

Project Report



**well NETWORKED
top CONNECTED**

business models • smart services • strategies • solutions

Yordi

MABS4.0 | DATUM: 17/11/2023

Contents

Introductie	3
StageBedrijf	3
Opdracht gever	3
Planning.....	4
Mogelijke implimentatie.....	4
De opdracht	4
Doel	4
Research Vragen	4
Hoofdvraag	4
Deelvragen.....	4
Aanpak.....	5
Test Aanpak	5
Beschrijving Process & Resultaat.....	5
Sprint 1	5
Samenvatting.....	5
Sprint 2	6
Samenvatting.....	6
Sprint 3	10
Samenvatting.....	10
Sprint 4	12
Samenvatting.....	12
Sprint 5	14
Samenvatting.....	14
Conclusie	16
Aanradingen	17
Move4Vitality App Verifiëren	17
Apple Health Implementatie	17
Persoonlijke reflectie	17
Producten	18
Mid Term Presentatie	18
Project Plan.....	18
Scrum Document.....	18
Research Report	18
Functioneel & Technisch Ontwerp	18

Introductie

StageBedrijf

Mabs4.0 is een klantgericht en innovatief next gen ICT professional services bedrijf met de mogelijkheden en ambitie om simply the best / world class te zijn/worden.

Opdracht gever

Move4Vitality is een bedrijf actief in Nederland wat mensen, kennis, processen en slimme systemen samen brengt. Het biedt een platform voor fysiotherapeuten bestaande uit een aantal digitale tools en diensten. Een van de tools is het all-in-one digitaal beweegprogramma op maat voor mensen die in behandeling zijn voor bijvoorbeeld chronische klachten zoals COPD, etalagebenen (claudicatio), artrose of Long COVID. Maar ook voor mensen die hun levensstijl willen veranderen of na een operatie moeten realiseren om weer zo vitaal mogelijk te worden of blijven.

Dit beweegprogramma is onderdeel van het Move4Vitality platform en is gebaseerd op de grondmotrische eigenschappen en belastingvariabelen waarbij patientdata gerelateerd aan het beweegprogramma continue inzichtelijk zijn. Daarbij wordt er voorzien:

Een speciale app voor de patiënt

Een dashboard voor de fysiotherapeut

De techniek om automatisch data te interpreteren en verwerken, in die support voor dagelijkse fysiotherapeutische behandeling en dossiervorming.

Move4Vitality heeft als missie de vitaliteit van de medemens in zijn woon- en werkomgeving continue te verbeteren. Met vitaliteit in de breedste zin van het woord, zowel fysiek als geestelijk. En met expliciet aandacht voor bewegen, eten, slapen, stress en leren.

Move4Vitality ziet het als haar maatschappelijke rol om een gezonde levensstijl voor iedereen toegankelijk te maken. Op deze manier dragen wij bij aan het welzijn van onze samenleving.

De alsmaar stijgende zorgkosten, toename van het aantal mensen met een chronische ziekte, toename van vergrijzing en een groeiend tekort aan zorgprofessionals (zoals fysiotherapeuten), zorgen ervoor dat het voorkomen van gezondheid gerelateerde problemen steeds belangrijker wordt. De Nederlandse Zorgautoriteit heeft niet voor niets als een van haar speerpunten het realiseren van passende zorg - de juiste zorg, op de juiste plek, op het juiste moment. De traditionele fysieke zorg gaat de komende jaren dan ook een drastische verandering doormaken waarbij E-health het vergroten van eigen regie (in lijn met het gedachtengoed van Positieve Gezondheid [Institute for Positive Health, 2011]) een belangrijke rol gaat spelen. Door gebruik te maken van Move4Vitality ben je in staat om meer mensen te helpen en meer aandacht te geven aan mensen die het harder nodig hebben. De traditionele fysieke zorg gaat de komende jaren dan ook een drastische verandering doormaken. E-health gaat hierbij een grote rol spelen.

Planning

Het project plan heb ik bij het begin van het project geschreven om een duidelijk plan te maken voor het project. In dit document kun je alles vinden over wat er te maken heeft met de voorbereiding op het project. Hieronder staat de beschrijving van belangrijke aspecten van het project meer is te vinden in het [Project Plan](#).

Mogelijke implementatie

Momenteel wordt door Move4Vitality alleen gebruik gemaakt van Garmin activiteiten trackers. Data die gegenereerd wordt door iWatch, fitbit, google health en apple health worden nog niet gebruikt. Om zo'n groot mogelijk bereik te creëren is het gewenst om Google Health en Apple Health te koppelen aan het platform. Zodoende is Move4Vitality onafhankelijk van de activiteiten tracker die een deelnemer heeft.

