

Subfaculteit wetenschappen



Probleemoplossen en ontwerpen 3

# Communicatieplatform

**Let the dogs out**

**Thomas Varheust  
Ruben Min Jou  
Manon Delobelle**

Titularis : Koen Van Den Abeele

Begeleider : Charlotte Deconinck

Academiejaar 2021 - 2022

# Inhoudsopgave

<b>1</b>	<b>Studie voor tools</b>	<b>4</b>
1.1	Taal . . . . .	4
1.2	App . . . . .	5
1.3	Website . . . . .	5
1.4	Database . . . . .	6
<b>2</b>	<b>Resultaten</b>	<b>7</b>
<b>3</b>	<b>Samenwerking</b>	<b>11</b>
3.1	Verantwoordelijkheden . . . . .	11
3.2	Taakverdeling . . . . .	11
	<b>Appendices</b>	<b>13</b>

## Inleiding

Blinde mensen hebben vaak honden nodig om hen te helpen in het dagelijks leven. Die blindengeleidehonden hebben daarvoor eerst een opleiding nodig. Ze moeten zowel stressbestendig als gehoorzaam zijn in drukke situaties om zo hun baasje optimaal te begeleiden. Hiervoor verblijven ze gedurende een periode bij een pleeggezin en worden ze getraind door hun gezin en trainers. In die periode is goede communicatie erg belangrijk. Zo kunnen de blindengeleidehonden optimaal worden opgeleid om later een blinde te begeleiden.

Op dit moment verloopt die communicatie niet altijd vlot met opleidingscentra. Dit heeft ongunstige gevolgen voor de opleiding van de hond. Daarvoor zoeken wij een oplossing.

Het doel is om een communicatieplatform te creëren voor die pleeggezinnen en hun trainers. Op die manier kan het gezin communiceren met de trainer en omgekeerd. Hiervoor heeft iedereen een persoonlijke login waarmee ze gegevens kunnen vinden over de opleiding van hun hond. Ze kunnen een kalender raadplegen waarop ze de trainingen en toediening van medicatie kunnen bekijken. Daarnaast is het ook mogelijk om een dagboek bij te houden van de verschillende activiteiten die het gezin kan doen met de hond, zoals bijvoorbeeld een terrasje doen. Die data kan worden opgevraagd via een overzicht om inzicht te krijgen in welke activiteiten er meer getraind moeten worden. Het gezin kan ook verslagen van trainingen opvragen van de trainers.

Hiervoor is zowel een Android-app als een website nodig: de gezinnen kunnen via hun app herinneringen krijgen voor trainingen van de hond, of voor medicijnen dat de hond moet innemen . . . Daarnaast kan de trainer via zijn computer de verslagen van trainingen uploaden die belangrijk zijn om te weten waarop er nog getraind moet worden.

## Integratie van vakken

Het hoofddoel is een app en een website te bouwen die samen een communicatieplatform vormen. Het vak 'Beginselen van programmeren' hebben we hierbij nodig aangezien programmeren een hoofdzaak is bij het bouwen van zo'n app en website. In die opleiding hebben we de taal **Python** geleerd en inzichten gecreëerd die handig kunnen zijn in dit project.

# 1 Studie voor tools

## 1.1 Taal

Er bestaan enorm veel programmeertalen. We kenden enkel **Python**, dus de andere talen zouden we moeten aanleren. Aangezien **Python** niet de beste taal bleek te zijn om een app te ontwikkelen, zochten we verder. Doordat we die talen nog moeten leren kennen, hebben we beslist om bekendere talen te gebruiken. Zodat we het op die manier makkelijker konden aanleren aan de hand van vele online video's die beschikbaar staan op het web. Wij hebben op basis van onze voorkennis en een onderzoek meteen een selectie gemaakt tot 6 talen: **Java**, **JavaScript**, **Python**, **Kotlin**, **C** en **C++**. Daarna zochten we uit welke talen het beste waren voor onze opdracht.

**Java** en **Kotlin** blijken de beste programmeertalen te zijn om een app te bouwen. Zowel **Java** als **Kotlin** zijn officiële programmeertalen voor Android development, daardoor sluiten ze goed aan op het besturingssysteem Android. **Kotlin** is makkelijk aan te leren, maar bestaat pas sinds 2011 waardoor er niet zoveel documentatie online staat. Online documentatie is voor sommige onderdelen (zoals bij chat) erg belangrijk doordat het erg moeilijk is om daarvoor van nul te starten met een taal die voor ons niet volledig gekend is. Om een webapp<sup>1</sup> te ontwikkelen wordt er aangeraden om dit met talen te doen die populair zijn in de webdevelopment wereld, zoals **Java**. We hebben dus gekozen om **Java** te gebruiken als programmeertaal voor de app.

Voor de *front-end*<sup>2</sup> ontwikkeling van de website kozen we ervoor om gebruik te maken van **HTML**, **CSS** en **JavaScript**. Dit zijn veruit de meest gebruikte talen voor de ontwikkeling van een *front-end* (Meer dan 94% van de websites maakt gebruik van **HTML**, **CSS** en **JavaScript**). Ook zijn **HTML** en **CSS** de eenvoudige talen om aan te leren en heel overzichtelijk. **HTML** wordt gebruikt voor de structuur. Voor de presentatie wordt **CSS** gebruikt en voor het gedrag van de site **JavaScript**.

---

<sup>1</sup>Een webapp is een app waarvoor het mogelijk is om die te benaderen met een browser.

<sup>2</sup>De *front-end* van een website is deel van het programma dat zichtbaar is voor de gebruiker. Dit zorgt voor het visuele gedeelte van de site.

## 1.2 App

Er bestaan veel tools om een app te ontwikkelen voor het besturingssysteem Android. De tool moest voldoen aan verschillende vereisten. Allereerst moet het compatibel zijn met Android. Het moet ook mogelijk zijn om de data weg te schrijven naar een database zodat er een website aan gekoppeld kan worden. Daarnaast hadden we geen budget waardoor we een gratis platform zochten. Tot slot moet de tool ook aan alle vereisten voor ons probleem voldoen, zo moet het mogelijk zijn om documenten up te loaden, om een dagboek bij te houden en om een chatfunctie te installeren.

