Realisatie

Stage CanAssist

Xavier Roex Digital Innovation

Academiejaar 2022-2023

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Dit document bevat informatie over mijn realisaties als stagestudent bij het CanAssist programma van de universiteit van Victoria.

In dit verslag zal u een volledig verhaal krijgen over de projecten waaraan ik heb gewerkt, wat de bedoeling van deze projecten was maar ook welke moeilijkheden ik heb gehad tijdens het ontwikkelen van deze projecten.

Voordat we aan het verslag zelf beginnen zou ik graag nog een aantal mensen bedanken. Eerst en vooral zou ik graag Paul Green en Michiel Verboven willen bedanken voor hun begeleiding en ondersteuning tijdens mijn stage zelf. Vervolgens zou ik ook Tinne Van Echelpoel en Sarah Mcquillan willen bedanken. Deze personen hebben het mogelijk gemaakt voor mij om naar Canada te komen voor mijn stage.

Inhoud

Inleiding

Het bouwen van technologieën voor mensen met een beperking is geen gemakkelijke opdracht. Aangezien je voor eenzelfde probleem waarschijnlijk 5 oplossingen hebt voor 5 verschillende personen. Anderzijds maakt dit het ook zeer interessant natuurlijk want dit zorgt ervoor dat je ten allen tijden naar nieuwe en interessante manieren op zoek bent om de geschikte oplossing te vinden.

In dit document zal het gaan over mijn stage bij CanAssist. In het eerste deel van het document zal ik dus ook beginnen met wat meer uitleg te geven over CanAssist, de mensen die er werken, het doel dat zij voor ogen hebben.

Vervolgens zal ik het hebben over mijn eerste project namelijk het AAI of Accessible Art Interface project. Hierbij zal ik toelichten wat het project is, wat het nut is van dit project, wat ik zelf heb kunnen bijdragen aan het project en ook welke moeilijkheden ik heb ondervonden met de gebruikte technologieën.

Tenslotte zal ik overgaan naar het 2de project waar ik aan heb gewerkt. Namelijk de CanPlan applicatie. Bij dit project zal ik terug de situatie wat schetsen rond waarom dit project is opgezet. Wat ik heb kunnen bijdragen aan het project en waar de moeilijkheden lagen voor dit project.

CanAssist

De organisatie

CanAssist is geen commercieel bedrijf en heeft als doel dus ook niet om winst te maken. Zij maken een deel uit van de universiteit van Victoria. Dit zorgt ervoor dat zij dus heel wat stagestudenten hebben of “co-op” studenten.

Figuur 1

Hoofddoelen

CanAssist ontwikkelt al bijna twintig jaar technologieën en is nu in de positie om zijn bereik te vergroten door "brede impact" oplossingen te ontwikkelen die veel individuen en groepen met vergelijkbare onvervulde behoeften ondersteunen. Door zich te richten op bepaalde demografische groepen, zoals senioren en kinderen kan CanAssist zijn expertise versterken op gebieden waar de vraag toeneemt.

Het doel van CanAssist is om de reikwijdte van de campusactiviteiten verder te vergroten en meer mogelijkheden te bieden voor dynamisch leren, zinvol onderzoek, inclusieve programmering en betrokkenheid bij de gemeenschap.

CanAssist heeft aanzienlijke vooruitgang geboekt in het ontwikkelen van partnerschappen met vele overheidsinstellingen en andere belangrijke organisaties in BC. Deze partners verschaffen financiële middelen voor de ontwikkeling van technologieën en programma's die de onafhankelijkheid en levenskwaliteit van mensen verbeteren en de stress voor gezinnen en verzorgers verlichten, terwijl de vraag naar gezondheidszorg en sociale voorzieningen afneemt. CanAssist zal zijn samenwerkingsverbanden uitbreiden om een levensvatbare provinciale en nationale bron te worden, zodat de financiële duurzaamheid op de lange termijn gewaarborgd blijft.

CanAssist bouwt aan een gezonde werkomgeving waar medewerkers worden ondersteund met de juiste hulpmiddelen, processen en professionele ontwikkelingsmogelijkheden om hun verantwoordelijkheden als lid van het team effectief uit te voeren. CanAssist is altijd op zoek naar nieuwe manieren om de processen te stroomlijnen, de communicatiemethoden te verbeteren en de organisatiecultuur te verrijken.

Geschiedenis

Sinds de begindagen heeft CanAssist aangepaste technologieën en softwaredownloads geleverd aan duizenden mensen, waaronder baby's, kinderen, volwassenen en senioren. Het team richt zich meestal op klanten uit British Columbia, maar heeft ook mensen met een handicap uit andere plaatsen in Canada en de Verenigde Staten geholpen, maar ook uit verre landen als Nieuw-Zeeland, China, Schotland en Kenia.

In 2014 had de organisatie bijna 6.000 studenten betrokken via co-op, afstudeer- en werkstudieplaatsen, presentaties, cursusonderricht, overzeese studieprogramma's en vrijwilligerswerk. Ook veel faculteitsleden van UVic zijn door de jaren heen betrokken geweest bij CanAssist-activiteiten, bijvoorbeeld op het gebied van onderzoek, klassikale instructie, verzoeken om technologische ontwikkeling en het voortdurend delen van ideeën en expertise.

CanAssist beantwoordt nog steeds regelmatig verzoeken van particulieren uit de gemeenschap, maar heeft de afgelopen jaren ook op maat gemaakte technologische oplossingen geleverd aan een aantal organisaties. Meestal zijn onze relaties met deze organisaties gebaseerd op contracten - waarbij onze diensten op doorlopende basis worden geleverd - of op projectspecifieke subsidies. Tot de organisaties behoorden organisaties uit de publieke, private, filantropische en non-profitsector. Dergelijke partnerschappen zijn voordelig gebleken voor de organisaties, voor CanAssist's voortdurende duurzaamheid en voor de individuele cliënten die door deze organisaties worden gefinancierd, hun families en de gehandicaptengemeenschap als geheel.

De Universiteit van Victoria heeft haar betrokkenheid bij CanAssist en de gehandicaptengemeenschap opnieuw bevestigd door de organisatie aan te wijzen als hoeksteen van het nieuwe Centre for Athletics, Recreation and Special Abilities, ook wel het CARSA-gebouw genoemd. CanAssist heeft zijn intrek genomen in het nieuwe gebouw toen het in mei 2015 werd opgeleverd.

Figuur 2

Het Engineering team

Vervolgens zou ik jullie graag willen voorstellen aan het CanAssist Engineering team. Oftewel het team waar ik de voorbije maanden mee heb samengewerkt. Onderstaande foto is van het team waarmee ik in mei heb samengewerkt. De reden dat dit anders is dan maart en april, is omdat de academische kalender anders is in Canada.

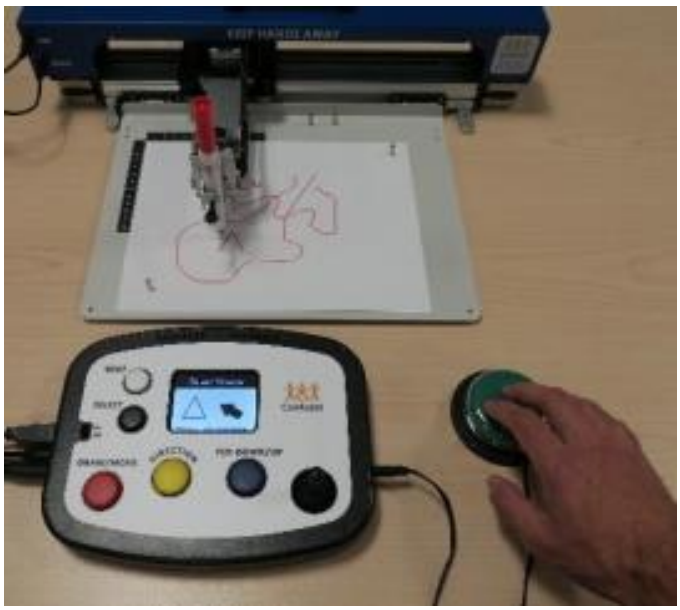
Figuur 3

De personen op deze foto van links naar rechts zijn: Alexander Velasco (electrical-engineering co-op), Summer Johnson (mechanical-engineering), Nicholas Bowen (mechanical-engineering co-op), Paul Green (Engineering supervisor), Logan Costa-Hemingway (electrical-engineering), Mezelf en Kaden Taylor (electrical-engineering co-op)

Accessible Art Interface

Inleiding

Dit deel van het document zal gaan over het eerste project waar ik aan gewerkt heb, namelijk: AAI of Accessible Art Interface. Dit is een project bedoeld om kinderen met een fysieke of mentale beperking een artistieke uitlaatklep te geven in de vorm van machine en controller waarmee zij kunnen tekenen. Onderstaande foto toont het product dat wordt afgeleverd.



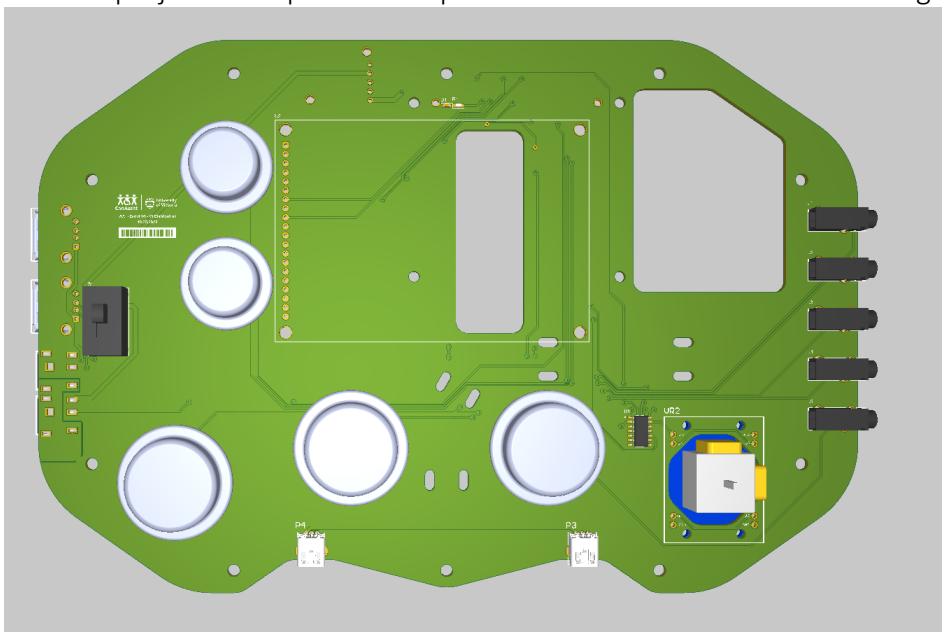
Figuur 4

Hardware & Software

In dit deel van het document zal ik in het kort wat belangrijke stukken hardware en software bespreken die noodzakelijk waren voor het project en voor mijn contributies. Dit product bestaat uit een AxiDraw tekenmachine die wordt aangekocht en een controller die zij volledig zelf maken. De controller zelf bestaat uit een pcb, een raspberry pi, 5 knoppen en een joystick. Qua software is het project volledig geschreven in Python. Gezien mijn voorgaande werk en projecten was dit dus zeer aangenaam om aan te starten. Er is echter wel een onderscheid tussen het bestaande project dat is gemaakt door CanAssist en de firmware voor de Axidraw die geschreven is door de fabrikant “EvilMadScientist”.

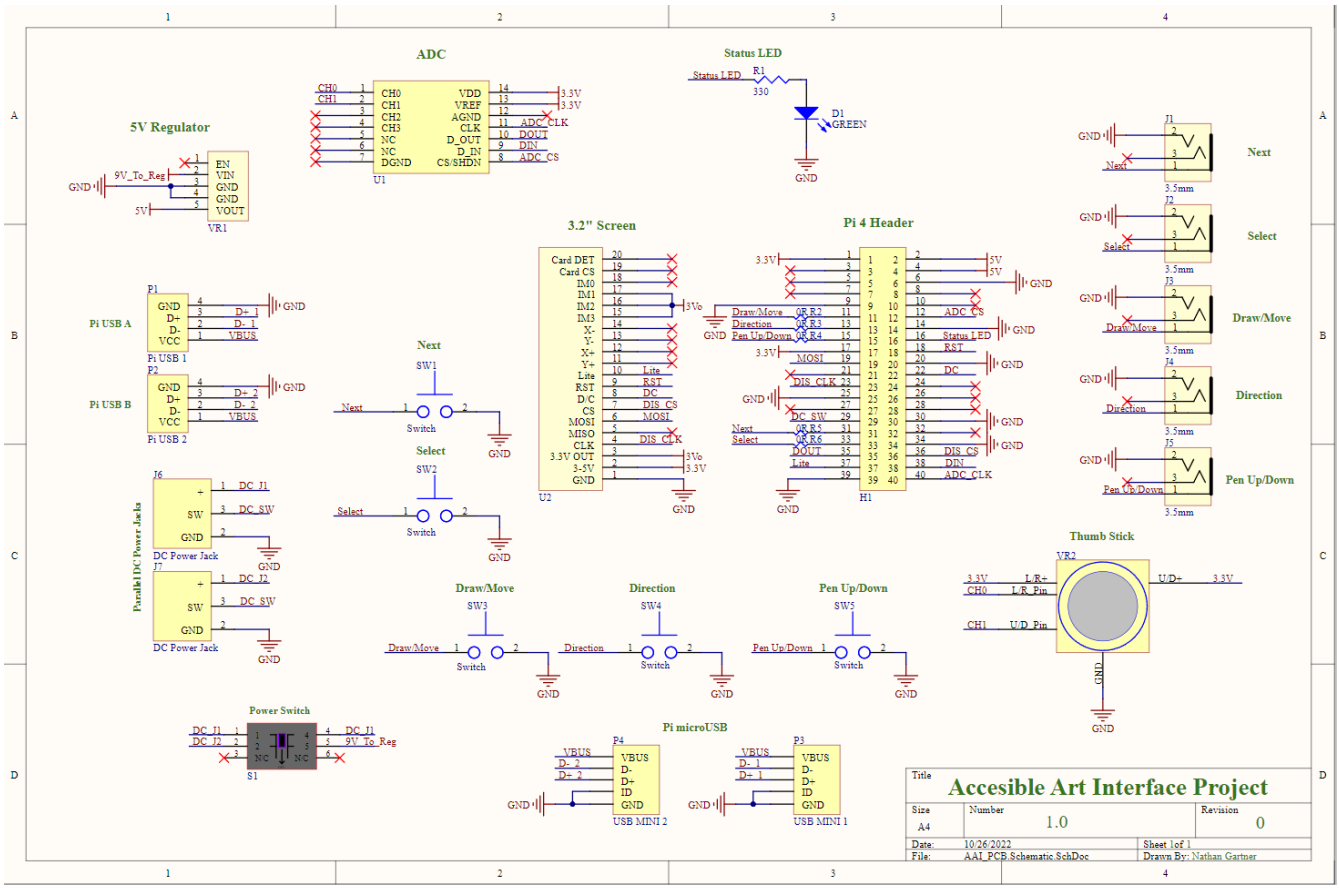
PCB

Voor dit project is er speciaal een pcb ontwikkelt. Deze ziet eruit als volgt:



Figuur 5

Hoewel mijn expertise niet bij PCB's of altium designer ligt heb ik het volgend schema wel goed kunnen gebruiken. Alsook kon ik hiervoor rekenen op de ondersteuning van Nathan Gartner een vaste werknemer in Electrical engineering die spijtig genoeg eind maart van CanAssist is weggegaan.