De opdracht

Ik zal een API ontwikkelen waarmee ik, met de toestemming van klanten, gegevens kan ophalen uit Google Health en Apple Health. Deze gegevens zullen vervolgens worden verwerkt en opgeslagen in een database. Het doel is om deze gegevens weer te geven op het dashboard van Move4Vitality. Hierdoor krijgen fysiotherapeuten de mogelijkheid om activiteiten van klanten via Google health en Apple health in te zien op hun eigen dashboard. waardoor Move4Vitality niet langer afhankelijk is van de activiteitstracker die een deelnemer gebruikt. Dit zal de onafhankelijkheid van het platform vergroten en klanten meer flexibiliteit bieden bij het bijhouden van hun gezondheidsgegevens.

Doel

Het gewenste eindresultaat is een Technische realisatie van de koppeling met Apple Health voor IOS, Google Fit & Health Connect for ANDROID. De data moet 24/7 opgehaald en gepresenteerd kunnen worden in het platform en de app. Ook moet gekeken worden naar de voorwaarden waarop dit kan/dient te gebeuren.

Research Vragen

Hieronder staan de research vragen vermeld die ik onderzocht heb tijdens dit project, de deelvragen hebben mij geholpen om mijn hoofdvraag te beantwoorden. Een uitgebreid onderzoek over deze vragen kunt u vinden in het [Research Report](#).

Hoofdvraag

Hoe kan een integratie van third-party gezondheidsdata op een veilige manier in de huidige bedrijfssoftware gerealiseerd worden?

Deelvragen

- Welke activiteiten apps hebben de meeste integraties van populaire smartwatches?
- Hoe kan ik de API's van Apple Health en Google Health aanroepen?
- Welke technische en functionele vereisten zijn nodig voor het ontwikkelen van een systeem dat in staat is om gegevens van externe API's te verzamelen en deze gegevens weer te geven?
- Hoe moet omgegaan worden (uit technisch en regelgevings oogpunt) met privé/gezondheidsdata?

Aanpak

Voor dit project ga ik de scrum methode volgen. De stage periode duurt 18 weken dus ik ga sprints hanteren van 3 weken lang zodat ik 6 sprints in totaal heb. De eerste sprint ga ik mij focussen op wat gaat mijn plan zijn voor dit probleem. Ik ga bedenken hoe ik mijn opdracht ga maken en wat ik hier allemaal voor nodig heb. Al mijn functionaliteiten ga ik indelen via een moscow zodat ik weet waar de prioriteiten liggen, ook ga ik planning poker gebruiken om de moeilijkheidsgraad van de functionaliteiten in te schatten. De Scrum master van dit project ben ik en de product owner is mijn stage begeleider.

Aan het begin van elke sprint bekijk ik de product backlog op mijn Jira board, en bepaal ik welke items ik de komende sprint af wil hebben. Elk item op mijn Jira board bevat een definition of done checklist dit zijn checkboxen waaraan het product moet voldoen om afgerond te zijn. Aan het einde van elke sprint heb ik een sprint review waarbij ik de producten die ik deze sprint gerealiseerd heb presenteer aan de stakeholders. Ook worden de retrospectives en burndown charts bijgehouden in het [Scrum Document](#) na elke sprint.

Test Aanpak

Ik zal mijn code grondig testen door unit tests op te stellen. Daarnaast zal ik de expertise inroepen van een ervaren programmeur binnen het bedrijf om mijn code te beoordelen. Dit zal ik doen voor elke belangrijke functionaliteit die ik voltooi. Op deze manier kan ik ervoor zorgen dat mijn code schoon en van hoge kwaliteit blijft. Ik hecht veel waarde aan het testen van mijn code, vooral op het gebied van clean coding, omdat dit mijn vaardigheden als programmeur helpt verbeteren.

Beschrijving Process & Resultaat

Sprint 1

Deze sprint was actief van 04/09/2023 – 25/09/2023. Hieronder staan de taken waaraan gewerkt zijn in deze sprint met de bijbehorende storypoints.

- [Project Plan](#) (20)

Samenvatting

Deze sprint is het begin van mijn stage Mabs4.0. Ik begon met een meeting met mijn stagebegeleider en collega Ruben, die betrokken zijn bij het Move4vitality-platform. Hier kreeg ik een gedetailleerde uitleg over de organisatiestructuur van Mabs4.0 en de integratie van Move4vitality. Daarnaast bespraken Ruben en ik de technische aspecten van het platform, inclusief de gebruikte programmeertalen en de algemene werking.