Aangezien wij een Android-app moeten ontwikkelen was de keuze snel gemaakt; we kozen voor Android Studio. Het brengt app-ontwikkeling eenvoudiger dankzij de *lay-outeditor*, *APK-analyzer* en *Vector Asset Studio*... Die eerste zorgt ervoor dat een lay-out snel gebouwd kan worden en er later makkelijk onderdelen kunnen gewijzigd worden. De *Analyzer* zorgt dat de *APK*<sup>3</sup>-grootte verkleint. Dankzij de *Vector Asset Studio* kunnen verschillende bestanden gegenereerd worden voor alle schermformaten. Android Studio is dus specifiek gericht op het Android platform. In Android Studio is het ook makkelijk om programmeertalen *Kotlin*, *Java* en *C++* te combineren bij het programmeren van de app. Het is ook mogelijk om aan de hand van Android-emulators de app te testen. Daarnaast kan Android Studio ook gebruikt worden met hulpprogramma's voor een beter en efficiënter groepswork. Het kan gesynchroniseerd worden met *GitHub*<sup>4</sup> waardoor alle groepsleden kunnen werken met de meest recente versie van de app. De verschillende functionaliteiten kunnen via deelprogramma's onafhankelijk worden ontwikkeld en getest. Tot slot kan Android Studio makkelijk gekoppeld worden met de database *Firebase* (Dit komt later nog aan bod in hoofdstuk 1.4 op pagina 6.) doordat *Firebase*-services rechtstreeks kunnen geïntegreerd worden vanuit Android Studio.

## 1.3 Website

Voor de ontwikkeling van de *back-end*<sup>5</sup> van de website bestaan er veel verschillende tools. We zochten een gratis tool die op verschillende *browsers* werkt. Hiervoor leken *Python Flask*, *PHP* en *Node.js* ons de beste opties. We hebben ervaring met *Python* door het vak 'Beginselen van programmeren' en moeten *JavaScript* beheersen voor de *front-end* van de website. Daarnaast hebben we ook maar 12 weken voor ons project, dus besloten we om *PHP* niet te gebruiken zodat we niet nog een nieuwe syntax moeten leren kennen. De *event-driven* architectuur van *Node.js* maakt het eenvoudiger om een chat-app te implementeren en is snel en makkelijk te begrijpen, waardoor onze keuze gemaakt was. *Node.js* is een open source en multiplatform van *JavaScript*.

---

<sup>3</sup>APK is een bestandsformaat voor software van het Android-platform.

<sup>4</sup>*GitHub* is een versiebeheersysteem. Telkens er een wijziging wordt gedaan in de code, wordt deze opgeslaan. Deze kunnen later gepusht worden zodat elk teamlid de wijzigingen ziet.

<sup>5</sup>De *back-end* van een website is deel van het programma dat onzichtbaar is voor de gebruiker. Dit zorgt ervoor dat de *front-end* van de website werkt.

## 1.4 Database

Zowel de trainers als de pleeggezinnen zullen bij het gebruik van de app informatie, zoals documenten, dagboekinzendingen, afspraken ... aan de app toevoegen. Al deze informatie moet ergens opgeslagen worden en vanuit die opslagplaats weer geraadpleegd kunnen worden. Daarom maken we gebruik van een database. We hebben meer bepaald een realtime database nodig, zodat onder andere chatberichten zo snel mogelijk gesynchroniseerd moeten worden.

Onze eerste keuze was om gebruik te maken van de gratis en zeer populaire database *MongoDB*. Dit bleek echter geen realtime database te zijn en was dus toch niet geschikt voor onze toepassing.

Daarom zijn we overgeschakeld naar de databases van *Firebase*. *Firebase* is een platform dat verschillende producten huisvest om het creëren van apps en websites te vergemakkelijken. De Realtime database en de Cloud Firestore, zijn de twee databases uit deze producten. Het voordeel van deze databases is data ze beiden real-time zijn. *Firebase* is bovendien een populair platform, waardoor er ook een uitgebreide documentatie en veel voorbeelden beschikbaar zijn. Daarnaast heeft het platform ook enkele andere producten die ons zullen helpen, meer bepaald een beveiligd authenticatiesysteem en de mogelijkheid tot opslag van foto's en video's in de cloud.

Een nadeel van *Firebase* is dan weer dat het platform maar beperkt gratis is. Voor elk product moet er vanaf een bepaalde limiet worden betaald om het verder te kunnen gebruiken. Het is echter zeer onrealistisch dat we deze limieten zullen overschrijden bij het ontwikkelen en testen van de app en de website. Zelfs bij het uitgeven van de app en het openbaar stellen van de website zullen de kosten, wegens het relatief kleine gebruikersaantal, laag blijven.

na de keuze voor *Firebase* moet de keuze tussen de twee databases nog gemaakt worden. Beide databases hebben ook verschillende specialiteiten en beperkingen, maar deze zijn meestal vrij specifiek en voor ons niet van toepassing. Over het algemeen wordt *Cloud Firestore* wel beschouwd als een verbetering van de oudere Realtime database, maar de Realtime database heeft door zijn leeftijd dan weer meer documentatie beschikbaar. We beslissen toch om gebruik te maken van *Cloud Firestore*, vooral omdat het filteren van data hierin gemakkelijker is en dit iets is dat we vaak moeten zullen doen.

Het switchen tussen deze databases zal niet veel werk vragen, dus indien de Realtime database toch beter blijkt te zijn, kunnen we er toch nog gebruik van maken. Daarnaast kunnen we, indien nodig, de twee databases ook tegelijk gebruiken.