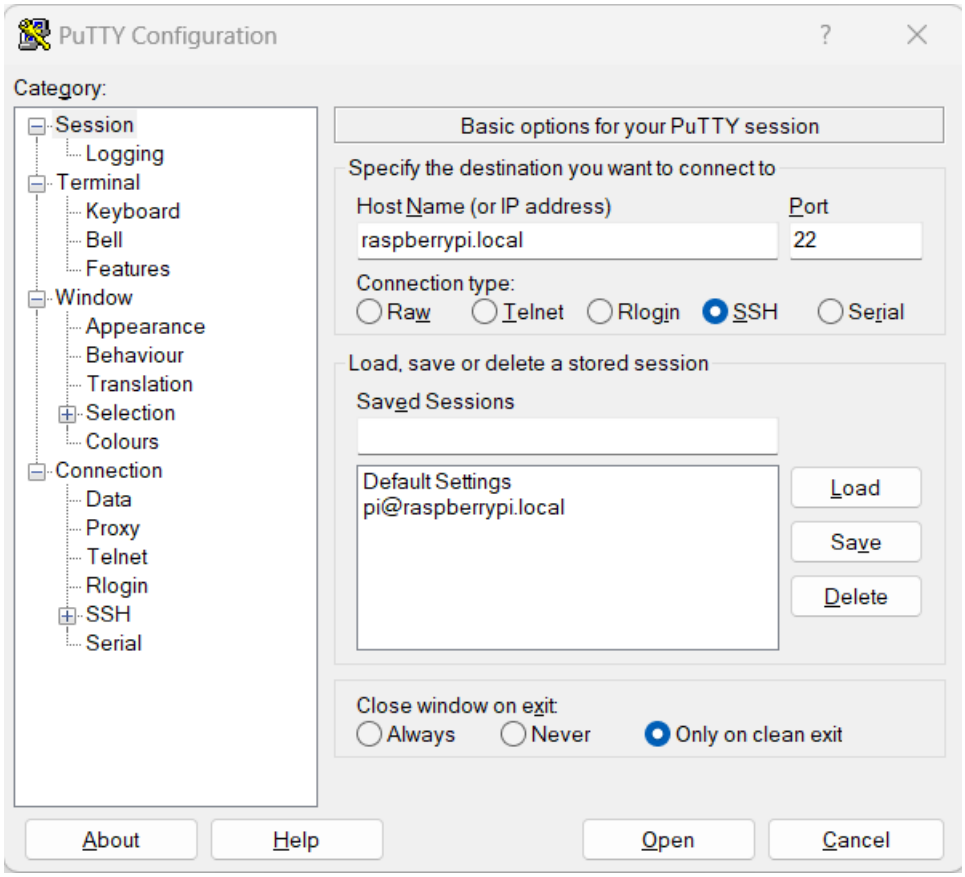


Figuur 6

Het bovenstaande schema toont namelijk hoe alles verbonden is met elkaar. Hieruit kunnen we dus informatie halen die belangrijk is om bijvoorbeeld een kortsluiting te vinden, en aangezien raspberry pi's hier toch wel wat problemen mee hebben was dit dus zeer welkome documentatie.

Putty

Putty was de software die ik heb gebruikt om een SSH-verbinding op te zetten naar een raspberry pi. Dit is software die in de IoT wereld redelijk veel gebruikt wordt en gaat als volgt in zijn werk.



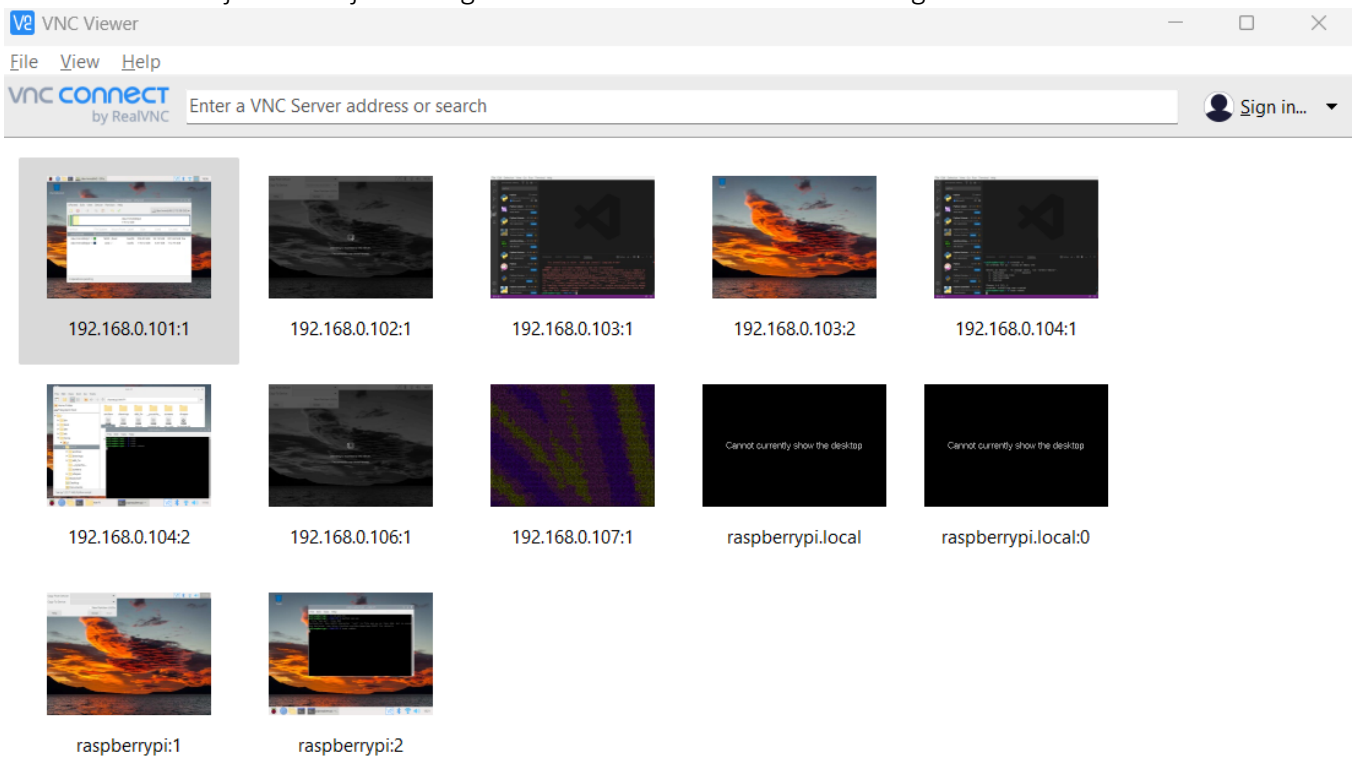
Figuur 7

Op dit scherm geef je een hostname of IP-adres in. Voor mij was dit raspberrypi.local en kies je de juiste poort. De poort voor SSH-verbinding is 22.

VNC-viewer

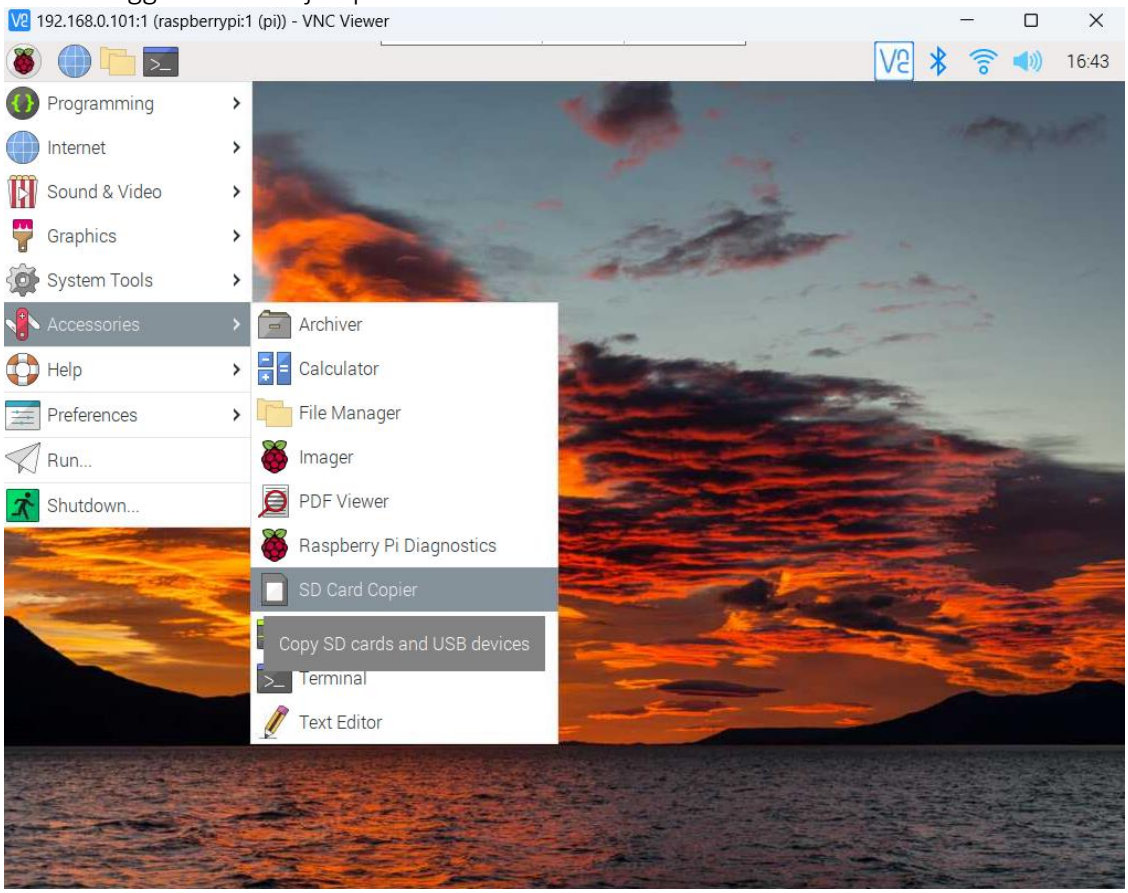
Deze software staat toe om een GUI of graphical user interface te creëren voor een raspberry pi. Aangezien dit project gebruikt maakt van een raspberry pi was het dus een goed idee om dit te gebruiken. Toen ik eerst bij CanAssist toekwam werd dit nog niet gebruikt. Deze

software is redelijk makkelijk om in gebruikt te nemen en ziet er als volgt uit:



Figuur 8

Op dit scherm kan je de IP-adressen of hostnames zien waarmee je eerder al eens verbonden hebt en in de balk bovenaan kan je nieuwe IP-adressen of hostnames ingeven. Dan is het nog enkel inloggen voordat je op dit scherm terechtkomt. De GUI.



Figuur 9

Achtergrondinformatie

CanAssist heeft heel wat technologieën voor kinderen die vooral bedoeld zijn dingen zoals tekenen, muziek, speelgoed, … zo inclusief mogelijk te maken. Dit is hier een voorbeeld van. Ikzelf heb meegewerkt aan V2. Dit heeft als reden dat er een soort van MVP klaar was maar dat naar de toekomst toe ze toch liever wat meer functionaliteiten wilden hebben.

Zoals vermeld in de inleiding is dit project vooral bedoeld voor kinderen met een beperking om een artistieke uitlaatklep te hebben. De meest voor de hand liggende doelgroep is kinderen met een fysieke beperking waarbij de beperking het voor hen onmogelijk maakt om een pen vast te houden of stabiel genoeg te houden om precies te tekenen, maar dit product is ook ontwikkeld voor kinderen met een mentale beperking. Dit is omdat het bepaalde functionaliteiten zoals de mogelijkheid om een .svg bestand zowel in 1 keer of in stukken te laten tekenen. Dit zorgt ervoor dat het nog altijd interessant is voor een bredere doelgroep.

Ik was zeer tevreden toen ik erachter kwam dat dit project met een raspberry pi was omdat ik hier zelf ook al enige voorkennis mee had. Deze voorkennis heeft ervoor gezorgd dat ik redelijk snel ingelezen was in het project.

Mijn contributies

Bij dit project heb ik mijn tijd grotendeels gespendeerd aan het toevoegen van functies. Deze functies waren grotendeels functies om het gemak van de gebruiker te verhogen.

Maar tijdens het ontwikkelen heb ik ook altijd wat tijd genomen om de code uniform te maken en ervoor te zorgen dat het relatief goed gestructureerde en dynamische code is. Omdat er voor mij al minstens 3 programmeurs aan hebben gewerkt met elk hun eigen stijl was de code met momenten zeer moeilijk leesbaar. Dus heb ik per functie en per feature gezocht naar welke code mij het beste leek en heb ik deze stijl overgenomen.

Een voorbeeld hiervan is het toevoegen van joystick functionaliteit in de menu. Maar ook heb ik gewerkt aan enkele features die het veiliger maken zoals een noodstop toevoegen wanneer de AAI automatisch tekent.

Maar waar ik eerst mee begonnen ben is het opzetten van een gemakkelijkere manier om te kunnen ontwikkelen. Aangezien de controller gebruikt maak van een raspberry pi leek het mij een goed idee om te werken met VNC server/viewer.

