Which-would-be-easier-and-less-frustrating-to-learn-Unity-or-Gamemaker>

5.5 WRM van verschillende platformen:

Requirements	Weight (/10)	Scores (%)				Verantwoording
		Gdevelop	Godot	Unity	Game maker	
Technische eisen, zie ook technische analyse per platform	10	95%	95%	100%	95%	Gdevelop: - Voldoet aan alle vereisten in de technische analyse GODOT: - volgends de technische analyse lijkt alles mogelijk te zijn met godot. Unity: - voldoet aan alle eisen GameMaker Studio 2: - voldoet aan alle eisen geformuleerd in de technische analyse
Prijs / licentiekost	7	90%	100%	30%	20%	Gdevelop: - Bevat licentie formules maar zijn erg laag geprijsd. GODOT:
Git / Teamwerk	10	100%	100%	100%	100%	Gdevelop: - Voorziet mogelijkheden om samen te werken in Git, mercurial en subversion GODOT: - Werkt met Git, Subversion, Mercurial ... Unity: - Unity heeft een Git plugin. https://unity.github.com/ GameMaker Studio 2: - github plugin aanwezig.
Code Skills nodig	7	100%	70%	70%	80%	Gdevelop: - Geen. Gdevelop is een no-code platform dat erg gebruiksvriendelijk is voor beginnende programmeurs. GODOT: - Visual scripting is mogelijk, maar het gebruik van GDscript lijkt onvermijdelijk. Unity: - Visual scripting mogelijk - Codetaal voor scripts is c# en soms c++. Je moet scripten om interactie met objecten aan te gaan, animatie te verwerken. Enige codeskills zullen waarschijnlijk nodig zijn in het totaal plaatje. GameMakers Studio 2 - De codetaal gebruikt in GameMaker is GameMaker Language (GML). Dit kan je vergelijken met Javascript of C. Je kan het ook converteren naar deze talen - GML Visual is de visuele scripting methode voor GameMakers Studio 2 Reviews geven aan dat het iets makkelijk is om games te ontwerpen in GameMaker Studio dan in Unity
Documentatie	9	60%	80%	90%	70%	Gdevelop: - Uitgebreide documentatie over het programma. Enkel is dit minder in vergelijking met langer bestaande frameworks aangezien gdevelop relatief jong is. GODOT: officiële documentatie waaronder een stap voor stap begeleiding naar het maken van een eerste 2D-game. Unity: - volledige manual toegankelijk met stap per stap uitleg. GameMakers Studio 2 - manual vanuit GameMakers beschikbaar - op bvb stack overflow is er weinig te vinden

Requirements	Weight (/10)	Scores (%)				Verantwoording
		Gdevelop	Godot	Unity	Game maker	
Support	9	60%	80%	95%	70%	<p>Gdevelop: - Stackoverflow is een grote bron voor support. De meeste vragen zijn al aangehaald en opgelost. - Verder geen andere platformen voor support. Info op stackoverflow is minimaal.</p> <p>GODOT: - Er lijkt een relatief grote community te zijn. In hun docs verwijzen ze naar verschillende youtube kanalen met tutorials. - Ook zijn er discord, facebook, reddit, en twitter groepen - Een google search naar 'godot tutorial' levert 6,12 miljoen resultaten op - Op stackoverflow zijn er 1300 vragen getagd met 'godot', hiervan zijn er ongeveer 500 van onbeantwoord. Stackoverflow lijkt dus niet meteen een solide bron voor support.</p> <p>Unity: - Uitgebreid support. Zo is er een platform community, blogs en forums waarbij allerlei aspecten aan bod komen. Afhankelijk van je success plans kun je ook technische support ondersteuning hebben via chat. Er is mogelijkheid om zonder licentie een vraag te submitten. - Ook op stackoverflow is er informatie aanwezig. - op youtube verschillende crashcourses te vinden met recente data. - Doordat Unity en Microsoft nauw samenwerken, vergroot dit de kennisdatabase - er zijn ook games stap per stap en onderdeel uitgelegd in korte tutorials</p> <p>GameMaker Studio 2 - manual beschikbaar. Zowel over het programma als de coding language GML. Er zijn ook verschillende tutorials op youtube beschikbaar. - eigen forum - input op bvb stackoverflow is minimaal</p>
Leercurve	6	100%	70%	35%	80%	Gdevelop: - De leercurve voor dit platform is erg laag aangezien het no-code is. Het kan
Uitwerking grafische elementen	9	40%	40%	100%	90%	Gdevelop: - De grafische elementen zijn eerder simpelstisch. Het is niet bedoeld om de gebruiker visueel te verbluffen met realistische scenes.
Total	67	53,2	53,6	54,75	52	