## 2 Resultaten

Op dit moment zijn we eraan uit welke tools we gaan gebruiken: Android Studio voor de app, HTML, CSS en JavaScript in combinatie met Node.js voor de website en *Cloud Firestore* als database. Daarna hebben we gezocht naar een structuur (die weergegeven wordt op pagina 7), die we wilden bereiken voor het communicatieplatform. Hiervoor hebben we verschillende pagina's nodig: login, menu, chatoverzicht, documenten, kalender, dagboek, activiteiten toevoegen, overzicht van activiteiten, documenten toevoegen en chatberichten. Het pleeggezin kan chatten, documenten lezen, kalender bekijken en een afspraak toevoegen, het dagboekje bekijken en een activiteit toevoegen. Voor het dagboekje zijn meerdere opties, ofwel wordt er een activiteit toegevoegd, ofwel wordt er een overzicht gevraagd van een bepaalde activiteit. Daarnaast moet de trainer ook bepaalde functies kunnen uitvoeren. Hij kan ook chatten en het dagboek bekijken voor een specifieke hond. Daarnaast kan hij documenten toevoegen en afspraken toevoegen in de kalender, hiervoor kiest hij of hij dat voor een specifieke puppy wil doen of voor alle puppy's tegelijk.

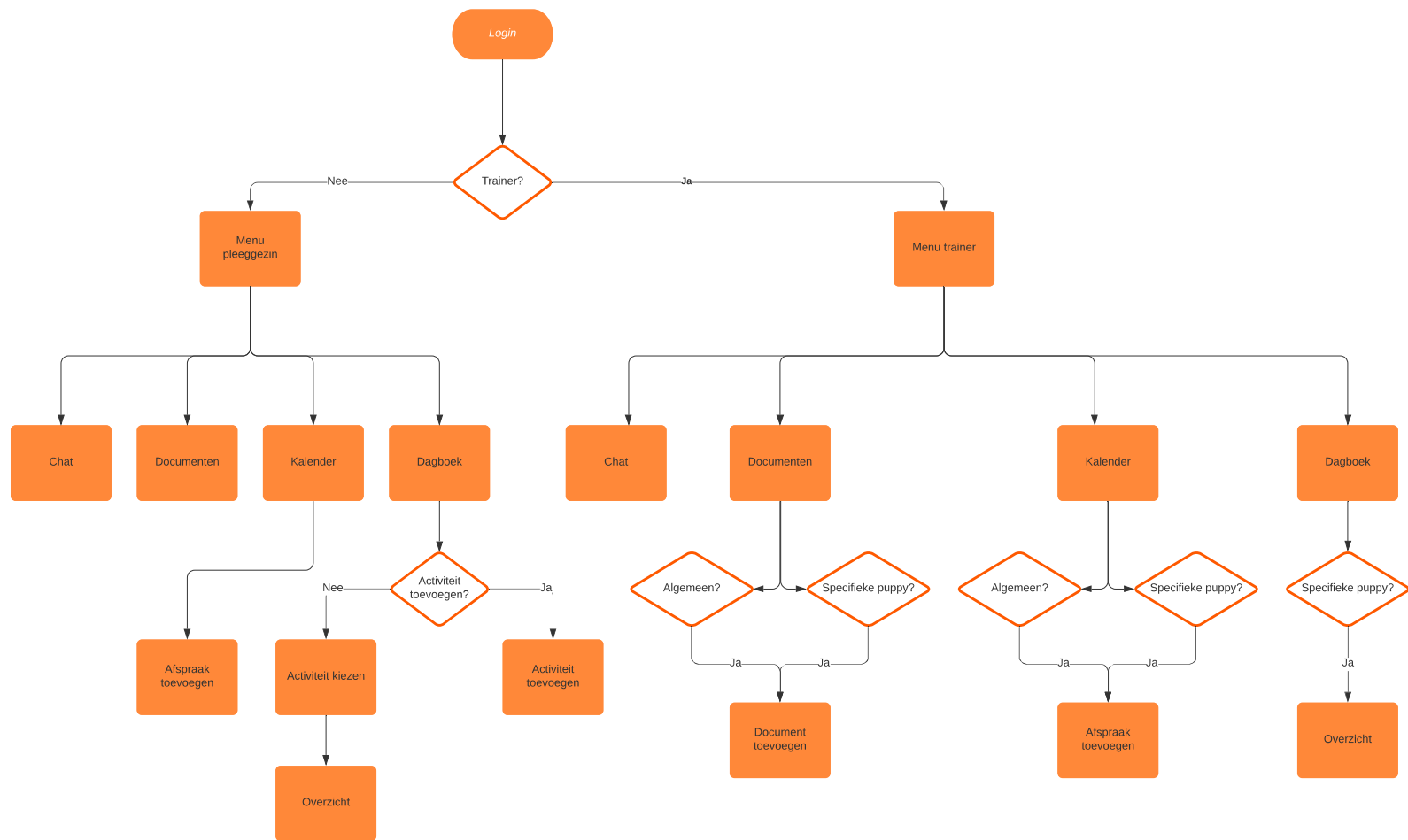
Voor de app is er al een uitgewerkt idee voor het design. Hierbij zijn de verschillende pagina's al ontworpen en is het al mogelijk om door te klikken tussen de pagina's. Ook is er al begonnen aan het programmeren van de kalender en het uploaden van documenten. Daarnaast is het gelukt om de app te verbinden met *Cloud Firestore*.

Met diezelfde database is het ook gelukt om de website te verbinden. Voor de website lukken het inlog-systeem en het chatten al en wordt er gewerkt aan het dagboekje.