Vervolgens wijdde ik mijn aandacht aan het onderzoek naar de koppeling van Google en Apple. Ik testte verschillende requests in de OAuth playground om inzicht te krijgen in de werking van de Google API.

In een opvolgende meeting met mijn assessor bespraken we het projectplan, waarna ik de rest van de week besteedde aan het implementeren van de ontvangen feedback. Ook had ik een gesprek met mijn stagebegeleider waarbij mijn genoteerde functionaliteiten en flowchart werden goedgekeurd.

De week daarop bracht ik verdere aanpassingen aan in het projectplan, waarbij ik onnodige use cases verplaatste naar het functioneel ontwerp. Hierdoor kreeg ik een overzicht van alle functionaliteiten in een document. Ik begon aan het technisch ontwerp waarin ik diagrammen van de testomgeving en notities noteerde over de data die ik van de Google API kon verkrijgen.

Ik besloot om een scrum document te maken zodat ik meer inzicht in het beheer van mijn scrum sprints kreeg en ontdekte tools in het Jira-board voor een helder overzicht, zoals rapportages en een

burn-down-chart. Tijdens mijn sprint review met mijn stagebegeleider heb ik de verzamelde data van de Google API besproken.

Na inlevering werd mijn projectplan goedgekeurd, wat betekent dat ik de volgende sprint aan de slag kan met mijn prototype en de doelen van mijn tweede sprint: afronding van de Google-login en gebruikerstoestemming verkrijgen, data ophalen en weergeven.

In deze sprint werd de keuze gemaakt om alleen Google Fit te integreren in mijn Proof of concept maar in latere sprints werd duidelijk dat dit alleen niet de beste keuze was omdat we dan veel integraties met smartwatches niet zouden hebben. Ik had eerder moeten kijken naar welke integraties het meest geschikt zouden kunnen zijn zodat ik mijn planning op deze verandering had kunnen voorbereiden.

Sprint 2

Deze sprint was actief van 25/09/2023 – 23/10/2023. Hieronder staan de taken waaraan gewerkt zijn in deze sprint met de bijbehorende storypoints.

- Als deelnemer moet ik kunnen inloggen via Google. (5)
- Als deelnemer moet ik toestemming kunnen geven om mijn gegevens te delen. (5)
- Als fysiotherapeut wil ik dat de dagelijkse stappen verzameld en verwerkt worden. (5)
- Als fysiotherapeut wil ik dat het aantal verbrande calorieën verzameld en verwerkt word.(5)
- Als fysiotherapeut wil ik dat de beweegminuten verzameld en verwerkt worden.(5)

Samenvatting

Deze sprint ben ik begonnen met een sprint review, waarbij ik mijn voortgang van de vorige sprint met mijn stagebegeleider heb besproken.

Vervolgens begon ik met de ontwikkeling van een prototype voor het koppelen van Google-accounts. Dit wordt geïmplementeerd via een Google-login, waarbij de gebruiker bij succes een toegangstoken genereert. Deze token wordt naar de Google API gestuurd om relevante data te verkrijgen.

Als eerste heb ik gewerkt aan de Google Login-functionaliteit en het verkrijgen van toestemming voor gegevens en dit met succes kunnen realiseren in een React Web applicatie.

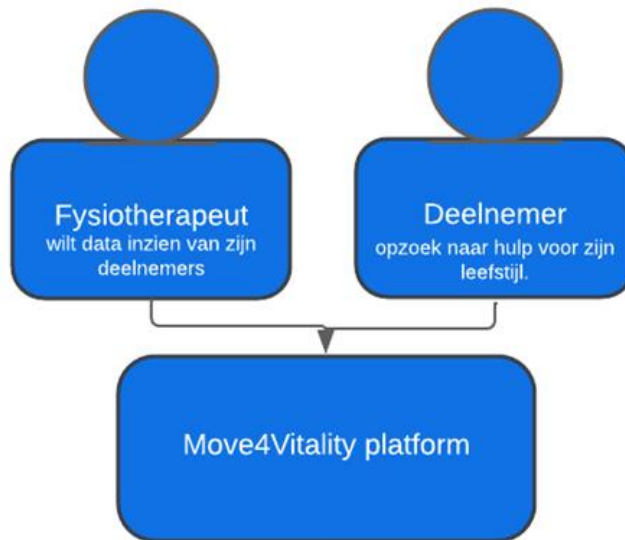
Tijdens een vergadering met mijn assesor en stagebegeleider werden mijn functioneel en technisch ontwerp besproken, ook vroeg ik om feedback over de infrastructuur van het project zodat ik zeker wist dat ik deze goed begreep.