Figuur 10

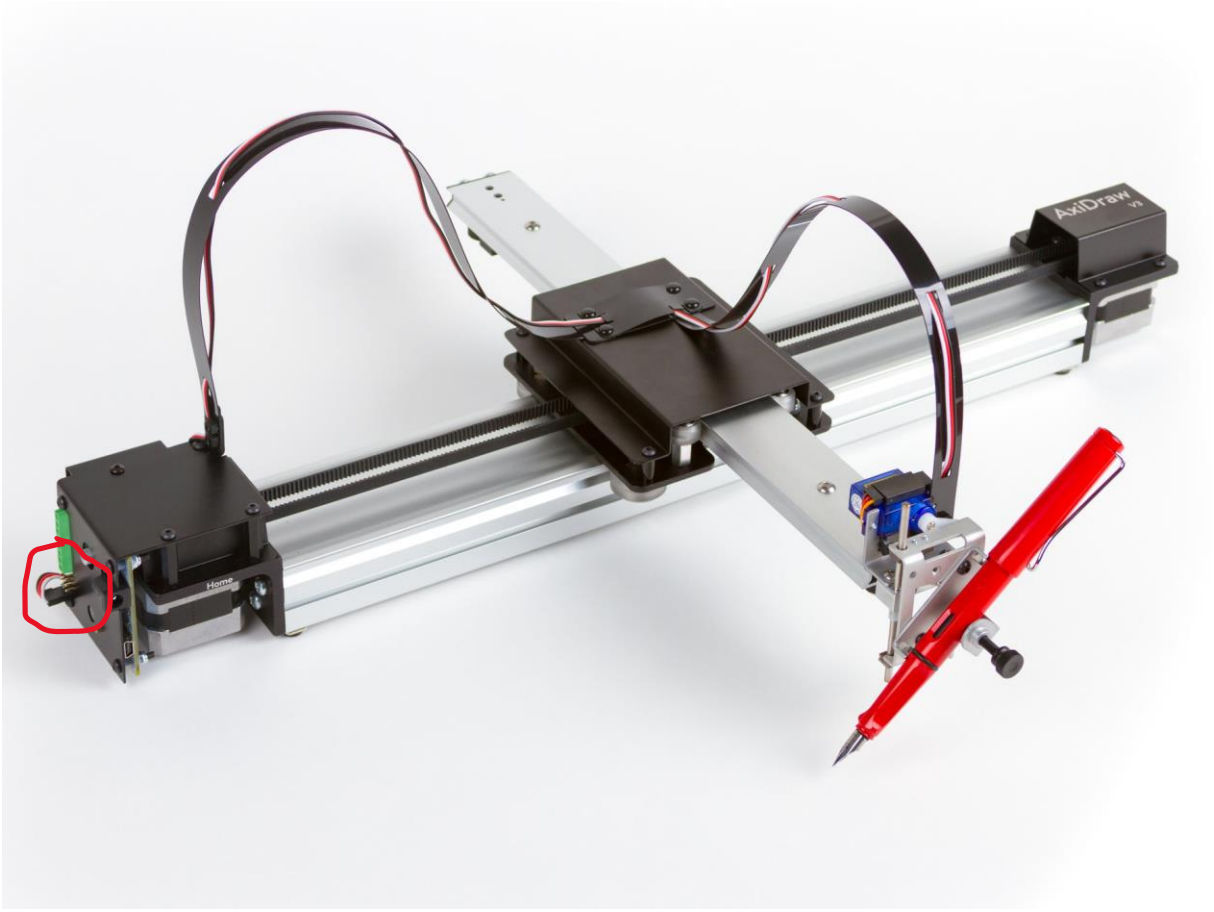
Waar ik niet bij had stil gestaan was dat de raspberry pi en de OS die hierop draait al relatief oud is. Hierdoor heb ik dus wat meer moeten uitzoeken over hoe ik VNC-server op een oudere versie van RaspbianOS kon krijgen. Een probleem dat zich op dit ogenblik ook stelde was dat UVIC het netwerk nogal hard beveiligd. Dit zorgt ervoor dat zelfs de SNTP-servers geblokkeerd zijn. Dit bracht natuurlijk heel wat problemen met zich mee. Uiteindelijk is dit opgelost geraakt door de DNS adressen te wijzigen naar UVIC DNS servers. Hoewel het probleem met de tijd niet opgelost geraakt is zorgde dit er wel voor dat ik eindelijk packages kon downloaden, inclusief de package voor de VNC-server.

Nadat dit eenmaal in orde was kon ik dus beginnen met ontwikkelen. Zoals eerder vermeld heb ik dus een aantal nieuwe features toegevoegd aan dit product op de komende pagina's zal u kunnen lezen welke dit exact waren wat de moeilijkheden en wat de uiteindelijke oplossing hiervoor was.

De eerste features waarmee ik begonnen ben waren redelijk gemakkelijke features om mee te beginnen en om wat in te werken in het project. Namelijk het kunnen gebruiken van de joystick in de menu en de Axidraw automatisch laten homen bij het opstarten (tot op dat moment wachtte de Axidraw totdat 2 knoppen ingeduwd werden). Een niet onbelangrijke feature, want wat voor nut heeft de joystick als deze maar voor de helft van de tijd gebruikt kan worden. Deze features waren redelijk simpel om in te voegen en waren dus ook een goed start van het project. U zal kunnen lezen dat mijn werkwijze voor het toevoegen altijd hetzelfde is gebleven namelijk dat ik eerst de code rustig doorneem om te gaan zoeken naar waar ik best de feature kan toevoegen en welke code ik moet aanpassen of herschrijven. Voor deze features was het redelijk simpel en redelijk weinig aanpassingen. Namelijk voor de automatische homing bestond de functie al en was het gewoon het stuk van de check voor de code van de 2knoppen niet te verwijderen maar in commentaar te zetten (een goede keuze waar ik later zeer tevreden mee was) en de home functie mee in de initialisatie fase van het programma steken.
Voor het toevoegen van de joystick functionaliteit in de menu's was het ook redelijk simpel omdat bijna alle achterliggende dingen voor de joystick al gedaan werden. Ik moest dus nog enkel checken of de juiste as naar boven of naar beneden gingen en dit laten uitlezen op de juiste plekken.

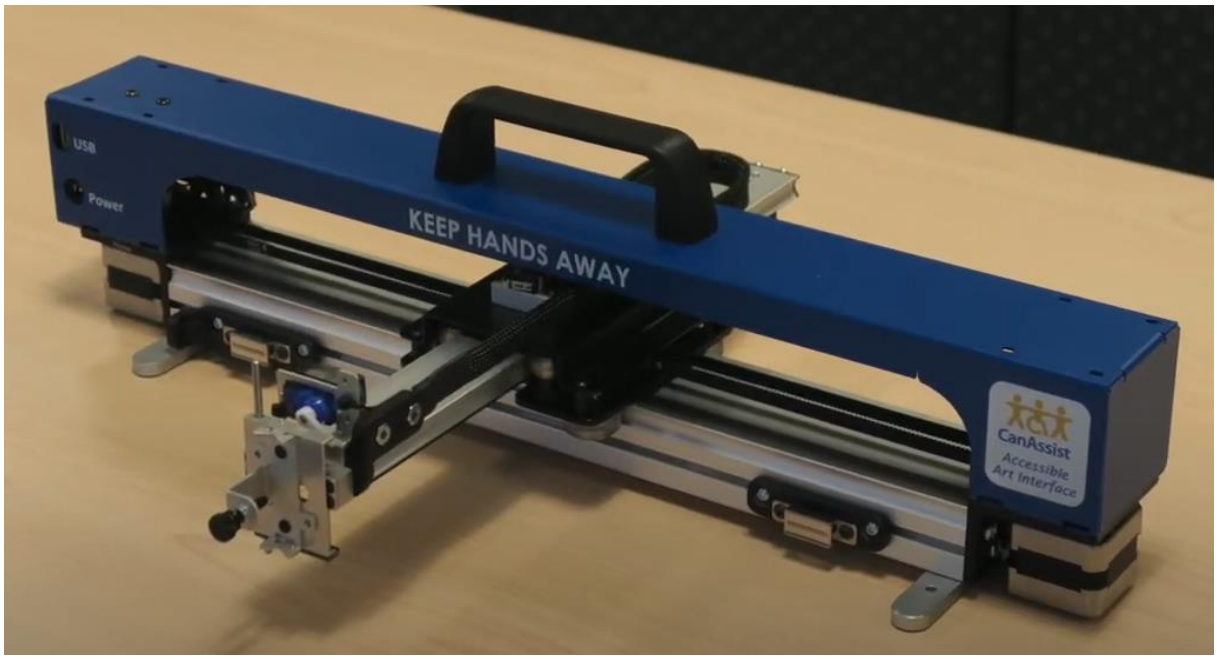
Vervolgens wilde ze de mogelijkheid om eigen tekeningen te kunnen laden vanop een USB-stick. Hier was al een beetje aan gewerkt. De oplossing die op dat moment was geïmplementeerd was dat de code ging checken of er een USB aanwezig was en als deze er was toonde de AAI enkel wat er op de USB beschikbaar was. Om terug naar de interne opslag te gaan moest je dus de USB terug uitnemen. Mijn manier van werken om dit probleem op te lossen en de nieuwe feature toe te voegen was eigenlijk vooral om me eerst wat in te werken in de code en wat te analyseren alvorens enige wijzigingen te maken. Dit was achteraf gezien de juiste keuze, zeker aangezien er in totaal een 5 mogelijke bestanden waren waarin deze feature kon toegevoegd worden van elks tussen de 300 en de 1000 lijnen python code. Verder heb ik gebruikt gemaakt van de bestaande code en heb ik de huidige functies aangepast. Want er werd al gecheckt of er een USB aanwezig is. Dus heb ik een menu toegevoegd die je laat kiezen of je interne of externe opslag wil gebruiken.

De volgende feature die geïmplementeerd moest worden was de noodstop voor het automatisch tekenen. Dit is een redelijk belangrijke toevoeging omdat dit product vooral bedoeld is voor kinderen met een beperking die niet altijd begrijpen dat het gevaarlijk kan zijn als de motoren niet gestopt zijn. Om dit te implementeren moest ik in nog andere files kijken. De files waar ik nu in moest kijken waren die van de Axidraw zelf. Dit was in totaal nog een 4 files van elk een 3000 à 4000 lijnen code. Omdat ik terug de code heb geanalyseerd ben ik erachter gekomen dat er een fysieke noodstop aanwezig zou moeten zijn. Na wat opzoekingen ben ik erachter gekomen waar deze zou moeten staan.



Figuur 11

Tot zover het goede nieuws. Het slechte nieuws is namelijk dat de afgewerkte AAI er als volgt uitziet.



Figuur 12

Wat u dus opmerkt is dat er een extra omhulsel is om de fragile kabels te beschermen. Waar er hardware matig dus geen rekening mee gehouden was, was dat de noodknop nu dus achter dit omhulsel zit. Nadat ik dit met mijn manager heb besproken was er besloten om dus verder naar een softwareoplossing te zoeken. Er was echter al wel een voordeel, omdat de noodknop aanwezig is wil dit ook zeggen dat er code aanwezig is om een noodstop uit te voeren. Dus moest ik dus naar een manier zoeken om een knop op de controller te linken aan het signaal van de noodknop. Dit bleek echter ook een probleem te zijn. Dit probleem leek zich voor te doen door de manier waarop de noodstop was geïntegreerd. Na heel wat brainstormen ben ik dan uiteindelijk bij de oplossing gekomen. Een relatief invasieve oplossing namelijk een OS-commando sturen om het tekenproces te stoppen. De reden dat dit werkt omdat het tekenen een volledig apart proces was en dus veilig gestopt kon worden zonder het hele programma te stoppen.

Nu stopte de AAI al maar bleef die op zijn positie staan en konden we ook niet verder tekenen. Dus heb ik hier ook weer een menuscherm aan toegevoegd om verder te kunnen gaan, maar eerst heb ik er ook voor gezorgd dat de homing functie wordt geroepen als de noodstop ingeduwd wordt. Dit was moeilijker dan het lijkt omdat dit communicatie vereist tussen de software van de Axidraw zelf en de CanAssist code.

Toen dit eenmaal klaar was had mijn supervisor een volgende taak klaarliggen voor mij. Deze was om een mogelijkheid te voorzien om je eigen tekeningen te kunnen opslaan. Dit was wel wat moeilijker dan het op eerste zicht lijkt. Dit is omdat alle tekeningen die tot op dit ogenblik bestaan SVG-bestanden zijn. Oftewel foto's die gevectoriseerd zijn. Hoewel dat op zich geen probleem is, is het wel een probleem dat ik voor de eigen tekeningen dus geen foto heb om een vector van te maken. Dus moest ik een andere oplossing bedenken. Na wat brainstormen en opzoekingswerk zag ik dat in de code van de Axidraw dat ik mogelijks de stappen die doorgestuurd worden naar de Axidraw eruit kon halen. Eens dat dit in orde was het nog belangrijk

om deze correct op te slaan. Aangezien dit dus ook dynamisch opgeslagen moet worden er geen toetsenbord aanwezig is heb ik bespreking met mijn supervisor ervoor gekozen om de benaming standaard te houden en een getal op het einde te verhogen per nieuwe afbeelding.
Het opslaan van de tekening zelf gaat dus als volgt in zijn werk. Als de gebruiker ervoor kiest om de tekening op te slaan wordt er een nieuw bestand geopend op de juiste plek. Dan als de gebruiker de Axidraw beweegt worden deze stappen niet alleen naar de Axidraw gestuurd maar ook naar het bestand. Dus is elke lijn in het bestand gewoon een volgende stap voor de Axidraw. Dit zorgt er dus ook voor dat het opnieuw tekenen hiervan zo simpel is als het uitlezen en rechtstreeks doorsturen van de commando's in dit bestand. Een voorbeeld van zo'n bestand is de foto op de volgende bladzijde.