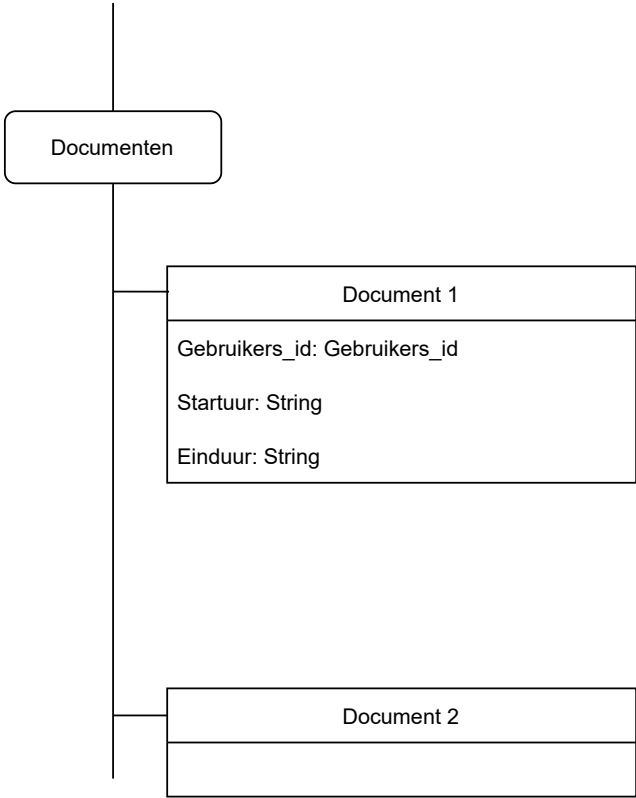
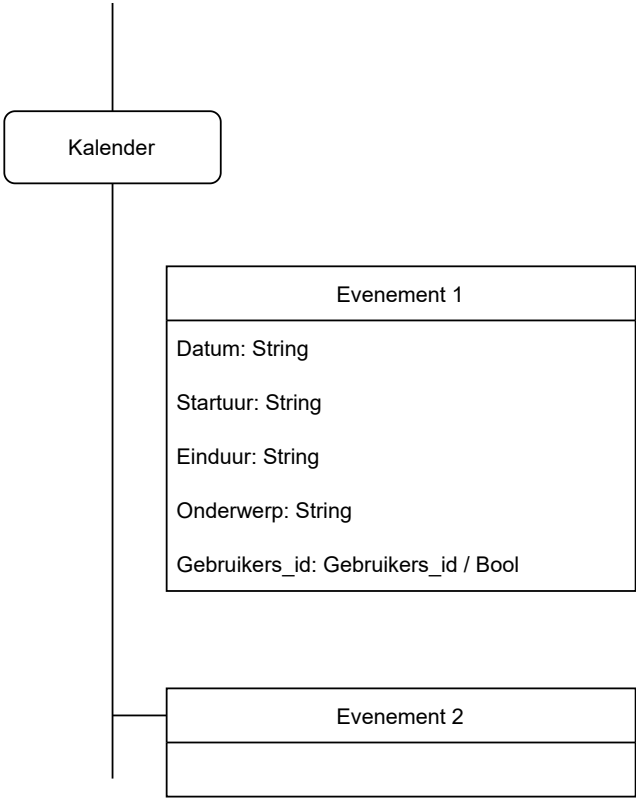
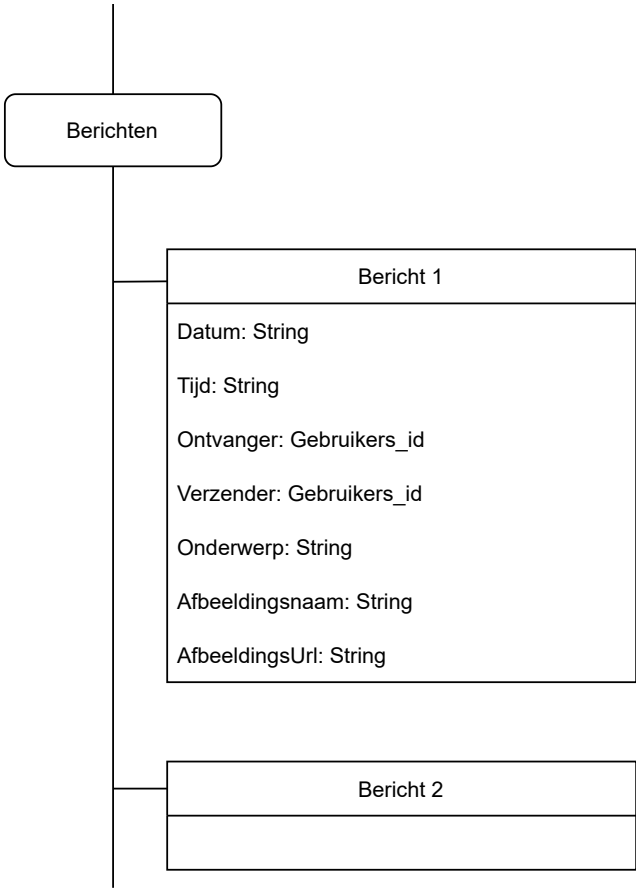
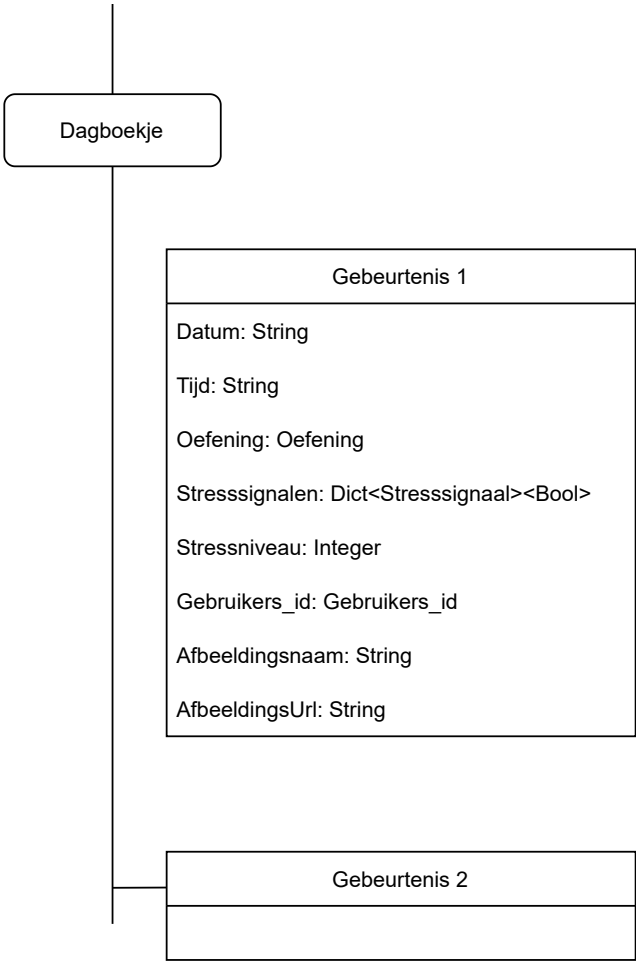
*Cloud Firestore* is dus gekoppeld met zowel de app als de website. Bij zo'n database is de structuur vooral belangrijk. Dit heeft een overzicht (zie pagina 9) voor hoe we alles gaan wegschrijven vanuit de app en website naar de database. Hierbij maken we verschillende collecties aan: Dagboekje, Berichten, Kalender, Documenten en Gebruikers. Elke collectie bevat verschillende items met elk hun eigen parameters/gegevens.