Ik heb overleg gevoerd met Ruben over de keuze van Cosmos DB als database voor het project hiervoor is gekozen met name voor het opslaan van gezondheidsgegevens, omdat dit grote hoeveelheid data is en dat is een stuk makkelijker om uit te lezen en op te slaan met een NoSQL database om de snelheid van het platform te behouden.

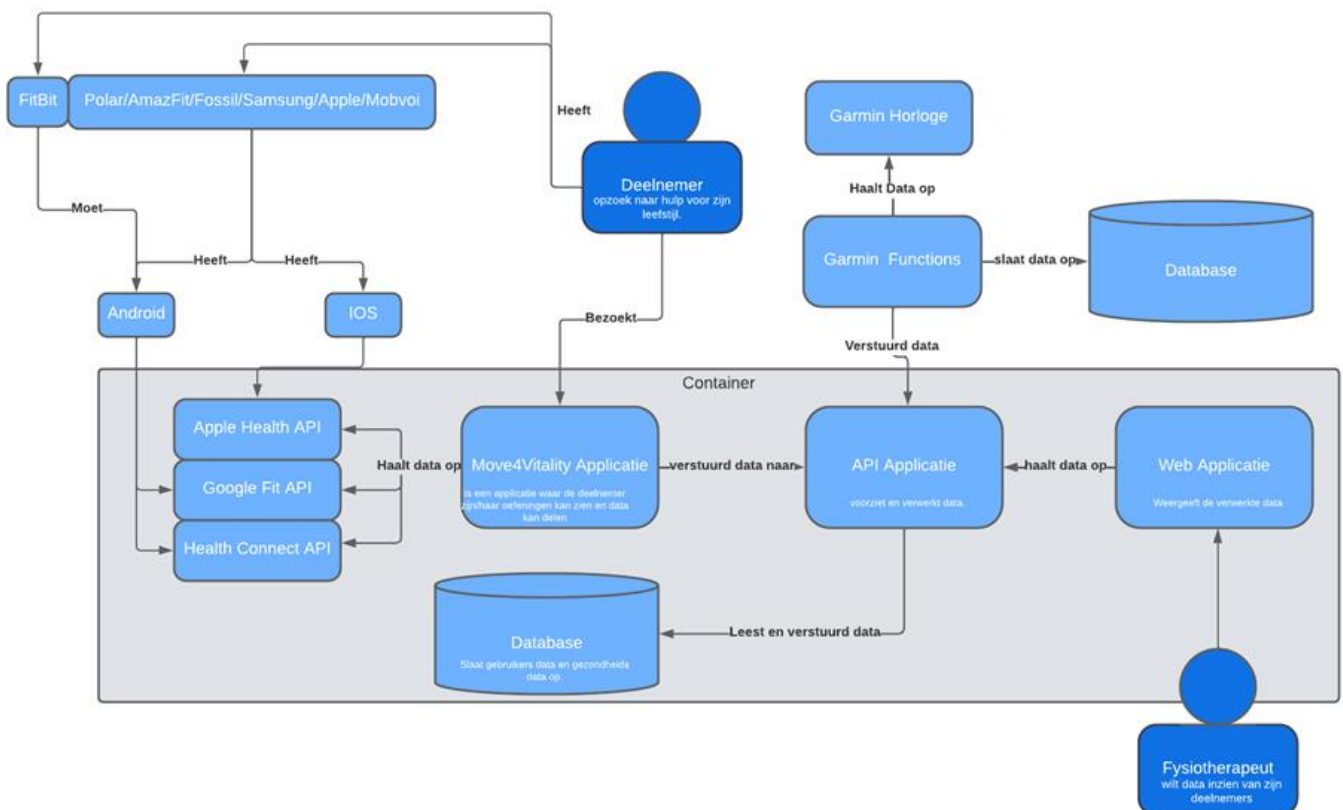
voorlopig ga ik mij concentreren op het verzamelen van activiteitengegevens via de Google API. Ondertussen kan ik het aantal stappen, het aantal beweegminuten en het basaal metabolisme verzamelen en verwerken in mijn Proof of Concept.

Ik had alle geplande functionaliteiten voor deze sprint eerder dan verwacht afgerond, hierdoor weet ik dat ik mijn werklading kan verhogen voor volgende sprint planningen. Ook heb ik klassendiagrammen gemaakt en de niveaus C1, C2 en C3 van het project uitgetekend, waardoor ik een duidelijk overzicht kreeg van het systeem.