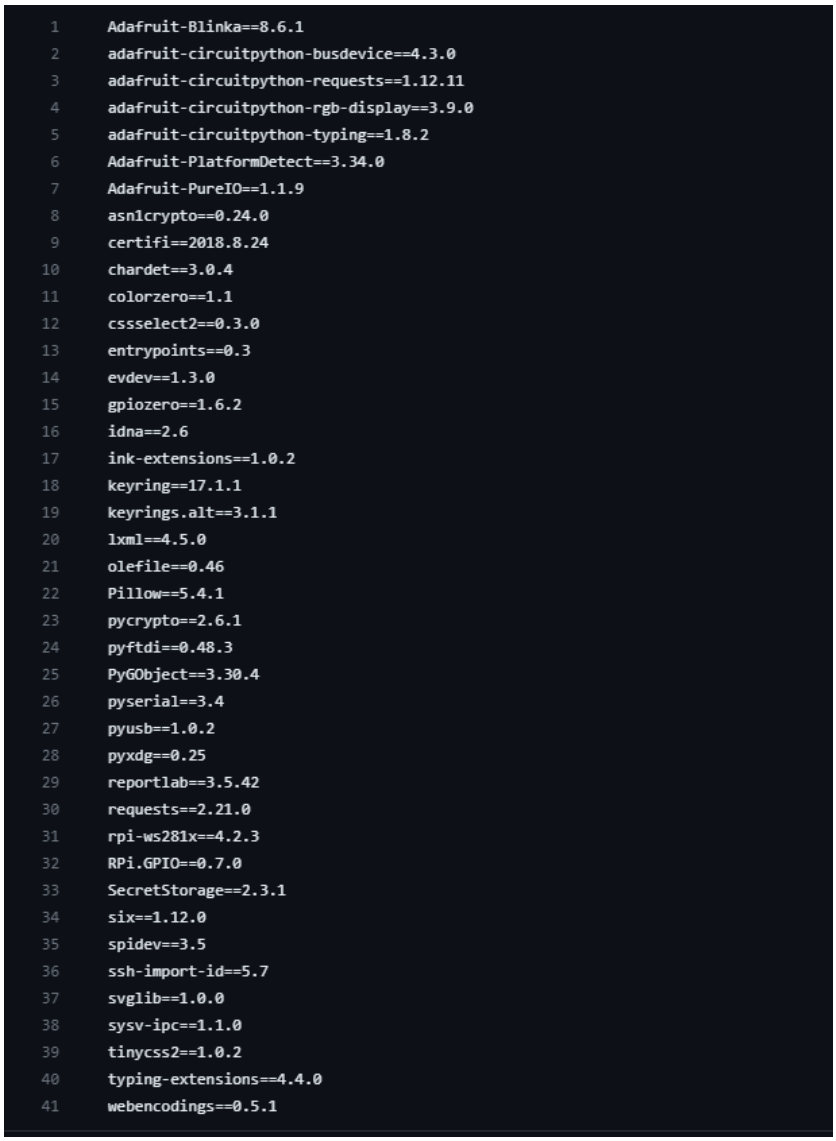
```
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,2,0
XM,25,3,0
XM,25,3,0
XM,25,3,0
XM,25,3,0
XM,25,3,0
XM,25,3,0
XM,25,3,0
XM,25,4,0
XM,25,4,0
XM,25,4,0
XM,25,5,0
XM,25,5,0
XM,25,5,0
XM,25,6,0
XM,25,13,0
XM,25,23,0
XM,25,26,2
XM,25,30,3
XM,25,33,5
XM,25,37,8
XM,25,43,14
XM,25,45,20
XM,25,47,25
XM,25,50,29
XM,25,53,34
XM,25,55,40
XM,25,57,43
XM,25,61,46
XM,25,62,48
XM,25,62,50
XM,25,65,54
XM,25,69,60
XM,25,74,65
XM,25,77,71
XM,25,79,76
XM,25,81,83
XM,25,84,86
XM,25,88,93
XM,25,89,96
XM,25,89,97
XM,25,90,102
XM,25,90,105
XM,25,91,106
XM,25,91,107
```

Figuur 13

Op deze foto ziet u dus hoe de commando's zijn opgebouwd.

Toen ik hiermee klaar was waren de nieuwe SD-kaartjes toegekomen en heb ik mijn baas Paul overtuigd gekregen om mij de RaspbianOS versie naar de laatste nieuwe te krijgen. Dit kwam op zich ook met heel wat problemen. Deze problemen waren een resultaat van hoe de SD kaart in orde was gezet voor de eerste versie en hadden betrekking tot partities en Linux file systems. Alsook heb ik heel wat problemen ondervonden met GitHub om de code hier goed en wel op te krijgen. Nadat de code eenmaal hierop stond was er niet echt meer een probleem. Dus heb ik voor de nieuwe RaspbianOS versie ook gekozen voor een iets andere setup zodat dezelfde problemen vermeden kunnen worden naar de toekomst toe.

De nieuwe RaspbianOS versie zelf zorgde eigenlijk voor geen echte problemen met de code maar wel kort even met de setup en installatie hiervan. Omdat sommige packages andere versies hadden die niet outdated of deprecated waren. Dus deze packages updaten of een andere oplossing hiervoor vinden was de oplossing. Om deze setup iets vlotter te laten gaan in de toekomst heb ik ook een requirements file aangemaakt met alle python packages. Deze ziet er zo uit:

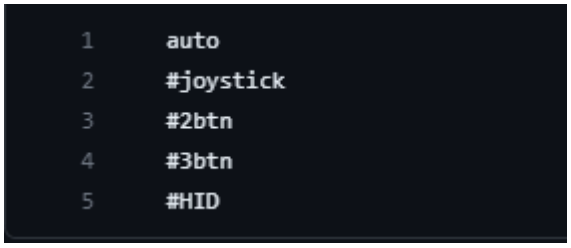


```
Adafruit-Blinka==8.6.1
adafruit-circuitpython-busdevice==4.3.0
adafruit-circuitpython-requests==1.12.11
adafruit-circuitpython-rgb-display==3.9.0
adafruit-circuitpython-typing==1.8.2
Adafruit-PlatformDetect==3.34.0
Adafruit-PureIO==1.1.9
asn1crypto==0.24.0
certifi==2018.8.24
chardet==3.0.4
colorzero==1.1
cssselect2==0.3.0
entrypoints==0.3
evdev==1.3.0
gpiozero==1.6.2
idna==2.6
ink-extensions==1.0.2
keyring==17.1.1
keyrings.alt==3.1.1
lxml==4.5.0
olefile==0.46
Pillow==5.4.1
pycrypto==2.6.1
pyftdi==0.48.3
PyGObject==3.30.4
pyserial==3.4
pyusb==1.0.2
pyxdg==0.25
reportlab==3.5.42
requests==2.21.0
rpi-ws281x==4.2.3
RPi.GPIO==0.7.0
SecretStorage==2.3.1
six==1.12.0
spidev==3.5
ssh-import-id==5.7
svglib==1.0.0
sysv-ipc==1.1.0
tinycss2==1.0.2
typing-extensions==4.4.0
webencodings==0.5.1
```

Figuur 14

En zorgt ervoor dat ik niet meer alle packages 1 voor 1 moet installeren maar gewoon een pip install van dit bestand kan doen.

De laatste functionaliteit die zij wilden was om een soort van setup file te hebben zodat de AAI automatisch in de juiste modus kan opstarten. In deze setup file moest er ook de mogelijkheid zijn om de automatische homing uit te schakelen en terug te vragen naar het indrukken van de 2 knoppen. Voor deze feature ben ik gegaan voor de meest simpele oplossing namelijk een bestand aanmaken dat er uit ziet als volgt:



Figuur 15

De reden dat dit zo simpel is omdat er niet veel veranderd moet worden, en het weinige dat veranderd moet worden kan zeer gemakkelijk aangepast worden met Python. Dus ga ik in de initialisatie fase dit document ophalen en check ik voor welke modus er ingeladen moet worden. Aangezien het aanpassen van dit document waarschijnlijk niet door de eindgebruiker zelf zal gebeuren is de menu om dit aan te passen ook verborgen achter een bepaalde combinatie van knoppen. Dit zorgt ervoor dat problemen hiermee niet zo gemakkelijk voorkomen.

Tegen het einde van mijn stage is de huidige code getest geweest en zijn er een aantal bugs gevonden die ik hieronder ook nog zal bespreken. Alsook heb ik de laatste dagen van mijn stage gespendeerd aan het in orde brengen van alle 10 controllers ook hierbij zal ik uitleggen hoe ik dit proces zo efficiënt mogelijk heb kunnen doen.

De eerste bug die gevonden was, was ook meteen een redelijk grote en belangrijke. Namelijk dat als je kiest voor het opnemen van een tekening het onmogelijk is om een bestaande tekening te tekenen. Dit bleek uiteindelijk te liggen aan een limitering in de code in verband met het openen en het sluiten van bestanden. Het probleem was dat ik eerst moest checken of er al een bestand open was en aangezien er dus al een bestand open was gaf deze een OS-error. Toen ik dit besproken had met mijn supervisor zijn we eigenlijk tot de conclusie gekomen dat het overbodig was om dit op te nemen. Dus heb ik een extra menu scherm toegevoegd dat vraagt of je de huidige opname wil stopzetten alvorens je verder kan gaan om een bestaande tekening te laten tekenen. Dit heeft het probleem op een elegante manier opgelost en zorgt er ook voor dat enkel de nuttige dingen in het programma blijven.

De volgende bug die er was, was er een die in het begin heel raar was omdat er aan de code van deze feature geen veranderingen meer waren gemaakt sinds de implementering en bij de implementering werkte deze zonder problemen. Namelijk het kunnen tekenen van een USB-opslag apparaat. Maar na wat brainstormen en overleggen met collega's over wat het probleem zou kunnen zijn, zijn we erachter gekomen dat de het de nieuwe OS-versie was waarbij er een verandering is gemaakt in het dynamisch pad naar een USB-opslag apparaat waardoor het bestaande pad in de code dus niet meer werkte. Nadat ik dit had veranderd naar het nieuwe pad werkte ook dit terug zonder enige problemen.

Nadat alle bugs opgelost waren heb ik van mijn manager de taak gekregen om deze code nu op alle 10 controllers te krijgen. Om dit te doen moest ik dus gaan uitzoeken wat de beste manier hiervoor was. Want ik wilde dit zo goed mogelijk documenteren zodat er in de toekomst geen problemen waren en zodat er eigenlijk ook geen verschil is tussen het in orde brengen van 10, 100 of zelfs 1000 controllers.

De eerste optie die ik heb overwogen was om ze allemaal vanaf het begin te maken zoals ik de eerste controller heb gedaan. Dus een nieuwe SD-kaart nemen een partitie maken vervolgens RaspbianOS installeren daarna ervoor zorgen dat alles geïnstalleerd geraakt qua code en dan uittesten dat alles werkt. Dit is geen slechte optie maar aangezien dat dit toch een bepaald niveau van technische en informatica kennis vereist dankzij de noodzaak om toch een 10-tal commando's uit te voeren. Ook niet de definitieve oplossing. Voor de uiteindelijke oplossing heb ik een stap teruggenomen naar hoe het was toen ik begon maar in plaats van het creëren van een volledige

image om deze op een shared drive te kunnen zetten. Ben ik na toch wat opzoekwerk gestuit op GPARTED. GPARTED geeft de mogelijkheid om de partitie van een SD kaart te verkleinen en aangezien ik aan het werken was op SD-kaarten van 128GB leek mij dit wel een vereiste. Want wat ik ook online had gevonden was namelijk dat bij de nieuwe versie van RaspbianOS ook een programma zat genaamd “SD Card Copier”. Wat dus perfect was wat we nodig hadden. Maar omdat niet elke SD-kaart 128GB is wilde ik de partitie eerst krimpen zodat het mogelijk was om de code op een veel kleinere SD-kaart te krijgen. Dankzij deze 2 programma’s is dit dus ook gelukt en heb ik uiteindelijk gewoon door de volledige partitie te klonen de code werkend gekregen op een SD-kaart van 16GB.

Figuur 16

Conclusie

Dit project was zeer productief en mijn contributies worden zeker geapprecieerd. De toevoegingen die ik heb gemaakt maken het product aangenamer om mee te werken en zorgen er ook voor dat het hopelijk langer meegaat.

Toen ik vertrok uit Canada werd de eerste van de 10 controllers met mijn code, volledig klaargemaakt om op te sturen naar een eerste eindgebruiker wat natuurlijk een heel aangenaam en goed gevoel is. Ik ben zeer trots met het werk dat ik bij dit project heb gedaan.

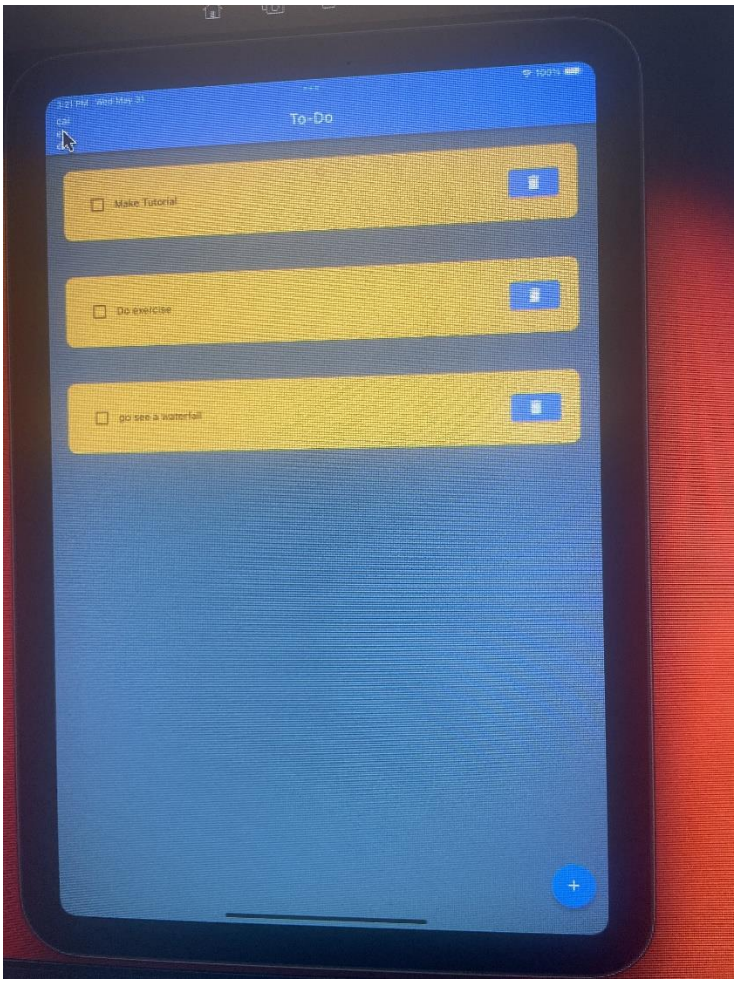
Want hoewel ik een aantal features heb toegevoegd ben ik ook zeer tevreden met het werk dat misschien iets minder op de voorgrond staat. Namelijk de OS-update maar ook het opschonen en uniform maken van de code.

Figuur 17

CanPlan

Inleiding.

CanPlan is een softwaretoepassing waarmee gebruikers met cognitieve beperkingen vrijwel elke taak kunnen opsplitsen in een reeks gemakkelijk te volgen foto's. De gebruiker doorloopt de activiteit samen met een begeleider of familielid. Samen maken ze foto's van elke stap in een taak (met of zonder tekst) en slaan deze op onder een bepaalde categorie in het programma. Naast deze volgordefunctie bevat de app ook een planner om ervoor te zorgen dat elke activiteit op tijd wordt afgerond.