Zie ook extern excel bestand. WRM_GameEngines.xlsx

Conclusie

Om tot een keuze te komen van welke game engine te gebruiken wordt er een WRM 'Weighted Ranked Method' opgemaakt om de verschillende platformen en engines met elkaar te vergelijken. Naast de technische mogelijkheden dit in hoofdstuk 10.2 besproken werden worden ook volgende factoren mee in rekening genomen:

- Prijs / licentiekost

Sommige engines werken met een licentiekost afhankelijk van de functionaliteiten die gewenst zijn of de schaal van het project.
- Git integratie / Teamwerk

Het is een project waar met verschillende teamleden tegelijk aan gewerkt zal worden. Een manier om aan teamwerk te kunnen doen aan het project is dus noodzakelijk
- Documentatie

Is er een officiële documentatie beschikbaar van de game-engine? OF moet alles worden uitgezocht via trial en error?
- Support

Is er veel informatie beschikbaar rond gebruikersvragen? Vind je veel vragen terug op forums zoals stackoverflow?
Deze input, en rond de documentatie is noodzakelijk (en hebben we dus een hoge score gegeven) omdat we hier nog geen voorkennis in hebben, en we het leertraject zelfstandig zullen doorlopen. Hierbij komt ook volgend item:
- Leercurve

Hoe makkelijk is het om een game in dit programma te maken voor de eerste keer? Is dit makkelijk (bvb GameMaker), of wordt dit toch eerder als pittig omschreven (bvb Unity)
- Code skills

Welke ervaring is er nodig om deze game te coderen? We merken dat voor het ontwerpen van minder complexe games, heel wat engines ook visual scripting aanbieden. Hierbij moet je geen

letterlijke code schrijven, maar wordt de code voor jou geschreven aan de hand van acties die jij instelt.

- **Grafisch element**

Als laatste was er het grafische element. De opdrachtgever wil een sterk grafisch verhaal, waardoor we dit hebben meegenomen in de WRM.

Voor een uitgebreidere verslaggeving per item verwijzen we naar het extern excel bestand:

WRM_GameEngines.xlsx

Als er gekeken wordt naar de elementen en hun gewichten komt Unity eruit met de hoogste score. Deze komt er wel het zwakst uit bij de leercurve. We zien dat bvb GameMaker veel makkelijker is voor beginnende gebruikers. Dit betekent dat we de eerste weken ons hard zullen moeten inwerken. Maar er wordt meer belang gehecht aan support & documentatie (ook voor dit project op lange termijn), Godot scoort hier ook sterk, Gdevelop minder omdat het een vrij recent platform is. Er is een gratis licentie (ook een aparte voor studenten), dit is bvb zeer beperkt bij GameMaker. Grafisch scoort Unity sterker als de andere engines, wat toch een belangrijk punt is in deze opdracht.

6 Skills studenten Thomas More ifv project coordinatie

We hebben de studierichtingen binnen Thomas More, relevant voor deze opdracht in kaart gebracht. Per richting zochten we welke vakken er relevante skills/kennis aanbrengen.