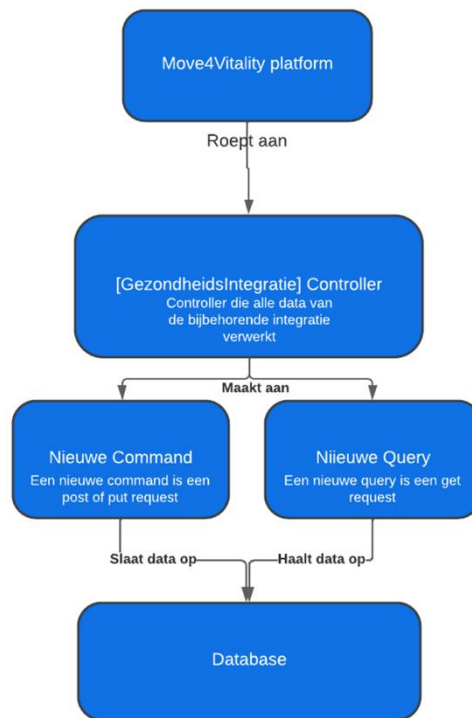
C1 Diagram



C2 Diagram



C3 Diagram



Deze sprint heb ik ook onderzocht hoe asynchronous (async) kan worden toegepast in mijn API en frontend waardoor mijn functionaliteiten zoals het ophalen van gegevens tegelijk uitgevoerd kunnen worden, om de efficiëntie van de applicatie te verbeteren. Ik heb toegang gekregen tot mijn eigen Cosmos DB en ben begonnen met het opslaan van gegevens.

In een gesprek met Roy werden nieuwe prioriteiten vastgesteld, waarbij de focus verschoof van Google Fit. Roy legde mij uit dat Google Fit alleen als implementatie niet de meeste dekking bied en ik moest gaan uitzoeken wat de beste implementatie zou zijn in de Move4Vitality app om de meeste dekking te krijgen met smartwatches. Ik startte een onderzoek naar de meest uitgebreide integratie terug te vinden in het [Research Report](#).

Voor mijn onderzoek gebruikte ik de meeste smartwatches die ik kon vinden in mijn bronnen en de populairste gezondheids apps in de App store & Google Play store. Na vergelijking van deze integraties concludeerde ik dat voor iOS Apple Health de beste keuze was. Voor Android kozen we voor Google Fit, ondanks de verouderde API, en voegden we Health Connect toe om ook dekking te hebben voor Samsung watches en FitBit. De Move4Vitality-app zal integraties bevatten voor Apple Health, Google Fit en Health Connect, waarbij Apple Health de hoogste prioriteit heeft.

Diagram van de gewenste dekking voor het Move4Vitality platform.

Watches Apps	FIT BIT	SAMSUNG	APPLE	GARMIN	POLAR	AMAZFIT	FOSSIL	MOBVOI
IOS(Apple health)	X	V	V	V	V	V	V	V
ANDROID(Google Fit/Health Connect/Garmin Connections)	V	V	V	V	V	V	V	V

In deze sprint realiseerde ik me dat het niet verstandig was om mijn Proof of Concept in een React Web Applicatie te ontwikkelen. Uiteindelijk moest ik het namelijk implementeren in een React Native-applicatie. Ik veronderstelde dat de code vrijwel hetzelfde zou zijn, maar de implementatie in React Native blijkt aanzienlijk anders te zijn. Met nader inzien had ik mijn Proof of Concept beter meteen in een React Native App moeten ontwikkelen.

Sprint 3

Deze sprint was actief van 23/10/2023 – 13/11/2023. Hieronder staan de taken waaraan gewerkt zijn in deze sprint met de bijbehorende storypoints.

- Data & gebruiks vergelijking maken met de Gezondheids integraties. (13)
- (HC) Als fysiotherapeut wil ik dat de dagelijkse stappen verzameld en verwerkt worden.(8)
- (HC) Als fysiotherapeut wil ik dat het aantal verbrande calorieën verzameld en verwerkt worden.(8)
- (HC) Als fysiotherapeut wil ik dat het basaal metabolisme verzameld en verwerkt word.(8)