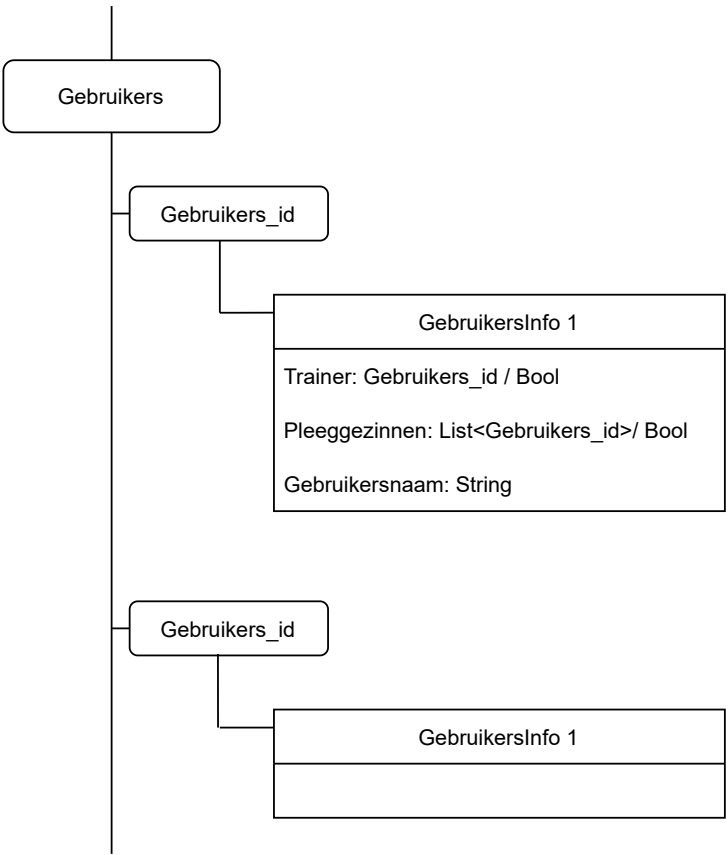
Er moeten nog veel pagina's geprogrammeerd worden bij de app. Ook bij de website is er nog veel werk aangezien er nog moet gekeken worden voor bepaalde pagina's te implementeren en voor het design van die pagina's. Daarnaast moet alles nog getest worden, of alles juist doorgegeven wordt via de database van de app naar de website en omgekeerd. Tot slot moet er nog gekeken worden voor het onderscheid tussen de functies te programmeren van trainers en pleeggezin-nen.

# Flowchart communicatieplatform









## 3 Samenwerking

### 3.1 Verantwoordelijkheden

Bij het begin van het project werden er rollen en dus verantwoordelijkheden gecreëerd. Thomas heeft de meeste voorkennis bij het ontwerpen van een website, dus werd de verantwoordelijke voor de site. Daarnaast heeft Ruben de verantwoordelijkheid van de app op zich genomen. En tot slot is Manon verantwoordelijk voor het design van het volledige platform en voor de rapportering van ons project.

### 3.2 Taakverdeling

Wij hebben samen gezocht naar verschillende tools om zo'n website en app te bouwen en met elkaar te verbinden via een database. Toen we een idee hadden welke tools we voor het project gingen gebruiken, werd de Gantt-grafiek (in bijlage op pagina 13) opgesteld en hebben we de taken (in bijlage op pagina 17) opgesplitst.

Thomas is toen begonnen met het connecteren van de database en de website. En hij heeft erna de structuur van de database gemaakt en is begonnen met de pagina's van de website te implementeren. Ruben heeft gekeken voor het connecteren van de database en de app. En hij heeft ervoor gezocht dat Android Studio via Github functioneert om zo een vlotter groepswork te creëren. Daarna is hij begonnen met het implementeren van enkele pagina's. Manon is begonnen met de app te ontwerpen: het design van de pagina's en de doorverwijzingen tussen de pagina's. Daarna is ze begonnen met het tussentijds verslag te schrijven, er structuur in te brengen en bijlagen te ontwerpen.

Voor het maken van het tussentijds verslag hebben we later ook de taken verdeeld. We namen elk een of meerdere onderdelen voor zich. Iedereen kreeg het onderdeel waarvan hij of zij het meest vanaf wist. Tot slot heeft iedereen het verslag doorlezen en goedgekeurd.