We merken (over de campussen heen) dat volgende opleidingen van meerwaarde kunnen zijn in het verder verloop van deze tool:

- professionele bachelor Bedrijfsmanagement
- professionele bachelor Organisatie en Management
- professionele bachelor Bedrijfsmanagement
- professionele bachelor Communicatie
- professionele bachelor Informatiemanagement en multimedia
- professionele bachelor Elektronica-ICT
- professionele bachelor Toegepaste Informatica
- graduatsopleiding programmeren
- graduat Systeem- en netwerkbeheer in dag- of avondonderwijs

Voor meer details (studierichtingen en vakken) : zie hiervoor bijlage excel: 02_3_Onderzoek skills binnen TM.xlsx

7 Analyse & onderzoek – conclusie

**01 | ANALYSE &
ONDERZOEK**
WAY FORWARD

WAY FORWARD

Gebaseerd op resultaten vanuit ons onderzoek afgestemd op het einddoel, kiezen we voor volgende ontwikkel omgeving:

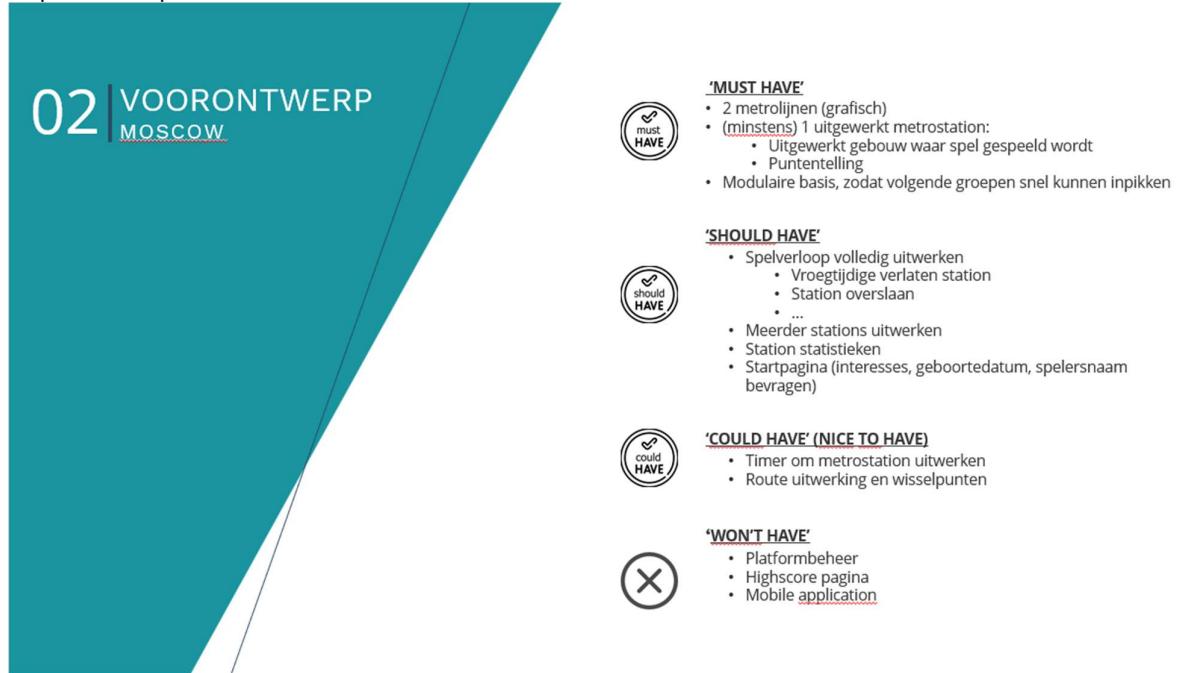
- Game-engine: **Unity**
 - sterke visuals
 - beste documentatie
 - Hoge overdraagbaarheid van geleverd werk
 - gratis studenten-licentie
- Hosting & database: **Firebase**
 - gratis package,
 - integratie met Unity, snelle reactietijd
- Back End: **Firebase**
 - **Integreert** vlot met de gekozen game-engine. **Geen extra framework** nodig om de communicatie met database te voorzien.