Samenvatting

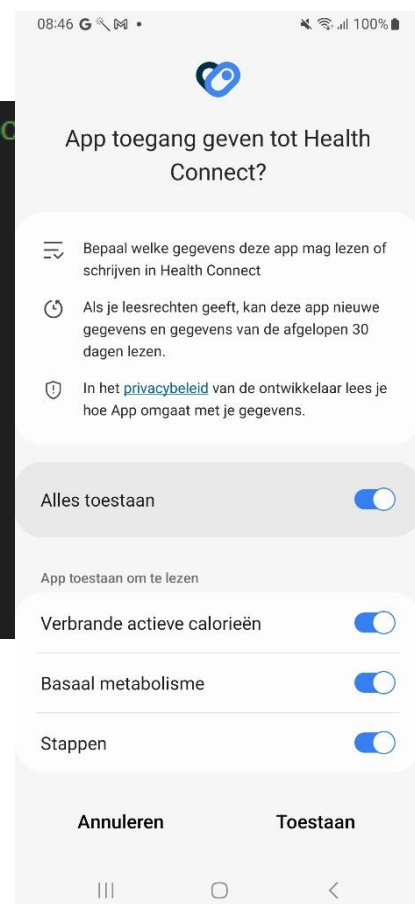
Deze sprint begon ik met de ontwikkeling van het Proof of concept in React Native, met als doel het zo te ontwikkelen dat het precies hetzelfde geïmplementeerd kan worden in de Move4Vitality app.

Voor Apple Health moest ik een MacBook gebruiken, die binnenkort beschikbaar zou zijn na het vervangen van de accu. Roy vertelde mij daarom om mijn sprintplanning aan te passen en te concentreren op de implementatie van Health Connect. Om het project op Android te testen, kreeg ik een Android-telefoon van de stage.

In de implementatiefase van Health Connect onderzocht ik toestemming vragen aan gebruikers en startte ik met het ophalen van dagelijkse stappen.

Deze code snippet weergeeft hoe toestemming gevraagd word.

```
//requests the permissions stated below
await requestPermission([
  {
    accessType: 'read',
    recordType: 'Steps',
  },
  {
    accessType: 'read',
    recordType: 'Distance'
  }
]);
```



Dit is hoe toestemming verlenen uitziet voor een deelnemer

```
async function retrieveSteps(timeRangeFilter:TimeRangeFilter) {  
  //retrieves steps from health connect using the filter.  
  const steps = await readRecords('Steps', { timeRangeFilter });  
  const totalSteps = steps.reduce((sum, cur) => sum + cur.count, 0);  
  setSteps(totalSteps);  
}
```

Omdat Expo Go geen native code ondersteunt, bouwde ik mijn eigen app via Expo Go om deze code te ondersteunen. Om testgegevens toe te voegen, koppelde ik tijdelijk mijn Google Fit account aan HealthConnect omdat je op HealthConnect zelf geen gegevens kan toevoegen.

In de volgende fase streefde ik ernaar de dagelijkse stappen naar mijn API te sturen, maar stuitte op een network error bij het versturen van het POST-verzoek vanaf de Android-telefoon. Ondanks aanpassingen aan de aanroep en het openen van poorten op mijn laptop, bleef het probleem bestaan, en vroeg ik Ruben om hulp zonder direct succes.

De daaropvolgende week concentreerde ik me op het oplossen van de network error en besprak ik dit met mijn assessor. Na diverse pogingen ontdekte ik dat mijn API automatisch HTTP naar HTTPS omzette, en na het verwijderen van het HTTPS-adres werkte alles zoals verwacht. Ik kon succesvol stappen, basaal metabolisme en verbrande calorieën verzamelen en naar mijn API sturen in mijn React Native App.

Post Request naar Move4Vitality API stuurt TrackerResult door.

```
export const postTrackerData = async (args: postTrackerDataArgs) => {  
  const {calendarDate, steps, distanceInMeters} = args;  
  //gets jwt token  
  const jwt = await getCredentials();  
  return DashboardApi.post(`/GoogleHealth/SaveTrackerResults`, {  
    calendarDate,  
    steps,  
    distanceInMeters  
  },  
  {  
    headers: {  
      Authorization: `Bearer ${jwt?.accessToken}`  
    }  
  }).then((result) => {  
    console.log(result);  
  }).catch((err) => {  
    console.log(err);  
  });  
};
```

Na een gesprek met mijn assessor diende ik mijn research report in voor feedback en besteedde ik de rest van de week aan het verwerken van ontvangen suggesties, met de intentie hier verder mee te gaan in de volgende sprint.

Sprint 4

Deze sprint was actief van 13/11/2023 – 01/12/2023. Hieronder staan de taken waaraan gewerkt zijn in deze sprint met de bijbehorende storypoints.

- [Research Report](#)(13)
- Implementatie van Health connect in de bestaande Move4Vitality app(20)
- Implementatie van Health connect code in de backend API(20)

Samenvatting

In de eerste week heb ik mijn Research Report verder aangepast op basis van feedback van mijn assessor en opnieuw ingeleverd. De sprint review met Roy was erg positief hij was erg tevreden over mijn inzet en wat ik tot nu toe had opgeleverd, Vooral omdat het proof of concept voor Health Connect was voltooid. Deze sprint zal ik werken aan de integratie van dit concept in het bestaande platform. Ik ben tevreden met mijn tussentijdse beoordeling van het assessment document waar ik vanuit stage een Goed voor kreeg.