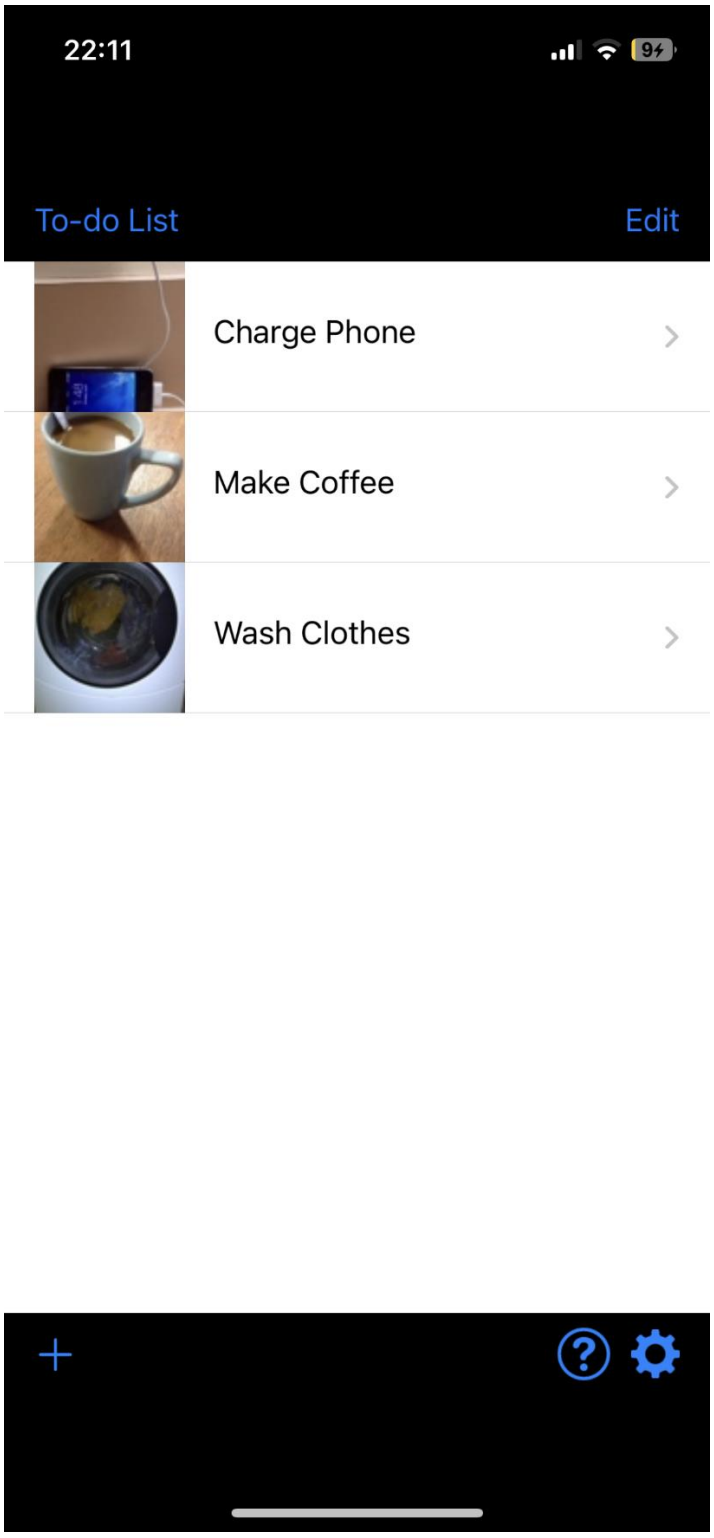


Figuur 18

Achtergrondinformatie

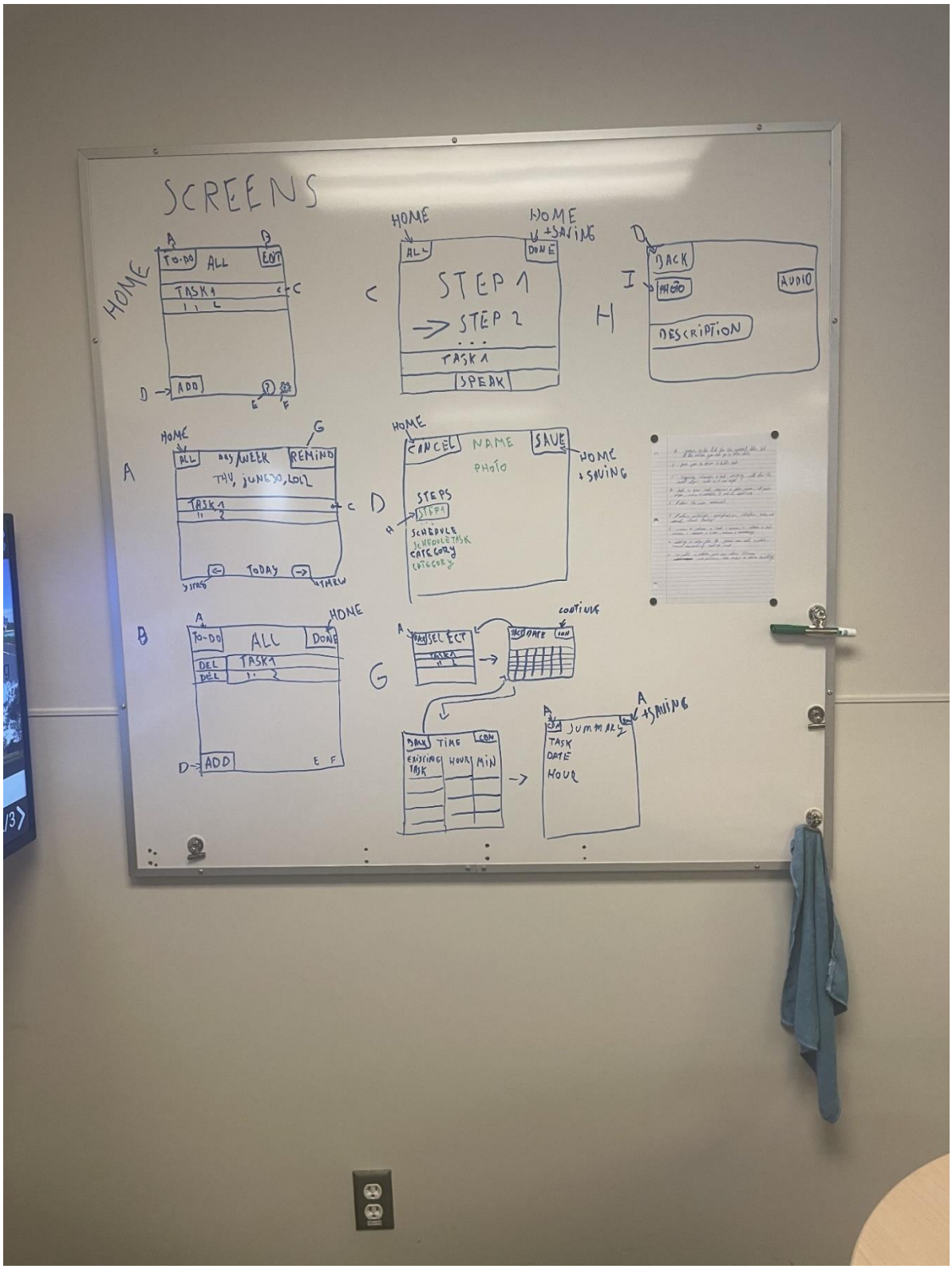
Zoals vermeld is deze applicatie dus vooral bedoelt voor mensen die een begeleider nodig hebben voor heel wat taken. Deze applicatie laat deze mensen dus samen met hun begeleider een volledig stappenplan maken voor heel wat taken zodat zij deze zelf ook kunnen uitvoeren.

Voordat ik begon bij CanAssist was er een bestaande applicatie die zeer verouderd was. Namelijk een applicatie die nog volledig geschreven is in Swift en dus ook enkel op Apple apparaten werkt. Deze was ook nog gebouwd voor IOS 9 dus van 2016. Er is sinds dan ook niks meer veranderd aan de code van deze applicatie. Hierdoor is de code en een aantal functies deprecated en zijn er heel wat dingen die niet meer zo goed werken en ziet er als volgt uit.



Figuur 19

Alsook krijgt CanAssist nog bijna dagelijks vragen in verband met deze applicatie. Dit kan gaan van vragen om bugs op te lossen tot het vragen voor nieuwe functies toe te voegen. Dit toont aan dat er een markt is voor deze applicatie en dat deze toch relatief populair is. Dankzij deze redenen hebben ze aan mij dus gevraagd om een nieuwe applicatie te maken.


SCREENS
HOME
TO-DO
ALL
DONE
TASK 1
ADD
STEP 1
STEP 2
SPEAK
BACK
PHOTO
AUDIO
DESCRIPTION
CANCEL
NAME
SAVE
PHOTO
STEPS
SCHEDULE
CATEGORY
DAY/WEEK
REMIND
THU, JUNE 30, 2022
TODAY
DEL
TIME
HOUR
MIN
SUMMARY
TASK
DATE
HOUR

Figuur 20

De foto op de vorige pagina was de schermanalyse die ik heb gemaakt van de bestaande applicatie. Alsook heb ik elke knop op deze pagina's geanalyseerd en in kaart gebracht. De reden dat ik dit zo heb gedaan is omdat er mij is gevraagd om de nieuwe applicatie zo veel mogelijk op de oude te doen lijken. Hierdoor leek het mij nuttiger om een schermanalyse te maken in plaats van een eisenanalyse.

De volgende analyse die ik heb gemaakt was om de keuze van een framework te bepalen. Eerst heb ik een meeting gehad met Paul mijn manager om te vragen of hij een voorkeur had van framework. Hij had geen voorkeur en liet de keuze aan mij, maar hij vroeg mij wel om nadat ik wat opzoekingen heb gedaan met hem te overlopen waarom ik mijn keuze heb gemaakt.

Dus in mijn opzoekingen heb ik rekening gehouden met een aantal factoren. De belangrijkste factoren waren ten eerste dat het een cross-platform framework is. Dit zorgt ervoor dat de app door meer mensen gebruikt kan worden. Ten tweede dat het een relatief gemakkelijk framework is om mee te werken en ten derde dat het een modern en populair framework dat naar de toekomst toe nog heel wat updates zal krijgen.

Er waren dus nog een aantal opties. Namelijk:

- Optie 1: Flutter

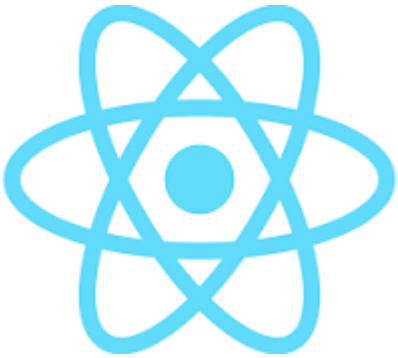
Figuur 21

Flutter is een open-source UI (User Interface) softwareontwikkelingskit (SDK) gemaakt door Google. Het stelt ontwikkelaars in staat om native gecompileerde applicaties te bouwen voor mobiele, web- en desktopplatforms vanuit één codebase. Flutter maakt gebruik van de Dart-programmeertaal, die ook door Google is ontwikkeld.

Het belangrijkste concept achter Flutter is het idee van "write once, run anywhere" (schrijf een keer, voer overal uit). Met Flutter kun je je applicatiecode eenmaal schrijven en deze implementeren op meerdere platforms, zoals Android, iOS, webbrowsers en zelfs desktop-besturingssystemen zoals Windows, macOS en Linux.

In het kort is dit dus een zeer goede fit voor dit project.

- Optie 2: React Native



Figuur 22

React Native is een open-source framework voor het ontwikkelen van mobiele applicaties. Het is gebaseerd op de populaire JavaScript-bibliotheek React, die oorspronkelijk is ontwikkeld door Facebook. Met React Native kunnen ontwikkelaars native mobiele apps bouwen met behulp van JavaScript en React, waarbij ze gebruik kunnen maken van gedeelde codebases en herbruikbare componenten.

Ook dit is dus een goede fit voor het project.

- Mijn keuze: Flutter

Hoewel het allebei goede keuzes zijn voor dit project aangezien het allebei framework opties zijn van grote Tech bedrijven wat er voor zorgt dat de documentatie in orde is en ze dus ook normaal gezien lange support zouden moeten krijgen. Dus kwam mijn keuze neer op de gebruikte programmeertaal.

Het was dus een keuze tussen Dart en Javascript. Na nog wat meer opzoekingen heb ik gevonden dat Dart gemakkelijker zou zijn dan Javascript om mee te werken maar belangrijker ook gemakkelijker is om te leren. Dit was dus de doorslaggevende factor.

Hoewel ik voorkennis had met React Native heb ik dus toch besloten om voor flutter te gaan. Omdat het voor dit project en voor CanAssist de betere keuze was.

Gebruikte technologieën

In dit gedeelte van het document zal ik wat meer uitleg geven over de vermeldde technologieën. Meer specifiek zal ik uitleggen wat de technologie is waarvoor ze allemaal kan gebruikt worden, op welke manier ze gebruikt kan worden en de manier en reden waarvoor ik ze gebruikt heb.

Flutter

Flutter is een open-source UI (User Interface) softwareontwikkelingskit (SDK) ontwikkeld door Google. Het stelt ontwikkelaars in staat om native gecompileerde applicaties te bouwen voor verschillende platforms, zoals Android, iOS, web en desktop, met behulp van een enkele codebase.

Flutter maakt gebruik van een speciale programmeertaal genaamd Dart, die is ontworpen om snel en efficiënt te zijn. Je schrijft code in Dart om de gebruikersinterface van je app te definiëren, inclusief knoppen, tekstvelden, afbeeldingen en meer. Maar het coole is dat Flutter niet alleen maar standaarduitziende knoppen en tekstvelden biedt. Je kunt ze volledig aanpassen en ze er precies zo laten uitzien als je wilt, zodat je app er uniek uitziet.

Een ander geweldig aspect van Flutter is de "hot reload"-functie. Dit betekent dat je wijzigingen kunt aanbrengen in je code en deze onmiddellijk kunt zien in je app terwijl deze draait. Het bespaart enorm veel tijd, omdat je niet telkens de app hoeft te herstarten om je wijzigingen te zien. Je kunt experimenteren, problemen oplossen en itereren op je code zonder oponthoud.

Flutter-apps zijn ook bekend om hun uitstekende prestaties. Ze voelen snel en soepel aan, met vloeiende animaties en snelle reactietijden. Dit komt doordat Flutter-apps worden gecompileerd naar native machinecode, wat betekent dat ze rechtstreeks worden uitgevoerd op het apparaat, zonder tussenliggende lagen die de prestaties kunnen vertragen.

Met Flutter heb je ook toegang tot alle functies en mogelijkheden van het apparaat waarop je app draait. Of het nu gaat om toegang tot de camera, sensoren, GPS of andere native functies, Flutter biedt API's en bibliotheken om je te helpen deze functionaliteiten in je app te integreren.

Kortom, Flutter maakt het gemakkelijk om snel, efficiënt en aantrekkelijke apps te bouwen voor zowel Android als iOS, met één enkele codebase. Het biedt krachtige aanpassingsmogelijkheden, uitstekende prestaties en de mogelijkheid om toegang te krijgen tot native functies, waardoor ontwikkelaars de vrijheid hebben om geweldige mobiele apps te maken.

Flutter was dus het framework dat ik heb gebruikt om de applicatie in te maken.

Dart

Figuur 23

Dart is een moderne programmeertaal die is ontwikkeld door Google. Het is de programmeertaal die wordt gebruikt bij het ontwikkelen van Flutter-applicaties, maar kan ook worden gebruikt voor het schrijven van stand-alone applicaties, webtoepassingen en serverside backend-systemen.