VOORONTWERP

8 Technisch plan van aanpak: realisatie

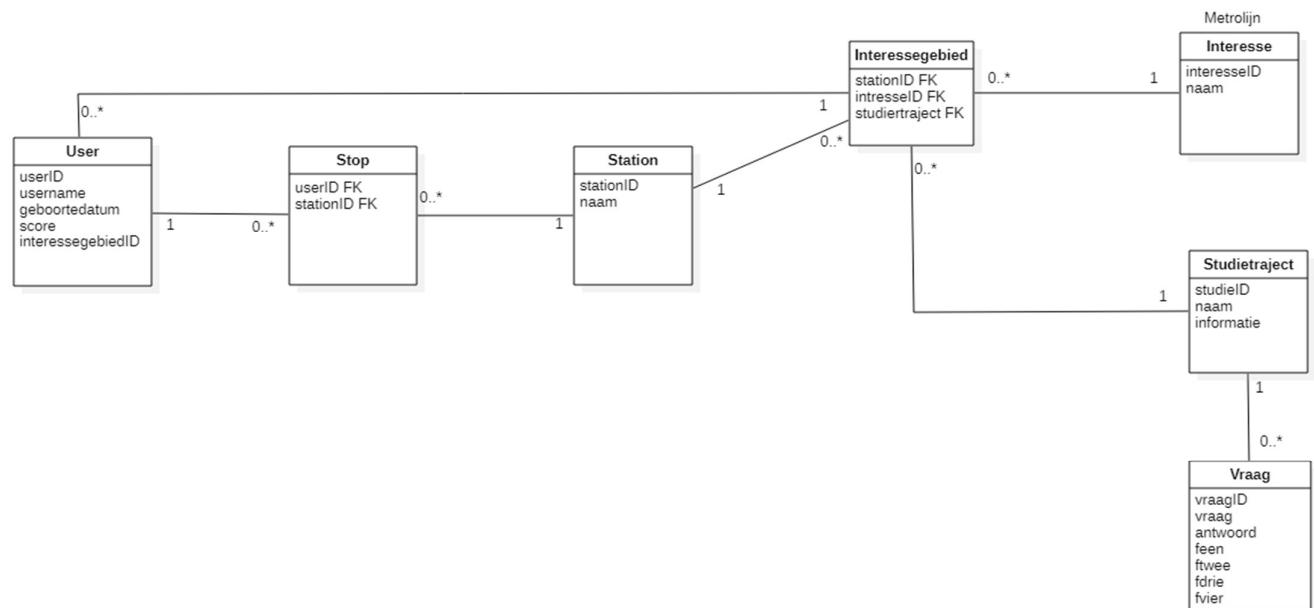
8.1 MoSCoW