Op de terugkomdag op school heb ik mijn tweede assessor ontmoet en een [presentatie](#) gegeven over mijn project. Ik ontving waardevolle feedback van medestudenten, met aandachtspunten zoals de opslag van gezondheidsdata met betrekking tot encryptie en veiligheid. Vervolgens ben ik begonnen aan mijn portfolio zodat ik hier niet op achterloop en ik het wekelijks kan updaten.

In de tweede week heb ik met mijn assessor over het implementeren van Health Connect in de API-backend gesproken. Ik ondervond uitdagingen vanwege de Clean Architecture structuur die ze gebruikten. Ik had namelijk niet door dat een API zo erg kon verschillen qua structuur, ik dacht dat alle API's wel min of meer hetzelfde qua structuur zijn. Na overleg met Ruben begreep ik dat de keuze voor structuur afhangt van teamvoorkeuren en ervaring. Ondanks de verschillen kon ik de backend succesvol implementeren, eerst heb ik een Api post request gemaakt die een parameter moet ontvangen die de huidige datum bevat, het totaal aantal stappen van de dag en de totaal afgelegde afstand in meters. Dit is de enige data die op dit moment belangrijk is om door te sturen naar de API volgens Ruben.

Api post call voor Health Connect en Google Fit wat Tracker Resultaat opslaat.

```
public class GoogleHealthController : ApiControllerBase
{
    [HttpPost("SaveTrackerResults")]
    [AuthorizeRole(RoleName.Client)]
    public Task SaveTrackerResults(SaveTrackerResultsBody body) =>
        Mediator.Send(
            new SaveTrackerResultsCommand
            {
                CalendarDate = body.CalendarDate,
                Steps = body.Steps,
                DistanceInMeters = body.DistanceInMeters,
            });
}
```

Deze api call maakt vervolgens een SaveTrackerResultsCommand aan waarbij deze data vervolgens gecontroleerd wordt en daarna wordt opgeslagen.

Method die aangeroepen word in `SaveTrackerResultsCommand` om data op te slaan in de database

```
1 reference
private async Task CreateOrUpdateTrackerDay(SaveTrackerResultsCommand command, CancellationToken ct)
{
    var calendarDate = command.CalendarDate.ToCalendarDate();
    var client = await TryGetClientAndSingleTrackerDay(_caller.Id, calendarDate, ct);

    if (client.UtcSubscriptionIncludesTrackerSince != null)
        throw new BadRequestException("Client should have subscription excluding tracker.");

    var trackerDay = client.TrackerDays.SingleOrDefault();
    if (trackerDay == null)
    {
        trackerDay = new TrackerDay(client, calendarDate);
        client.TrackerDays.Add(trackerDay);
    }

    var utcNow = _dateService.UtcNow;
    trackerDay.UpdateSteps(command.Steps, utcNow);
    trackerDay.UpdateDistance(command.DistanceInMeters, utcNow);
    client.UtcTrackerSynchronised = utcNow;
    await _db.SaveChangesAsync(ct);
}
```

Over deze method heb ik unit tests geschreven om deze goed te kunnen testen.

Unit test for checking if error message is thrown when client does not have subscription.

```
[Fact]
0 references
public async Task WhenClientWithoutSubscriptionExcludingTracker_ShouldThrowBadRequestException()
{
    //arrange
    var entities = BootstrapDbEntities();
    entities.BaselineProgramme.ActivatedAndCurrentOn(FakeToday);
    await _dbContext.SaveChangesAsync();

    var expectedSteps = 9999;
    var expectedDistanceInMeters = 200;
    var saveTrackerResultsBody = new SaveTrackerResultsBody
    {
        CalendarDate = FakeToday.ToDefaultDateString(),
        Steps = expectedSteps,
        DistanceInMeters = expectedDistanceInMeters
    };

    //act
    var response = await HttpPost(entities.Client, RequestUri(), saveTrackerResultsBody);

    //assert
    await response.ShouldBeBadRequestWithErrorMessageContaining("Client should have subscription excluding tracker.");
}
```

In de derde week ligt de nadruk op mijn inspanningen om Health Connect te implementeren in de React Native-app van Move4Vitality. Na enkele dagen van technische uitdagingen, waaronder het verhelpen van foutmeldingen en het bijwerken van SDK-versies, lukte het me uiteindelijk om Health Connect succesvol aan de bestaande Move4Vitality app te koppelen. Op vrijdag heb ik een koppelingsknop toegevoegd, waarmee deelnemers toestemming kunnen geven voor het lezen van benodigde gegevens.