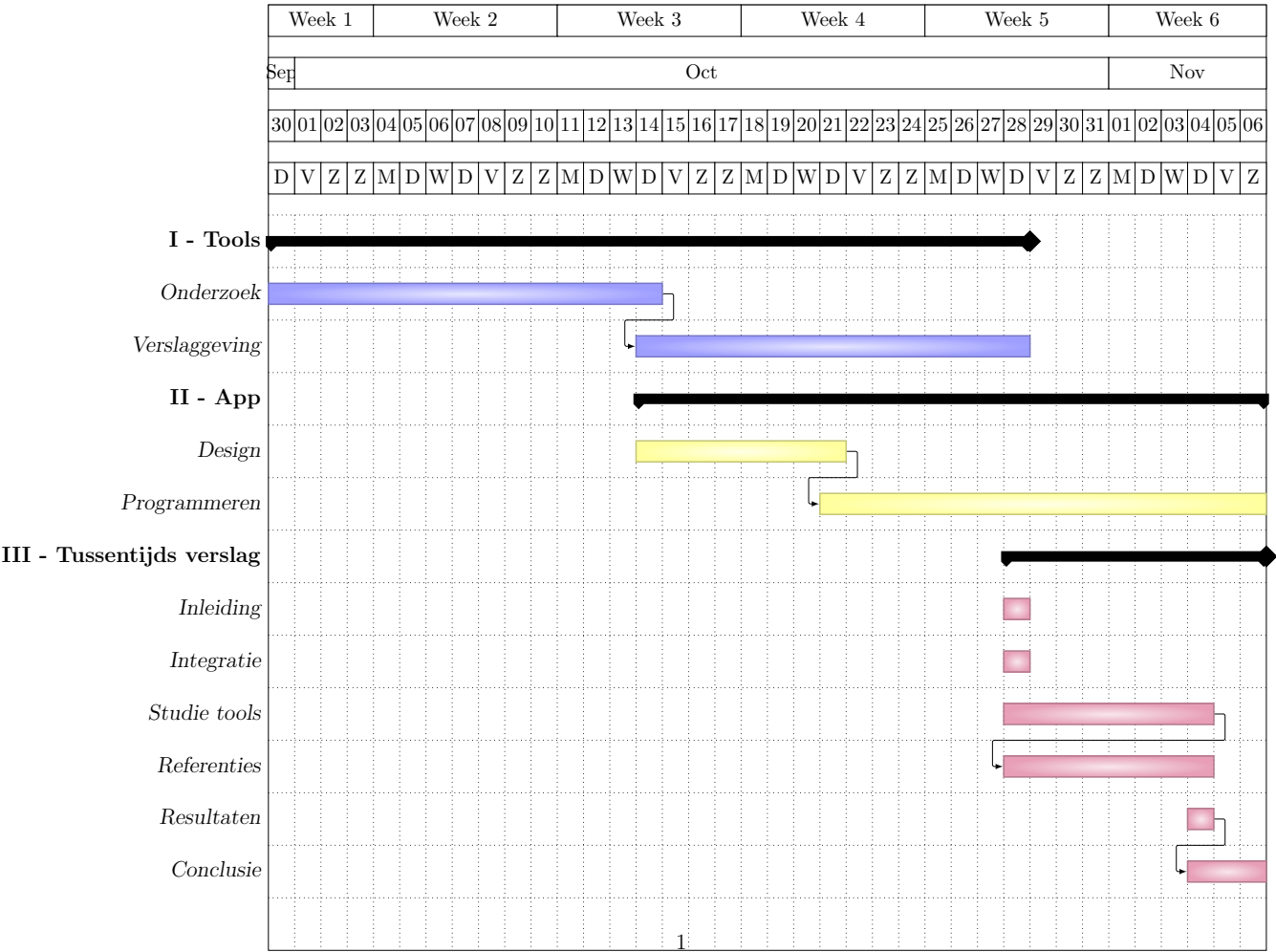
## Conclusie

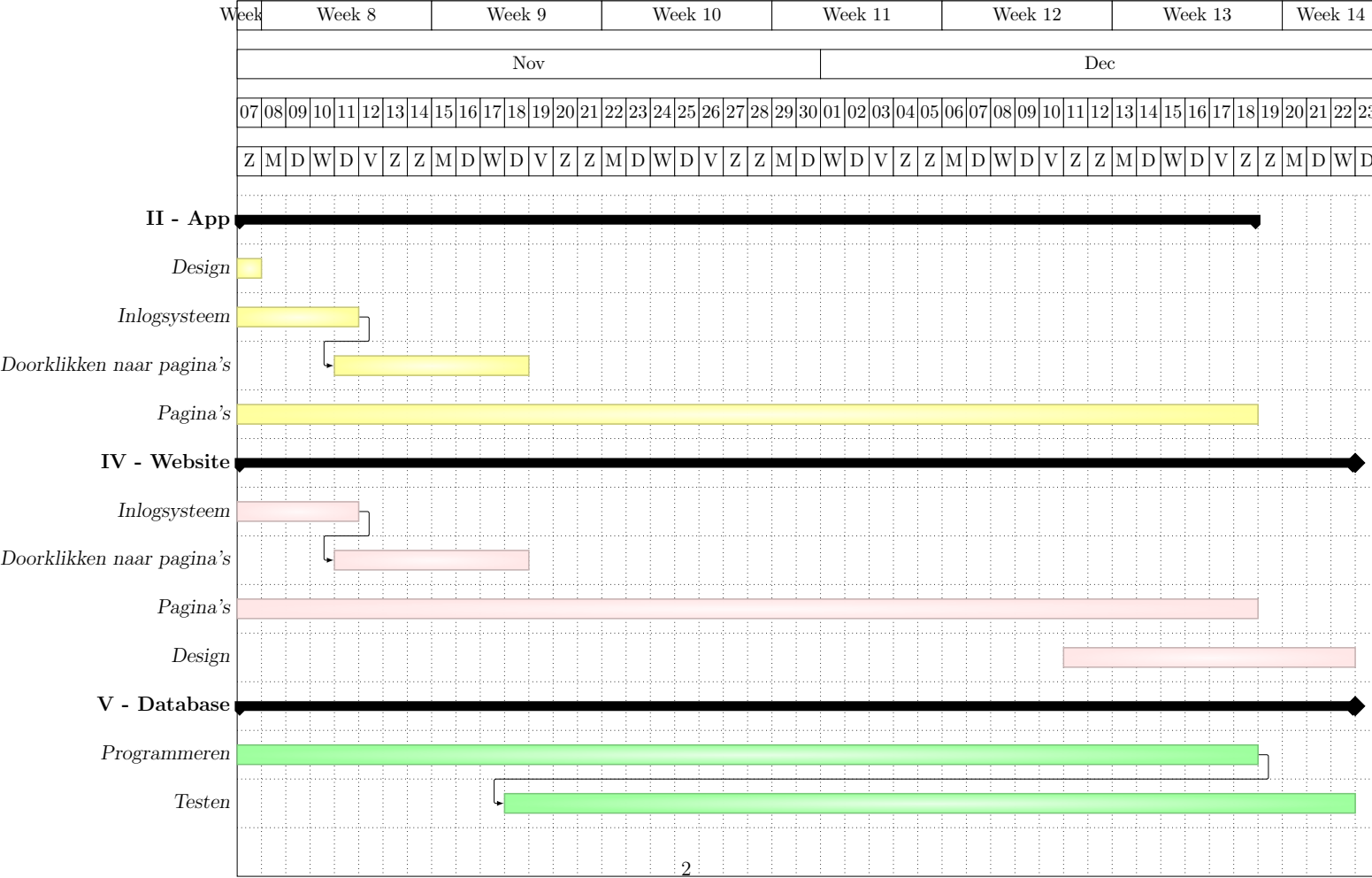
Het doel was om een communicatieplatform te vormen voor trainers en pleeggezinnen van blindengeleidehonden in wording. Honden in opleiding logeren bij pleeggezinnen en worden doorheen het dagelijks leven getraind. Tijdens die training is er niet altijd een optimale communicatie met de trainer. Daarvoor ontwikkelen wij een app en website die in connectie staan met elkaar via een database.