Bepalen van prioriteiten en vereisten van WEBOOS

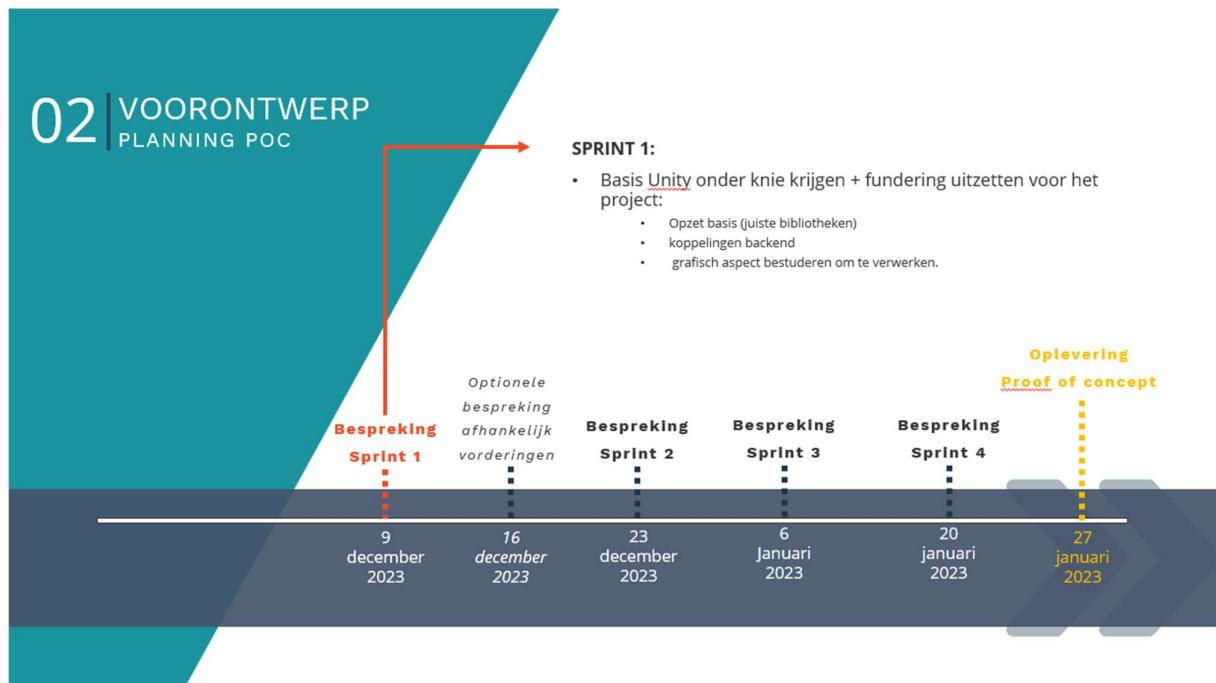
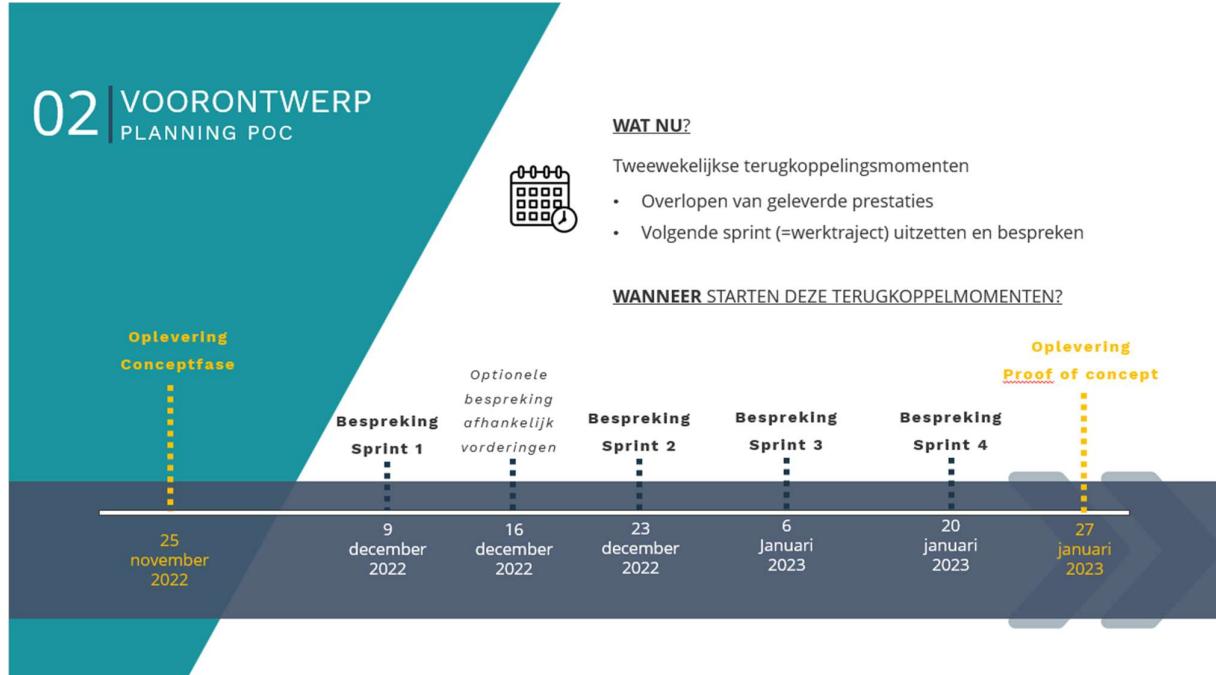