Hier zijn enkele belangrijke kenmerken en concepten van Dart:

Objectgeoriënteerd: Dart is een objectgeoriënteerde taal, wat betekent dat alles in Dart als een object wordt behandeld. Het ondersteunt klassen, objecten, overerving, interfaces en andere objectgerelateerde concepten. Hierdoor kunnen ontwikkelaars code schrijven die goed georganiseerd, herbruikbaar en onderhoudbaar is.

Optionele typen: Dart biedt optionele typen, wat betekent dat ontwikkelaars de keuze hebben om variabelen te voorzien van expliciete typen of het type te laten infereren op basis van de toegewezen waarde. Dit geeft ontwikkelaars de flexibiliteit om te werken met statisch getypeerde code voor betere foutdetectie tijdens de compilatiefase, of om te werken met dynamisch getypeerde code voor meer flexibiliteit en snellere ontwikkeling.

Asynchrone programmering: Dart heeft native ondersteuning voor asynchrone programmering, waardoor het gemakkelijk is om taken uit te voeren die mogelijk veel tijd kosten, zoals netwerkverzoeken, bestandstoegang en databasebewerkingen. Met behulp van het async/await-patroon kunnen ontwikkelaars niet-blokkerende en responsieve code schrijven, waardoor apps soepel blijven draaien, zelfs tijdens langdurige operaties.

Garbage collection: Dart maakt gebruik van automatische geheugenopruiming door middel van garbage collection. Dit betekent dat ontwikkelaars zich geen zorgen hoeven te maken over handmatig geheugenbeheer, omdat Dart automatisch geheugen vrijmaakt van objecten die niet meer worden gebruikt, waardoor het risico op geheugenlekken wordt verminderd.

Bibliotheken en packages: Dart wordt geleverd met een uitgebreide standaardbibliotheek die een breed scala aan functionaliteiten biedt, zoals bestandstoegang, JSON-serialisatie, HTTP-verzoeken, concurrency en meer. Bovendien heeft Dart een actieve gemeenschap en een uitgebreid ecosysteem van packages via het pub.dev-platform, waar ontwikkelaars duizenden

open-sourcepackages kunnen vinden en gebruiken om functionaliteit toe te voegen aan hun Dart- en Flutter-projecten.

Tooling: Dart wordt geleverd met een set krachtige ontwikkelingstools die de productiviteit van ontwikkelaars verhogen. De Dart SDK wordt geleverd met een interactieve shell (dartrepl) en een opdrachtregelinterface (dart), waarmee ontwikkelaars Dart-code kunnen uitvoeren en beheren. Daarnaast zijn er uitstekende code-editors en IDE's, zoals Visual Studio Code en IntelliJ IDEA, die uitgebreide ondersteuning bieden voor Dart, zoals code-aanvulling, foutopsporing en refactorings.

In het algemeen is Dart een veelzijdige en moderne programmeertaal met functies en concepten die het ontwikkelaars mogelijk maken om efficiënte, schaalbare en betrouwbare code te schrijven. Het wordt veel gebruikt in combinatie met Flutter, maar kan ook zelfstandig worden gebruikt voor verschillende soorten applicaties

Hive

Hive is een snelle en lichtgewicht databasebibliotheek voor het opslaan en ophalen van gegevens in Flutter- en Dart-applicaties. Het is ontworpen om een eenvoudige en efficiënte oplossing te bieden voor het persistent opslaan van gegevens, zoals configuratie-instellingen, gebruikersvoorkeuren, cachegegevens en meer.

Hier zijn enkele belangrijke aspecten en kenmerken van Hive:

NoSQL-database: Hive is een NoSQL-databasebibliotheek, wat betekent dat het geen traditionele relationele database is. In plaats daarvan slaat Hive gegevens op in binaire bestanden op de schijf. Dit maakt het snel en efficiënt, omdat er geen overhead is van complexe query's en relationele schema's.

Key-Value-paaropslag: Hive maakt gebruik van een key-value-paarbenadering voor gegevensopslag. Elke waarde in Hive wordt geïdentificeerd door een unieke sleutel. Dit maakt het eenvoudig om gegevens op te halen en bij te werken door simpelweg de juiste sleutel te gebruiken.

Typeveilige gegevensopslag: Hive biedt typeveilige gegevensopslag, wat betekent dat je specifieke gegevenstypen kunt definiëren voor je waarden. Hierdoor worden typemismatch-fouten tijdens compileertijd gedetecteerd en kun je de gegevenstypen gemakkelijk controleren bij het ophalen van gegevens.

Snelle prestaties: Hive is ontworpen om snel en efficiënt te zijn. Het maakt gebruik van geheugencaching en minimaliseert I/O-operaties om de prestaties te maximaliseren. Hive kan ook asynchrone bewerkingen uitvoeren om de responsiviteit van de app te verbeteren.

Pluggable backend: Hive biedt de mogelijkheid om verschillende backends te gebruiken om gegevens op te slaan. Standaard wordt Hive geleverd met een backend die gegevens opslaat in bestanden op de schijf, maar je kunt ook andere backends, zoals Hive in-memory, Hive op een externe server of zelfs Hive met SQL-ondersteuning, gebruiken door plug-ins te gebruiken.

Eenvoudig te gebruiken: Hive is ontworpen met eenvoud in gedachten. Het heeft een eenvoudige API en minimale configuratie, waardoor het gemakkelijk is om te integreren in je Flutter- of Dart-projecten. Je kunt gegevens snel opslaan, ophalen, bijwerken en verwijderen met slechts een paar regels code.

Kortom, Hive is een lichtgewicht en eenvoudige databasebibliotheek voor het opslaan en ophalen van gegevens in Flutter- en Dart-applicaties. Het biedt een snelle en efficiënte oplossing voor het persistent opslaan van gegevens met een key-value-paarbenadering en typeveilige gegevensopslag. Met Hive kun je gemakkelijk gegevens beheren zonder de complexiteit van een traditionele relationele database.

Xcode

Xcode is een geïntegreerde ontwikkelomgeving (IDE) die speciaal is ontworpen voor het ontwikkelen van applicaties voor Apple-platforms, zoals iOS, macOS, watchOS en tvOS. Het is ontwikkeld door Apple en biedt ontwikkelaars een complete set tools en functies om hoogwaardige apps te bouwen.

Hier zijn enkele belangrijke aspecten en functies van Xcode:

Interface Builder: Xcode bevat een krachtige tool genaamd Interface Builder, waarmee ontwikkelaars de gebruikersinterface (UI) van hun apps visueel kunnen ontwerpen. Met behulp van een drag-and-drop-interface kunnen ontwikkelaars UI-elementen, zoals knoppen, tekstvelden, afbeeldingen en meer, op het scherm plaatsen en hun lay-out en gedrag definiëren. Dit maakt het gemakkelijk voor ontwikkelaars om aantrekkelijke en responsieve apps te bouwen.

Code-editor: Xcode bevat een geavanceerde code-editor met functies zoals syntax highlighting, code-aanvulling, automatische correctie en intelligente foutopsporing. Ontwikkelaars kunnen hiermee snel en efficiënt code schrijven, bewerken en debuggen. De code-editor ondersteunt meerdere programmeertalen, waaronder Swift en Objective-C.

Simulator: Xcode wordt geleverd met een simulator die het mogelijk maakt om iOS- en macOS-apps te testen zonder dat je een fysiek apparaat nodig hebt. Met de simulator kun je je app draaien en testen in verschillende omgevingen en apparaatconfiguraties, waardoor je snel problemen kunt opsporen en controleren hoe je app eruitziet en werkt op verschillende apparaten.

Ingebouwde tools en hulpprogramma's: Xcode biedt een breed scala aan ingebouwde tools en hulpprogramma's om ontwikkelaars te helpen bij het ontwikkelen, testen, debuggen en implementeren van hun apps. Enkele voorbeelden hiervan zijn de Interface Builder, Debugger, Instruments (voor het analyseren van prestaties), Core Data Editor (voor het beheren van databases) en Asset Catalog (voor het beheren van app-afbeeldingen en bronnen).

App-distributie: Xcode biedt hulpmiddelen en mogelijkheden voor het distribueren van je app naar de App Store. Ontwikkelaars kunnen met Xcode de app bundelen, ondertekenen en verzenden naar Apple voor goedkeuring en publicatie in de App Store. Het biedt ook mogelijkheden voor app-testen en distributie binnen een ontwikkelteam.

Kortom, Xcode is een krachtige IDE die specifiek is ontworpen voor het ontwikkelen van apps voor Apple-platforms. Het biedt ontwikkelaars de tools en functies die ze nodig hebben om de UI te ontwerpen, code te schrijven, apps te testen en te distribueren. Met Xcode kunnen ontwikkelaars hoogwaardige apps bouwen die naadloos integreren met het ecosysteem van Apple-apparaten.

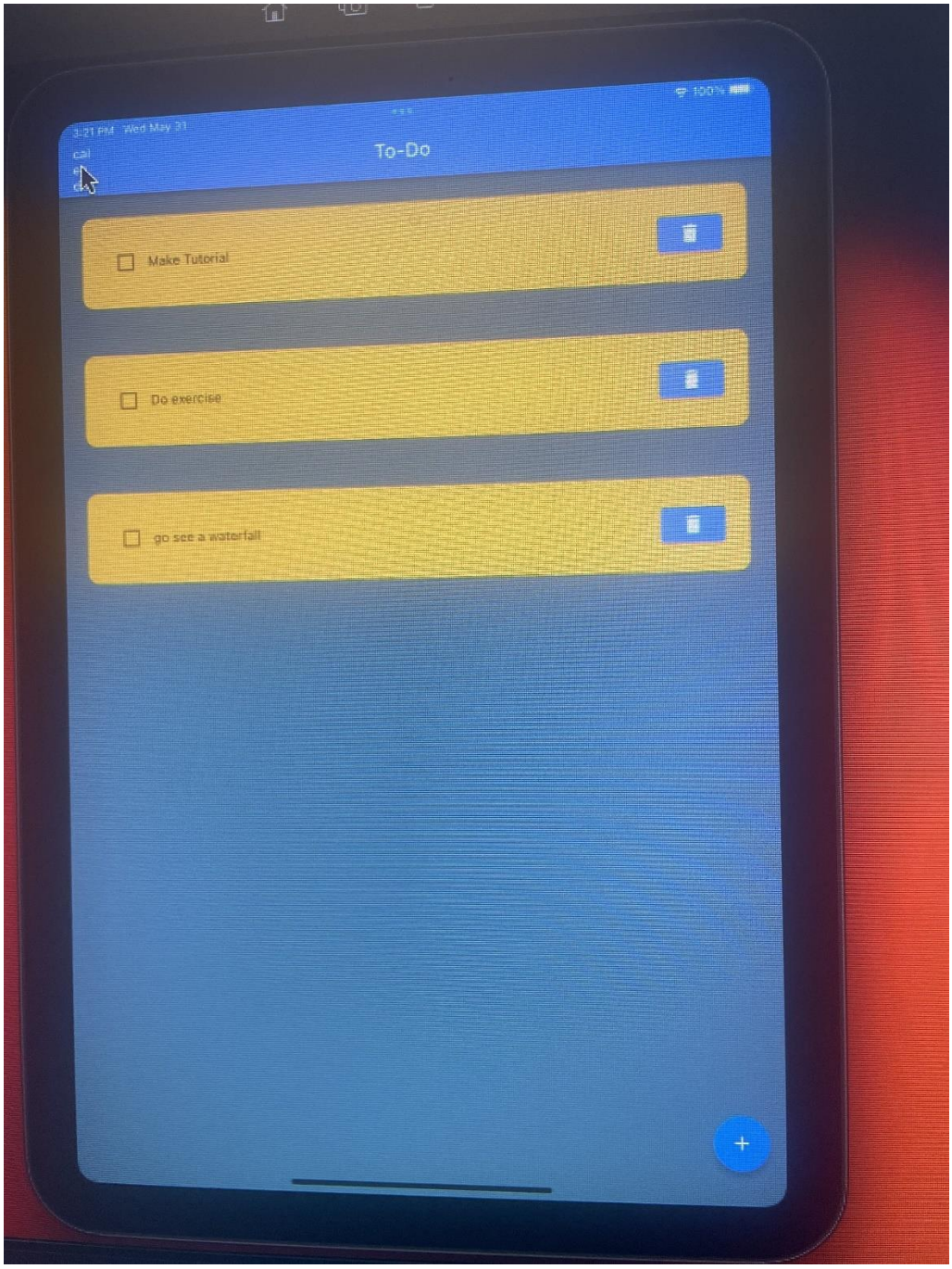
Mijn contributies

Dit project was ook een individueel project aangezien ik de enige softwareontwikkelaar was bij CanAssist. Dit heeft er dus wel voor gezorgd dat ik Flutter redelijk goed heb leren kennen.

Voor bepaalde beslissingen heb ik natuurlijk wel goedkeuring moeten vragen aan mijn manager. Maar ook voor deze beslissingen kreeg ik bijna altijd vrije keuze zolang dat ik mijn keuze kon verklaren en argumenten kon voorleggen voor waarom dit de betere optie was.

Op de volgende pagina's zal ik dus door elke pagina van de applicatie gaan en bespreken wat er juist gebeurt per pagina.

Dus laat ons beginnen met de homepagina, deze ziet eruit als volgt:



Figuur 24

Dit is dus ook de pagina waarmee ik ben begonnen. Wat ik dus eerst moest uitzoeken was hoe widgets werken in Flutter. Dit was op zich niet te moeilijk maar wel even een aanpassing van het python project.