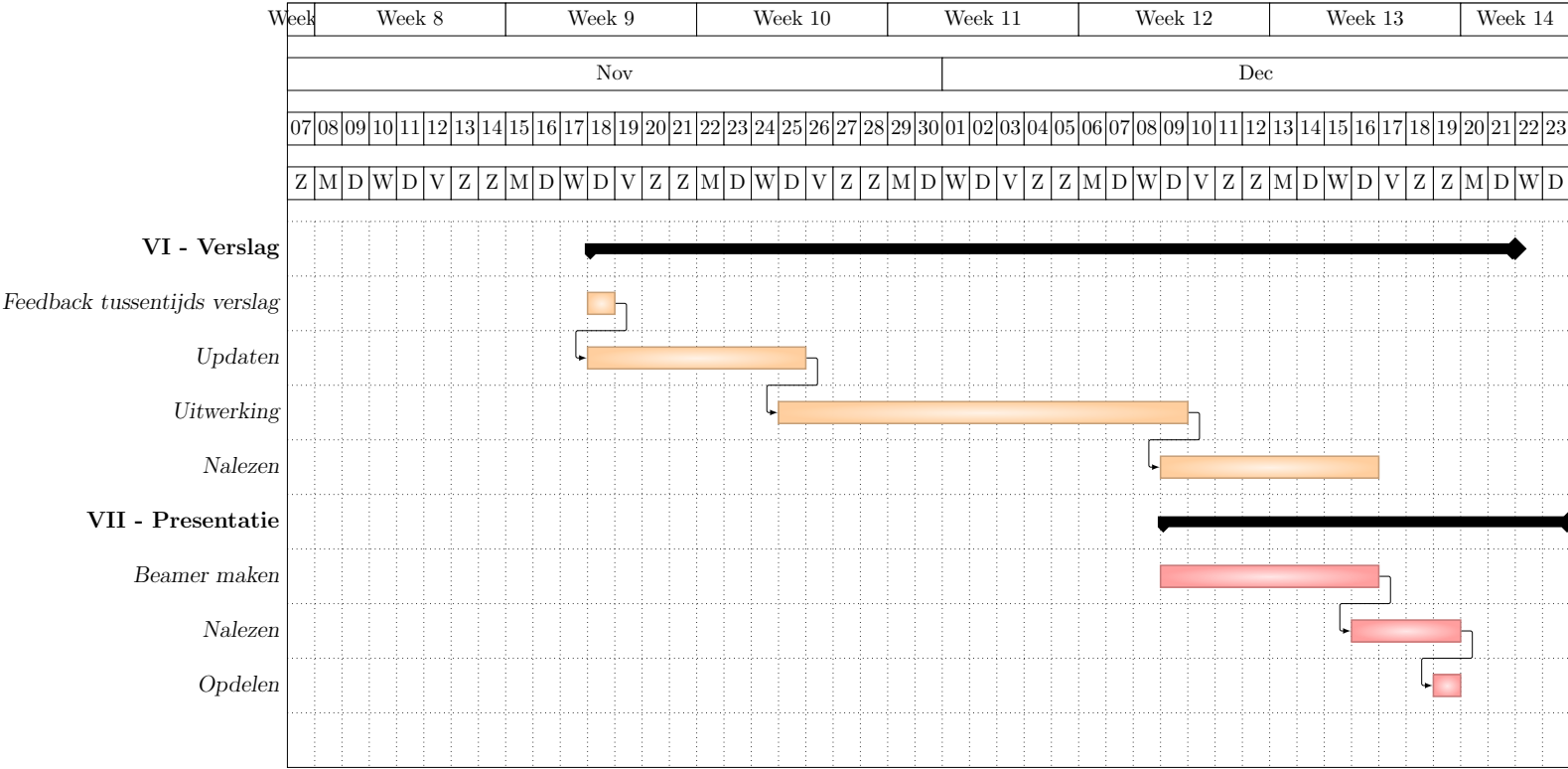
Voor het communicatieplatform te creëren was de eerste stap de tools zoeken die we konden gebruiken voor de app en website. Dit nam veel tijd in beslag aangezien dit een belangrijk gegeven was in ons project. De tools moeten kunnen voldoen aan alle eisen van de probleemstelling. Daarna was het belangrijk om kennis te maken met de werking van de tools. Voordat we begonnen aan het ontwerpen van de app en website, bouwden we een kennis op van het gebruik van de programma's. Toen we dit onder de knie hadden, konden we beginnen met het echte werk. We brainstormden over hoe we het project het best konden aanpassen. We dachten na over de structuur van de app en website. We bedachten ook hoe we de verschillende functies die de trainers en pleeggezinnen moeten kunnen doen, gingen implementeren. Tot slot begonnen we met het design en het implementeren van de app en website.