8.2 Datamodel

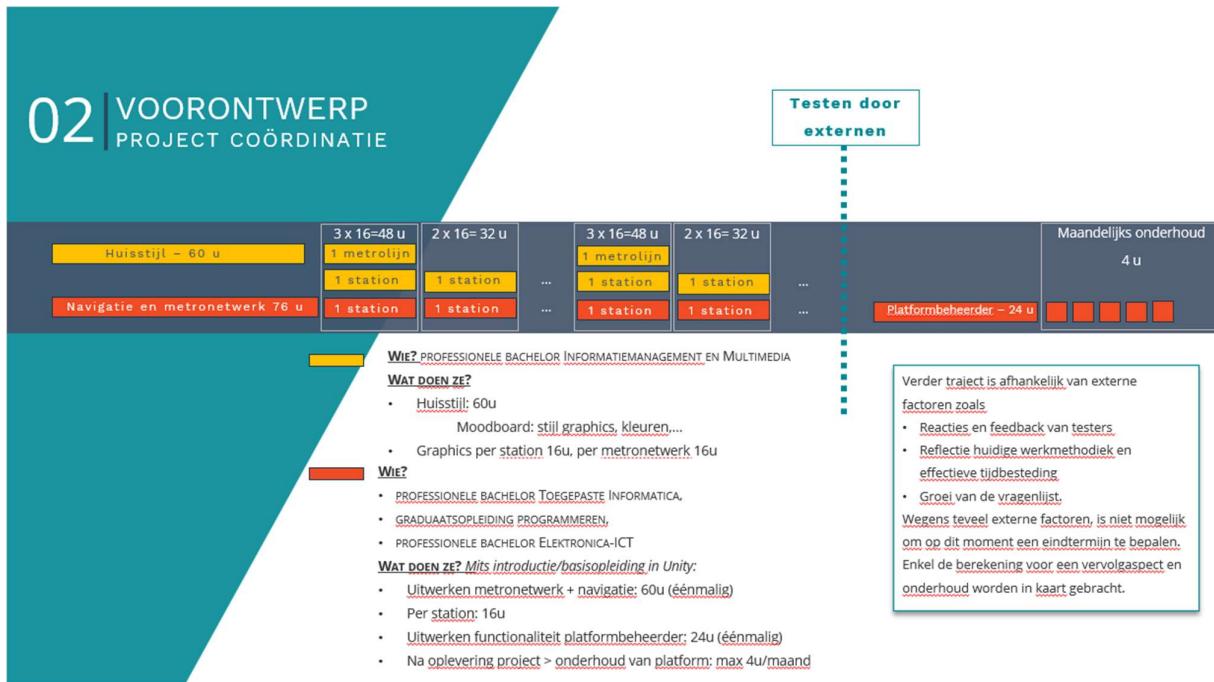
Op basis van de gevraagde gegevens en spelconcept werd een datamodel uitgewerkt:



8.3 Planning

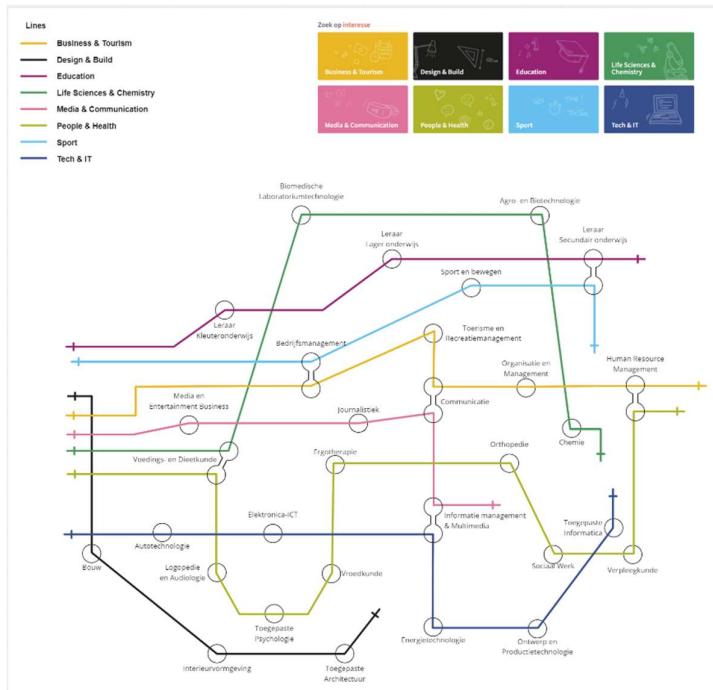


02| VOORONTWERP PROJECT COÖRDINATIE



9 Vertaling Thomas More opleidingen naar Metrolijnen

Om een zicht te krijgen voor het grafische aspect van het prototype . En hoe dit vertaald kan worden naar een Unity game, is er een initiële mapping uitgezet. In deze mapping zijn de opleidingen binnen Thomas More uitgezet aan hand van de metrostations. De metrolijnen corresponderen met de interessegebieden. En ook is er nagedacht over mogelijke koppelpunten.