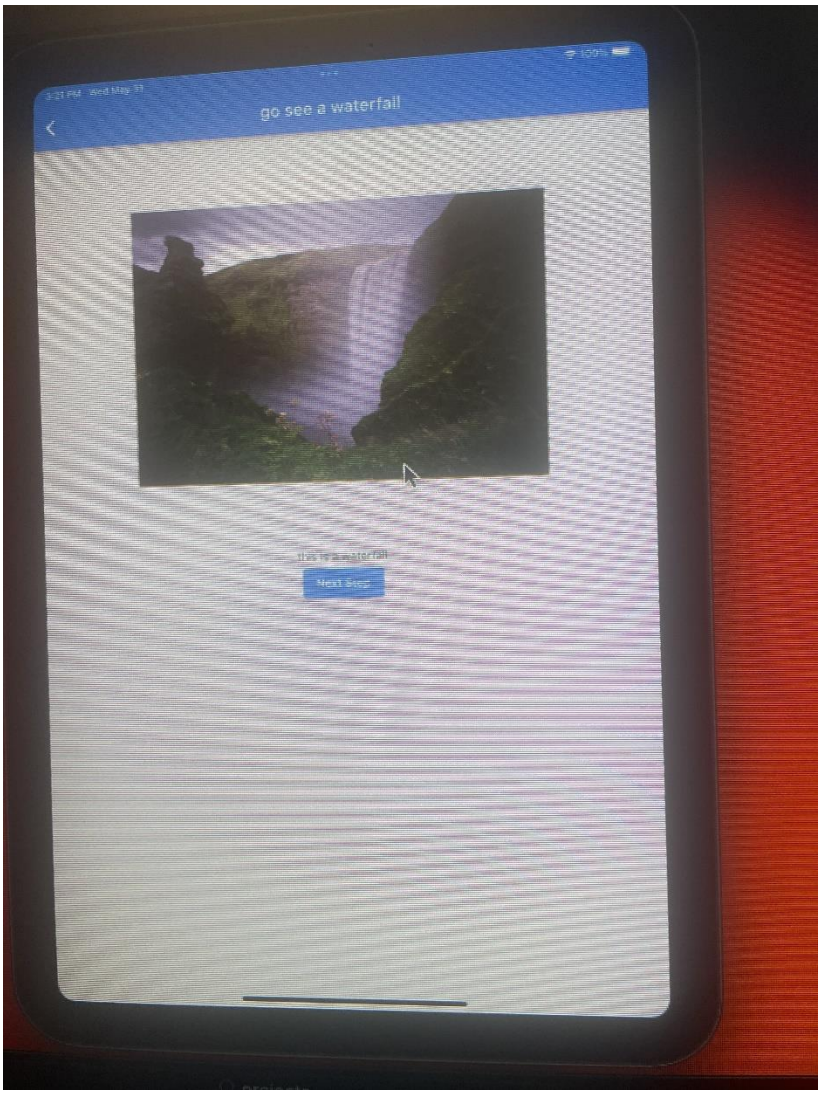
In de linkerbovenhoek staat een knop naar de kalenderpagina, rechts onderaan staat een knop die naar de pagina om een nieuwe taak toe te voegen.

Het belangrijkste op deze pagina is een lijst van alle taken. Daarmee ben ik dan ook begonnen. Eerst ben ik begonnen met zelf een lijst aan te maken in de code en deze hardcoded te tonen.

De volgende stap was om een manier van opslag te vinden. Dit heb ik dan gevonden in Hive.

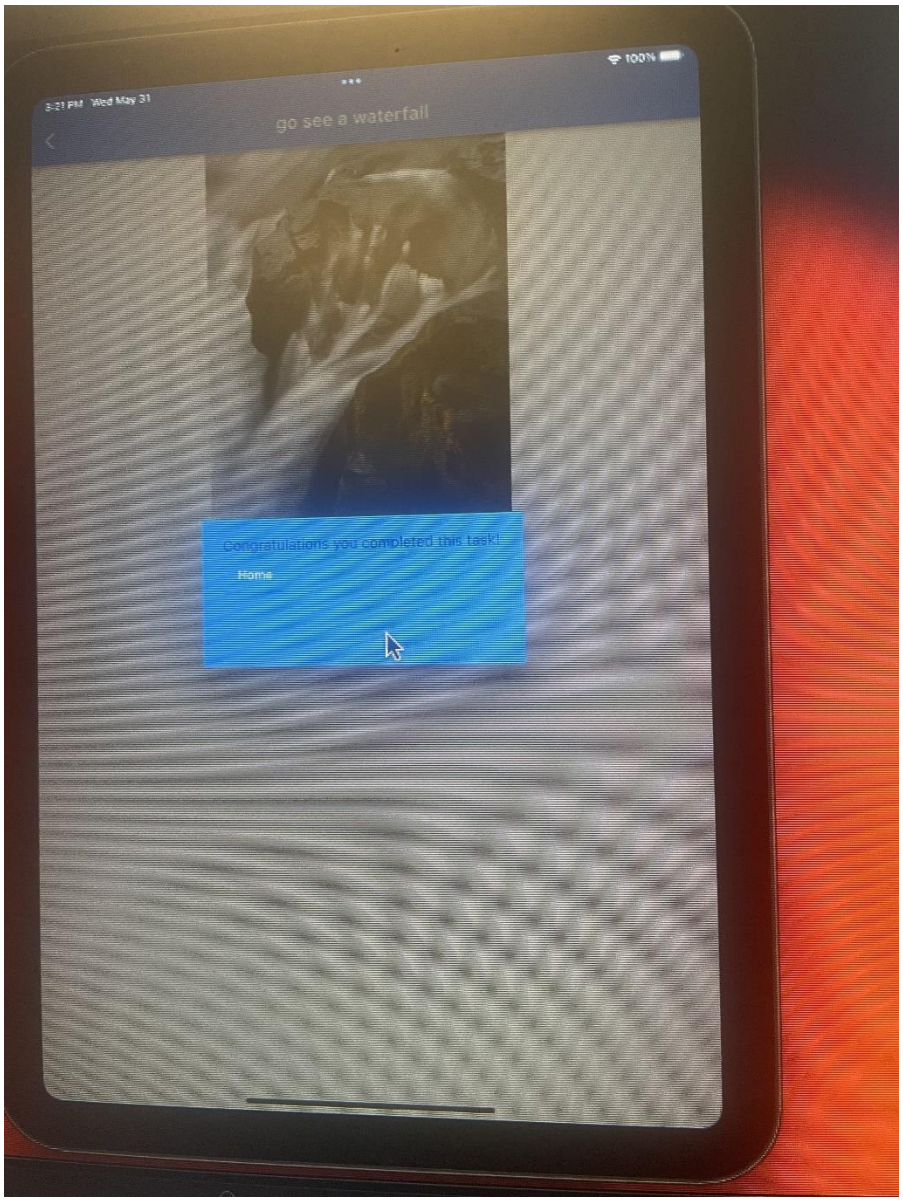
Om deze pagina werkende te krijgen ben ik op een aantal problemen gestoten. Het voornaamste probleem was vooral mijn kennis van Flutter en Hive die nog niet voldoende was. Dit zorgde op zijn beurt dan weer voor kleinere problemen zoals bijvoorbeeld syntax, type errors en andere fouten die je veel tegenkomt als beginner.

Deze pagina laat dus toe om een om taak af te vinken als klaar, om een taak te verwijderen als deze niet meer nodig is maar het belangrijkste is dat de balk van de taak een knop is om naar het stappenplan te gaan van een taak. Dit ziet eruit als volgt:



Figuur 25

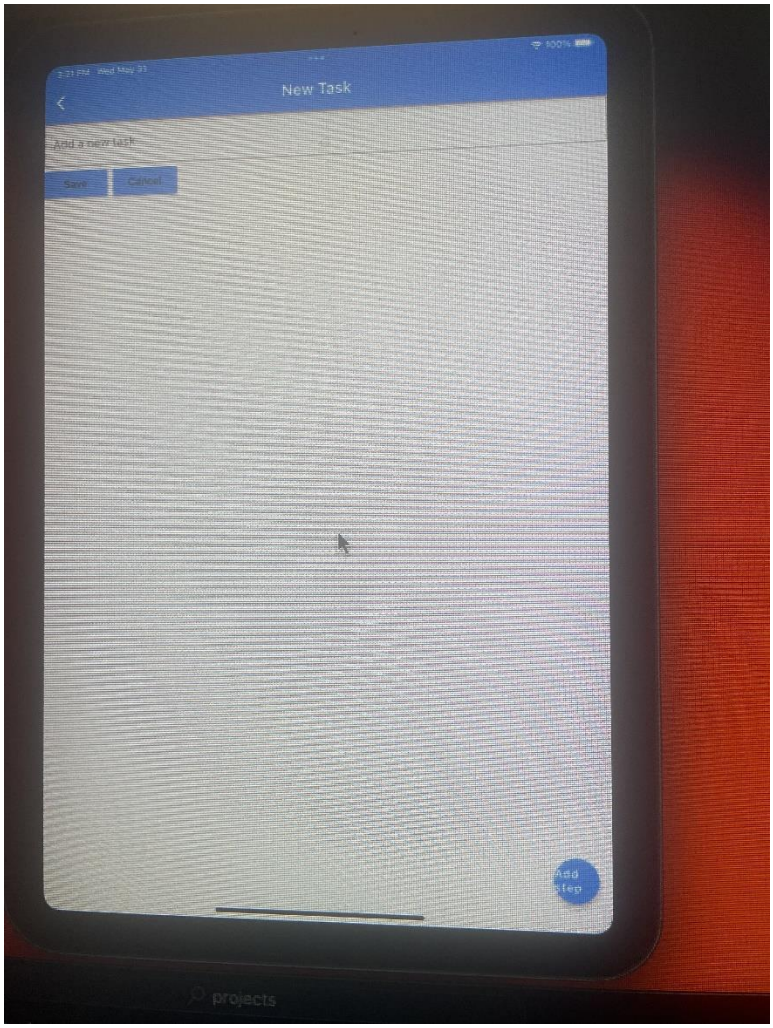
Deze pagina is dus redelijk simpel en heeft een enkele knop om naar de volgende stap te gaan en toont de bijhorende foto en tekst per stap. Als het de laatste stap is en je duwt op verder krijg je dus dit te zien.



Figuur 26

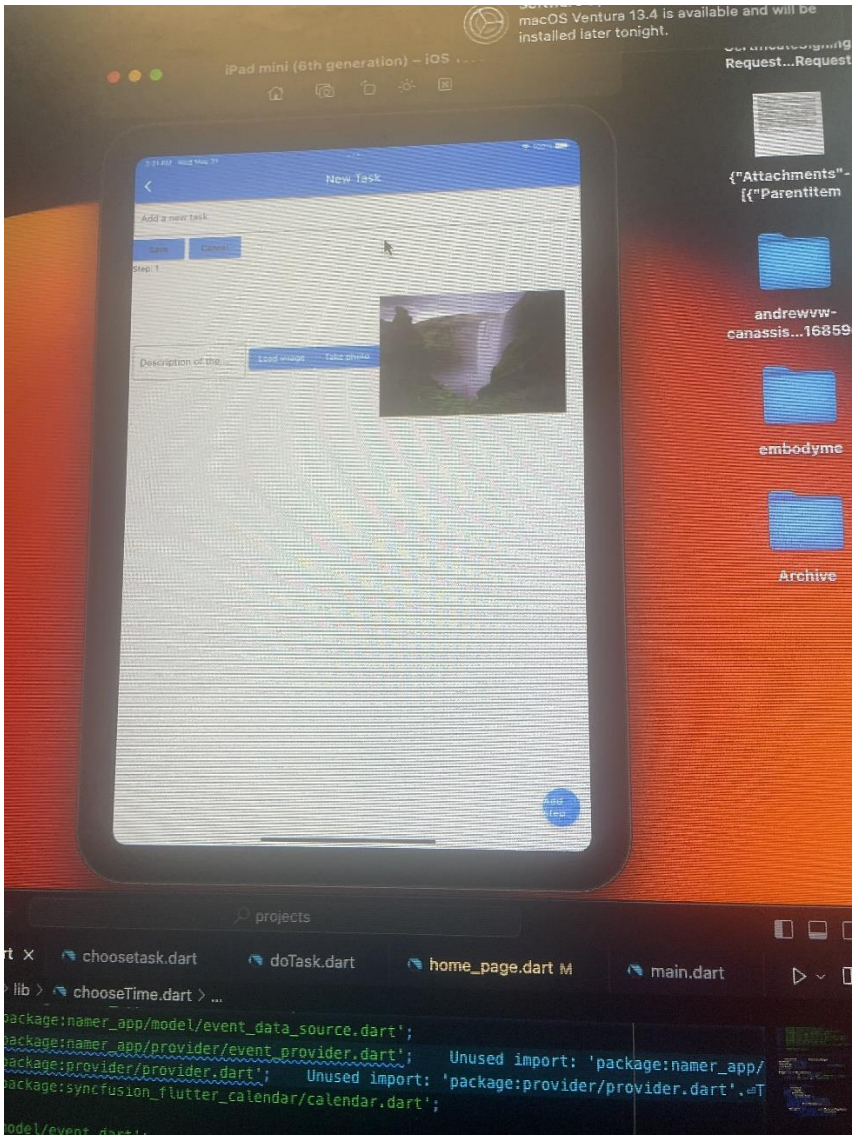
De pop-up heeft een knop om terug naar de homepagina te gaan.

De volgende pagina ziet u hieronder en toont de pagina om een nieuwe taak toe te voegen.



Figuur 27

Zoals u kan zien zijn er ook hier niet veel knoppen aanwezig. De enige knoppen die er wel staan zijn de “save” en “cancel” knop redelijk bovenaan de pagina. De knop rechts onderaan zorgt ervoor dat je een nieuwe stap dynamisch en ongelimiteerd kan toevoegen. Een stap toevoegen ziet er zo uit:

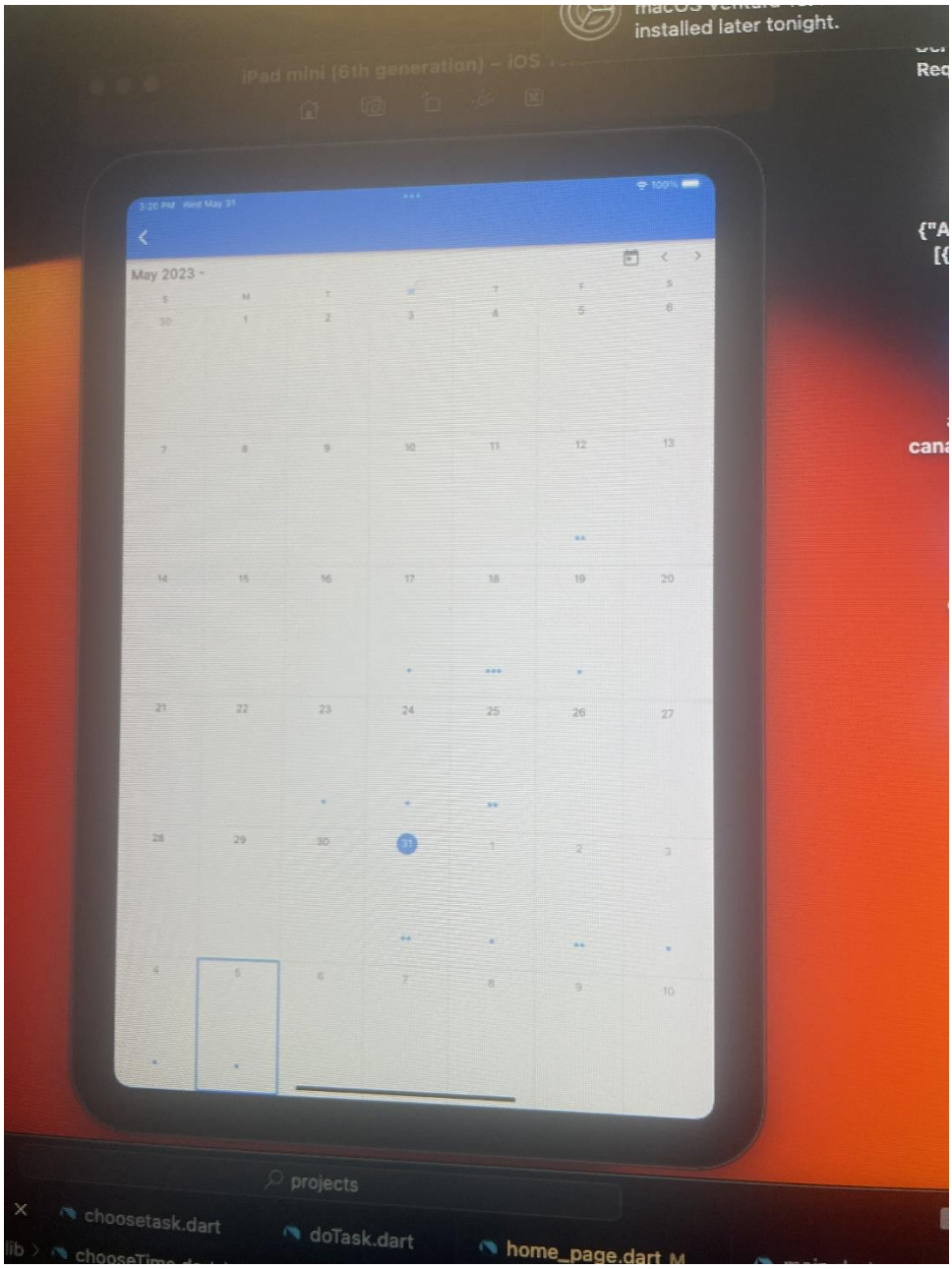


Figuur 28

Met plaats voor tekst en de keuze om een foto te kiezen uit de galerij of om een nieuwe foto te nemen. Bij dit stuk was de moeilijkheid vooral om de foto mee in de database te krijgen en om dit dynamisch te doen en dus ook met zogenaamde textcontrollers te werken. Ondertussen begon Flutter toch al iets beter te gaan en lukte deze pagina ook wat beter.