We hebben dus nog veel werk voor de boeg. De app zijn chat- en dagboekfunctie moeten nog worden geïmplementeerd. Voor de website moeten de kalender- en documentenfunctie nog worden gebouwd. Daarnaast moeten de pagina's waaraan we zijn begonnen nog worden afgewerkt. Ook moet er nog rekening worden gehouden met wie er inlogt. Als de trainer inlogt, moet hij de geschikte functies kunnen uitvoeren. Dit geldt ook voor het pleeggezin.

# Appendices









Code	Taak	Status
1	Tools	OK
1.1	Onderzoek	OK
1.1.1	Methoden om website en app te linken	OK
1.1.2	Onderzoek over databases	OK
1.2	Verslaggeving	OK
1.2.1	Takenstructuur	OK
1.2.2	Gantt-chart	OK
1.2.3	Taakverdeling	OK
2	Rapportering	niet OK
2.1	Tussentijds verslag	OK
2.1.1	Inleiding	OK
2.1.2	Integratie	OK
2.1.3	Studie Tools	OK
2.1.4	Referenties	OK
2.1.5	Resultaten	OK
2.1.6	Conclusie	OK
2.1.7	Nalezen	niet OK
2.2	Eindverslag	niet OK
2.2.1	Inleiding	niet OK
2.2.2	Integratie	niet OK
2.2.3	Studie Tools	niet OK
2.2.4	Referenties	niet OK
2.2.5	Resultaten	niet OK
2.2.6	Conclusie	niet OK
2.2.7	Nalezen	niet OK
2.3	Eindpresentaie	niet OK
2.3.1	Beamer maken	niet OK
2.3.2	Opdelen	niet OK
2.3.3	Inoefenen	niet OK

Code	Taak	Status
3	Website	niet OK
3.1	Design	niet OK
3.1.1	Inlogsysteem	niet OK
3.1.2	Dagboekje	niet OK
3.1.3	Kalender	niet OK
3.1.4	Chat optie	OK
3.2	Programmeren	niet OK
3.2.1	Inlogsysteem	OK
3.2.2	Dagboekje	niet OK
3.2.3	Kalender	niet OK
3.2.4	Chat optie	OK
3.2.5	Doorverwijzingen	niet OK
4	App	niet OK
4.1	Design	niet OK
4.1.1	Inlogsysteem	OK
4.1.2	Dagboekje	OK
4.1.3	Kalender	OK
4.1.4	Chat optie	niet OK
4.2	Programmeren	niet OK
4.2.1	Inlogsysteem	OK
4.2.2	Dagboekje	niet OK
4.2.3	Kalender	niet OK
4.2.4	Chat optie	niet OK
4.2.5	Doorverwijzingen	OK
5	Database	niet OK
5.1	Programmeren	niet OK
5.2	Testen	niet OK

## Referenties

- [1] Charlotte Deconinck. Opgave communicatieplatform. 2021.
- [2] Programmeertaal. <https://programmeerplaats.nl/>. (Accessed on 2021).
- [3] Wikipedia. <https://nl.wikipedia.org/>.
- [4] Html, css and javascript for beginners. <https://www.freecodecamp.org/news/html-css-and-javascript-explained-for-beginners/>. Accessed on 10/08/2021.
- [5] Meest gebruikte programmeertalen. <https://blog.andwork.com/wat-spreken-developers-dit-zijn-de-8-meest-gevraagde-programmeertalen/>. Accessed on 02/01/2021.
- [6] Meest gebruikte programmeertalen voor websites. <https://onlinerocket.nl/websites/top-9-programmeertalen-voor-websites/>. Accessed on 06/07/2021.
- [7] Androidstudio. <https://developer.android.com/studio>.
- [8] Youtube tutorials voor android studio: Dj malone. <https://www.youtube.com/channel/UCi3r7qQejDD-joRZCvso-Bg>.
- [9] Youtube tutorials voor android studio: Cn-code. <https://www.youtube.com/channel/UCXIQaYiP3AUQJ1SOflrGwlg/videos>.
- [10] Verschil tussen front end en back end. <https://www.code14.nl/blog/wat-is-het-verschil-tussen-front-end-en-back-end/>.
- [11] Javascript for web-application-development. <https://torquemag.io/2018/06/why-millions-of-developers-use-javascript-for-web-application-development/>. Accessed on 07/06/2018.
- [12] Html5 vs flash. <https://www.educba.com/html5-vs-flash/>.
- [13] Firestore. <https://modularfirebase.web.app/common-use-cases/firestore/>.
- [14] Django vs express. <https://www.monocubed.com/django-vs-express/>. Accessed on 24/08/2021.
- [15] Php vs node.js vs flask vs django. <https://dev.to/leviathanprogramming/php-vs-nodejs-vs-flask-vs-django-which-is-the-best-backend-language-in-your-opinion-339c>. Accessed on 02/01/2021.
- [16] Flowchart - lucidchart. <https://www.lucidchart.com>.
- [17] MongoDB. <https://www.mongodb.com/>.
- [18] Firebase. <https://firebase.google.com/>.

- [19] Firebase documentation. <https://firebase.google.com/docs>.
- [20] Pricing plans. <https://firebase.google.com/pricing>.
- [21] Choose a database: Cloud firestore or realtime database.  
<https://firebase.google.com/docs/database/rtdb-vs-firestore>.