10 Prototype:

10.1 Prototype visualisatie

Er werd getracht een beeld te vormen van hoe de tool er zou kunnen uitzien en zou kunnen werken. De uiteindelijke beeldvorming kan afwijken, maar onderstaande beelden geven een idee van hoe er in de game/tool kan genavigeerd worden...

Startscherm:



Highscores:

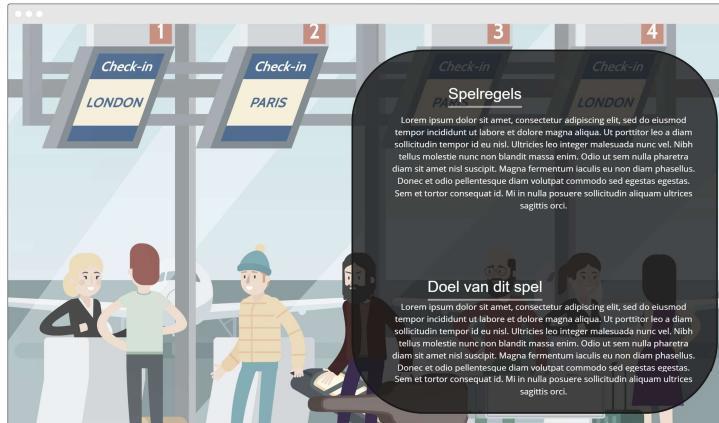


About this:

02| PROOF OF CONCEPT PROTOTYPE



The image shows the exterior of a modern train station building with large glass windows and a prominent sign reading "CENTRAL STATION". A small digital screen in the foreground displays a game menu with options: "Start Game", "Highscores", and "About this".



A perspective view of a station platform. In the background, there are four check-in kiosks labeled 1, 2, 3, and 4. Each kiosk has a sign above it that says "Check-in" and the name of a city: LONDON, PARIS, LONDON, and PARIS respectively. In front of the kiosks, several cartoon-style characters are standing, some holding luggage. To the right, a large black rounded rectangle contains text and a sub-section titled "Doel van dit spel".

Spelregels

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut pellentesque id eu nisl. Ultricies leo integer malesuada nunc vel. Nibh tellus molestie nunc non blandit massa enim. Odio ut sem nulla pharetra diam sit amet nisl suscipit. Magna fermentum acicilis eu non diam phasellus. Donec et odio pellentesque diam volutpat commodo sed egestas egestas. Sem et tortor consequat id. Mi in nulla posuere sollicitudin aliquam ultrices sagittis orci.

Doel van dit spel

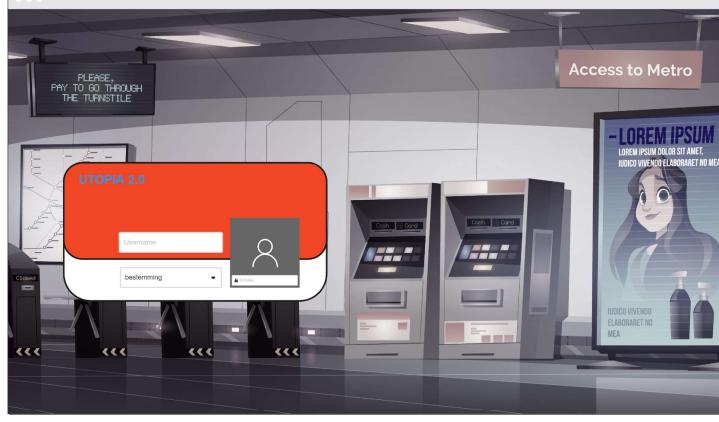
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut pellentesque id eu nisl. Ultricies leo integer malesuada nunc vel. Nibh tellus molestie nunc non blandit massa enim. Odio ut sem nulla pharetra diam sit amet nisl suscipit. Magna fermentum acicilis eu non diam phasellus. Donec et odio pellentesque diam volutpat commodo sed egestas egestas. Sem et tortor consequat id. Mi in nulla posuere sollicitudin aliquam ultrices sagittis orci.