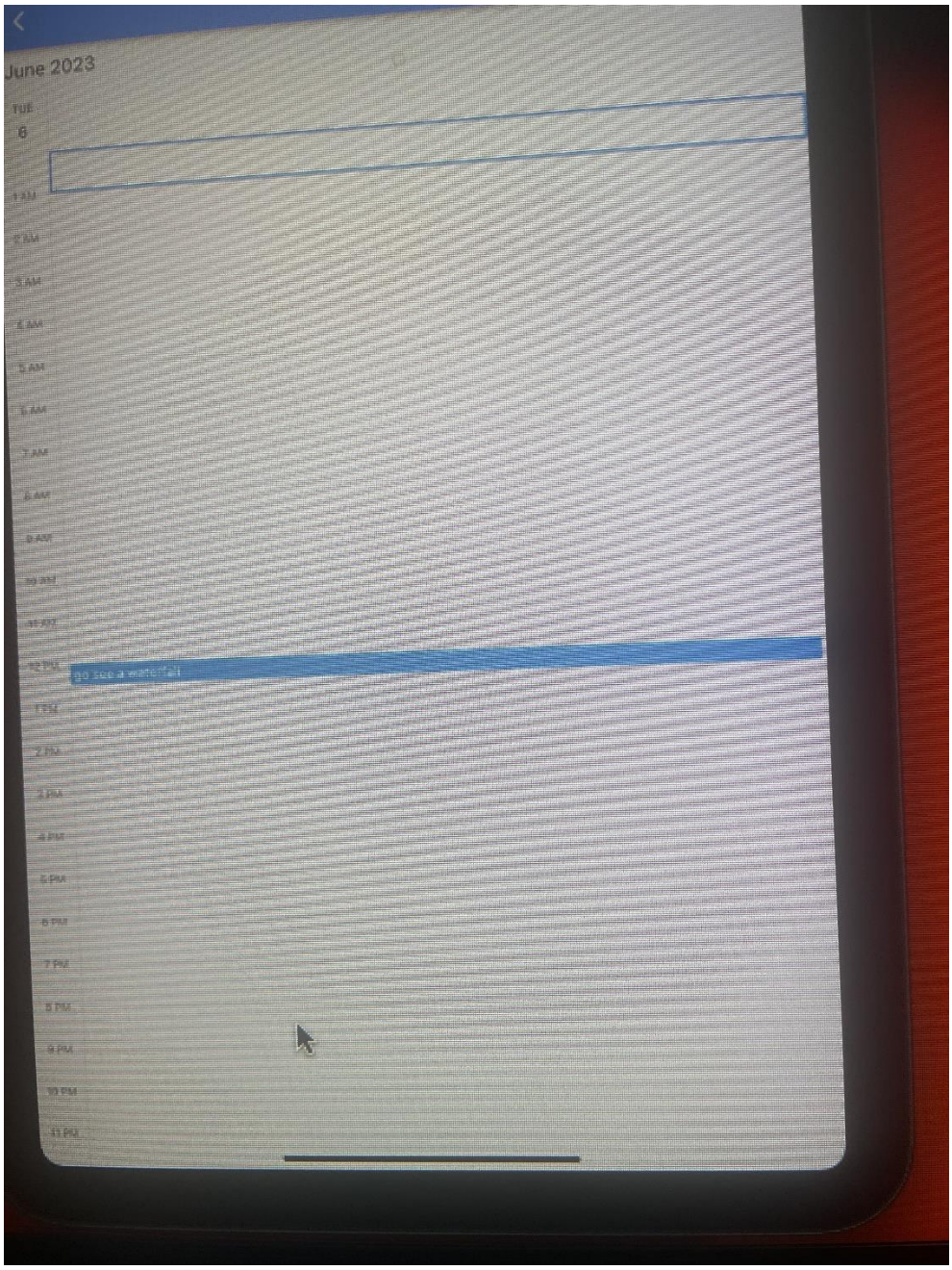
Deze pagina's waren alle functionaliteit rond de taak zelf. De volgende pagina's zijn functionaliteiten om de taak in te kunnen plannen in de tijd.

Eerst en vooral hebben we een maandoverzicht:



Figuur 29

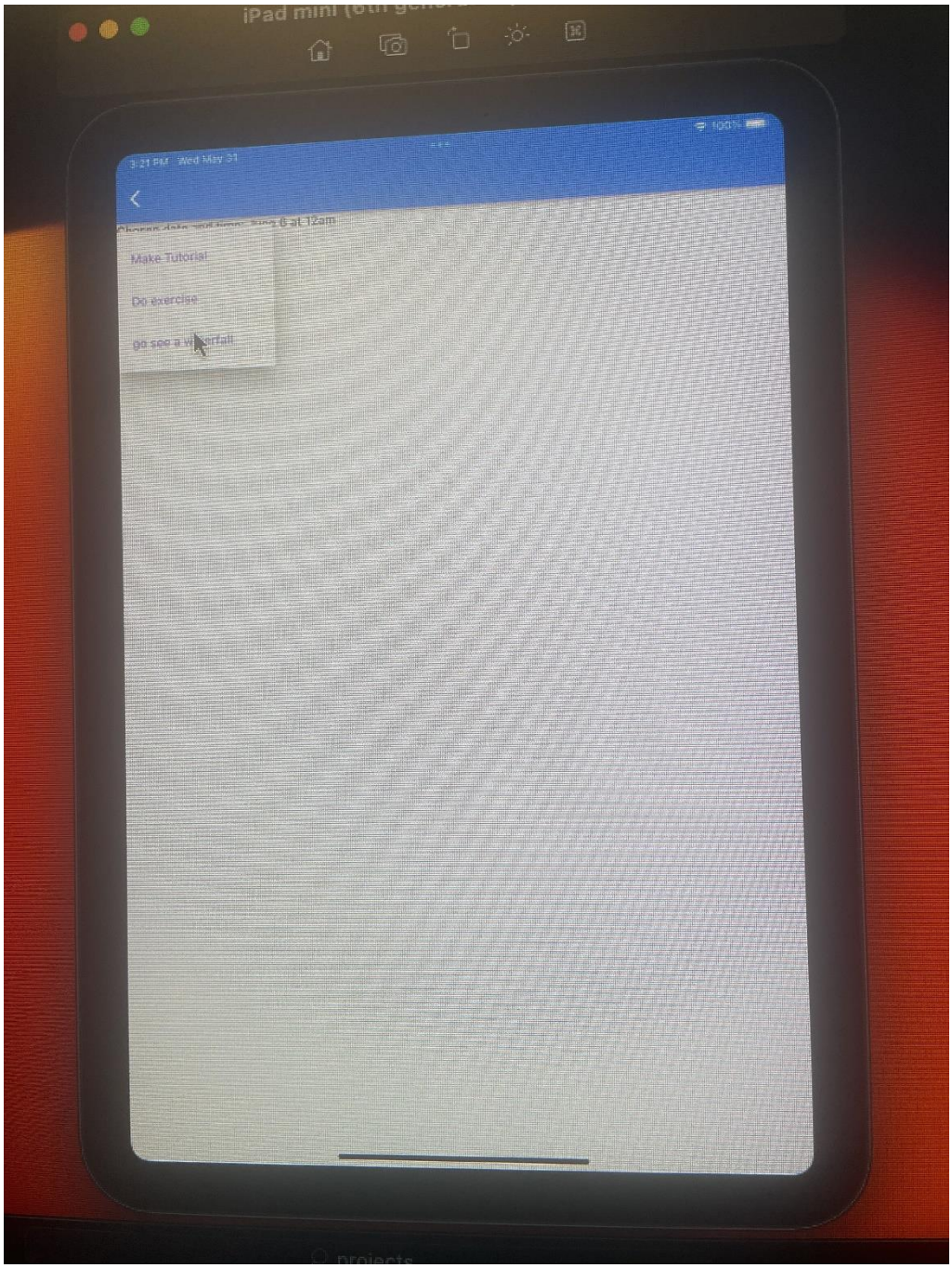
Als je hier op een van de dagen klikt ga je verder naar een dagoverzicht dat er als volgt uitziet:



Figuur 30

Hier kan je dus een uur kiezen om de taak in te plannen. Om naar de volgende pagina te gaan is het dan ook zo gemakkelijk als klikken op het uur.

Dan komen we op deze pagina:



Figuur 31

Op deze pagina wordt er automatisch een dropdown menu gegenereerd met alle taken die zijn aangemaakt. Dan is het zo gemakkelijk als de juiste taak kiezen en op opslaan klikken. Alsook kan je op deze pagina nog even nakijken of je de juiste datum en het juiste uur hebt aangeduid.

Volgende stappen

Ik vind het spijtig maar ik heb dit project niet volledig klaargekregen door een tekort aan tijd. Hoewel de applicatie van een puur functioneel standpunt in orde is, is er toch nog wat werk aan. Dus zal ik op deze pagina nog even kort overlopen wat er nog aan moet gebeuren.

Eerst en vooral zijn er een aantal bugs die ik heb gedocumenteerd maar waarvoor ik niet meer de tijd had om ze op te lossen.

Vervolgens is het best om de schermvolgorde in orde te brengen. Deze had ik voor het ontwikkelen iets anders gezet om het testen van bepaalde stukken vlotter te doen verlopen.

De laatste belangrijke stap is om de applicatie nog te stylen. Het valt redelijk hard op dat er aan de styling nog niets gebeurd is. Dit heeft een aantal redenen. De eerste en meest belangrijke reden is het tijdstekort, maar een 2de belangrijke reden is dat ik niet de meest creatieve persoon ben en dus niet de juiste persoon om deze app te stylen. Vooral omdat deze app duidelijk moet zijn voor mensen met bepaalde beperkingen vind ik dat er iemand met wat meer expertise de styling van deze applicatie voor zich zou moeten nemen.

Conclusie

In het algemeen ben ik tevreden met het werk dat ik heb kunnen doen aan dit project en denk ik dat ik op de tijd die ik had toch redelijk ver ben geraakt. Want op een 2 maanden tijd ben ik gegaan van geen kennis over Flutter naar het kunnen toevoegen van een kalender, database, camerafunctionaliteit, etc.

Natuurlijk vind ik het wel enorm spijtig dat ik het project niet volledig klaar heb gekregen maar de laatste weken dat ik bij CanAssist heb gewerkt was het AAI-project terug de prioriteit voor mijn manager Paul en dus ook voor mij.

Het project is ook gedocumenteerd zodat er een volgende ontwikkelaar aan kan verder werken. Nu ik terugkijk op dit project ben ik toch zeer tevreden met mijn keuze voor Flutter en Dart. Ook omdat het dus iets gemakkelijker is om mee te starten voor een volgende ontwikkelaar. Anderzijds omdat ik weer een volledig nieuwe technologie heb kunnen bijleren.

Nawoord

Het is met een zekere trots en voldoening dat ik dit document zou willen afronden. Eerst en vooral voor het werk dat ik heb kunnen leveren aan CanAssist en de mensen die zij dagdagelijks verder helpen. Ik hoop dat mijn contributies aan hun projecten ook het verschil kunnen maken voor een aantal van deze mensen. Maar ook heb ik een zekere trots voor het Erasmus avontuur dat ik succesvol afgerond heb.

Ook hier zou ik nog even een aantal mensen willen bedanken. Eerst en vooral mijn manager en supervisor bij CanAssist Paul Green, vervolgens mijn begeleider van ThomasMore Michiel Verboven. Voor hun hulp en begeleiding tijdens dit traject.

Dan wil ook nog even alle mensen van het CanAssist team willen bedanken voor de interessante 3 maanden en voor hun steun en begeleiding. Alsook zou ik hier nog Sarah Mcquillan en Tinne Van Echelpoel willen bedanken voor al hun hulp met het in goede banen leiden van mijn buitenlandse stage.

Om af te sluiten zou ik dit avontuur willen aanraden aan alle ThomasMore studenten die deze kans krijgen dit is voor het werk dat CanAssist doet, de mensen die er werken en het land waar ze gevestigd zijn. Dit zorgt voor een voor mij onvergetelijke combinatie en een onvergetelijke periode van mijn leven. Bedankt!

Bronnenlijst

CanAssist. (2023, maart 1). Opgehaald van CanAssist: https://www.canassist.ca/

ChatGPT. (2023, juni 1). Opgehaald van ChatGPT: https://chat.openai.com/

Dart documentatie. (2023, juni 1). Opgehaald van Dart documentatie: https://dart.dev/guides

Flutter documentatie. (2023, mei 1). Opgehaald van Flutter documentatie: https://docs.flutter.dev/

Github. (2023, juni 1). Opgehaald van Github: https://github.com

Hive documentatie. (2023, mei 1). Opgehaald van Hive documentatie: https://docs.hivedb.dev/#/

raspberry pi documentatie. (2023, mei 1). Opgehaald van raspberry pi documentatie: https://www.raspberrypi.com/documentation/

Stack overflow. (2023, juni 1). Opgehaald van Stack overflow: https://stackoverflow.com