Spel spelen:

02| PROOF OF CONCEPT PROTOTYPE



The image shows the exterior of a modern train station building with large glass windows and a prominent sign reading "CENTRAL STATION". A small digital screen in the foreground displays a game menu with options: "Start Game", "Highscores", and "About this".



A perspective view of a subway station platform. In the center, there is a large, illuminated red and white kiosk with the text "UTOPIA 2.0" on top. The kiosk has a screen displaying a user interface with fields for "bestemming" (destination) and a user icon. Above the kiosk, a sign reads "PLEASE PAY TO GO THROUGH THE TURNSTILE". To the right, there are two grey ticket vending machines. On the far right, a large digital billboard displays a woman's face with the text "- LOREM IPSUM" and "LOREM IPSUM DOLOR SIT AMET, DUCIB VIVENDI ELABORET IN MEA". Above the billboard, a sign says "Access to Metro".

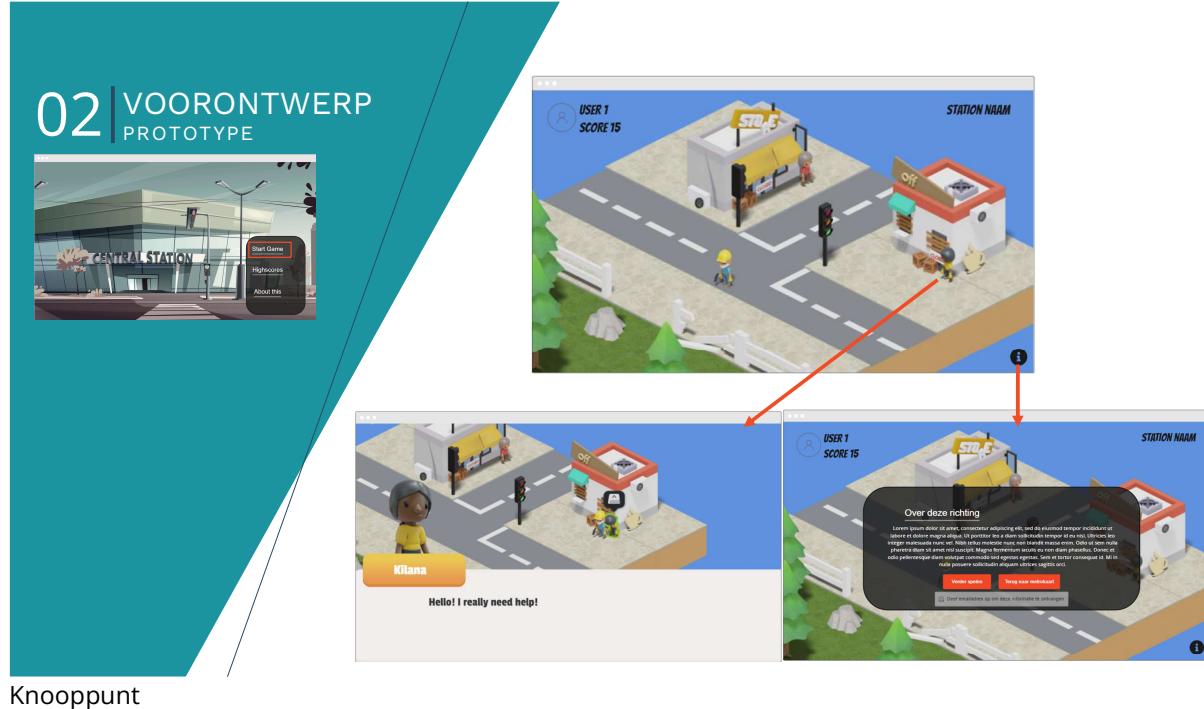
Metrokaart:



Je krijgt optie om te spelen:



Spelen gekozen (rondwandelen om vragen te beantwoorden + info knop onderaan)



Knooppunt



Bij het afsluiten van de game, krijgt men de vraag om feedback achter te laten:



Na dit te verzenden, wordt men teruggeleid naar het startscherm.
Spel uitgespeeld:



Na dit te verzenden of keuze opnieuw spelen, wordt men teruggeleid naar het startscherm.