Hoewel het dashboard nu het aantal stappen weergeeft, werk ik nog aan het verzenden van deze gegevens naar de API, wat nog niet operationeel is. Helaas kon ik Health Connect niet volledig implementeren deze sprint, maar dit zal worden opgepakt in de volgende sprint.

In deze sprint ontdekte ik tijdens het implementeren van mijn API-code in de bestaande API dat hun API-structuur aanzienlijk verschilt van die in mijn Proof of Concept. Hoewel ik wist dat code sterk kon verschillen, ging ik ervan uit dat de API-structuur overal min of meer hetzelfde zou zijn, met controllers, managers, entiteiten en DTO's. Echter, de realiteit bleek anders te zijn, wat voor mij een eyeopener was om me verder te verdiepen in verschillende structuren en deze later uit te proberen.

Sprint 5

Deze sprint was actief van 04/12/2023 – 22/12/2023. Hieronder staan de taken waaraan gewerkt zijn in deze sprint met de bijbehorende storypoints.

- Implimentatie van Health connect in de bestaand Move4Vitality app(20)
- Implimentatie van Google fit connectie in de bestaande Move4Vitality app(12)
- Dagelijkse stappen ophalen & doorsturen van Google Fit(5)
- Dagelijkse afgelegde afstand ophalen & doorsturen van Google Fit(5)

Samenvatting

Tijdens een meeting met mijn stagebegeleider hebben we de huidige status van het project doorgenomen en de planning voor de komende sprint besproken. De eerste week heb ik gewerkt aan het doorsturen van de dagelijkse stappen naar de API in de bestaande Move4Vitality app. In het instellingen scherm kan de deelnemer nu op de knop “healthconnect” drukken, toestemming geven om gegevens te lezen, en vervolgens worden het aantal dagelijkse stappen en de afstand doorgestuurd en opgeslagen in de database.

In week 2 stond er een afspraak gepland met Ruben om mijn code te laten reviewen en te zorgen dat deze voldoet aan de standaard hier is Ruben met mij door heengelopen, hebben we het een en andere samen verbeterd en vervolgens was mijn code goedgekeurd. Ik heb al mijn code voorzien van commentaar om duidelijk te maken hoe alles werkt.

Tijdens een gesprek met Ruben heb ik ook mijn zorgen geuit over de beveiliging van gezondheidsgegevens, zoals ik had aangegeven tijdens mijn tussentijdse presentatie. Ruben heeft me gerustgesteld door te vertellen dat de databases in Azure zijn versleuteld, waardoor ik me geen zorgen hoeft te maken. Ik vroeg ook of het nodig was om gezondheidsdata te versleutelen bij het versturen naar de API, en Ruben verzekerde me dat HTTPS voldoende is om dataverlies te voorkomen, en dat ik voor de scope van mijn project mij hier niet op hoeft te focussen.

Vervolgens ben ik mij gaan focussen op het implementeren van Google Fit in de Move4Vitality app naar de wens van Roy. Google Fit werkt vanuit de Google Cloud daarom moet de Fitness API toegevoegd worden aan het Google Cloud project van Move4Vitality. Hiermee ben ik naar Ruben gegaan om te vragen of hij deze API kan toevoegen aan het Google Cloud-project. Ruben had ook geen toegang tot het project waardoor ik hiermee naar Roy ben moeten toestappen om Ruben toegang te verlenen waardoor hij uiteindelijk de Fitness API heeft kunnen toevoegen.

Tijdens het implementeren van Google Fit kwam ik erachter dat je ook de Google Fit Rest API in de React Native app kunt aanroepen in plaats van de Google Fit Android API. Dus ik heb ervoor gekozen om de Rest API te gebruiken in de Move4Vitality app, omdat de Google Fit Android API verouderd is en eind 2024 vervalt. Door dit te doen, kan de Google Fit-integratie permanent zijn en is het geen tijdelijke implementatie zoals we hadden verwacht.

Het enige nadeel hiervan is dat de gebruiker bij koppeling met Google Fit moet inloggen met zijn google account, dit zou niet hoeven als de Android API gebruikt zou worden want dan zou het account van de Google Fit app op de telefoon gebruikt worden om automatisch in te loggen.

Ik heb op het instellingenscherf een knop toegevoegd voor de Google Fit-verbinding. Zodra de deelnemer hierop klikt, moet de deelnemer inloggen met zijn Google-account en vervolgens toestemming geven, zodat de app data kan uitlezen.

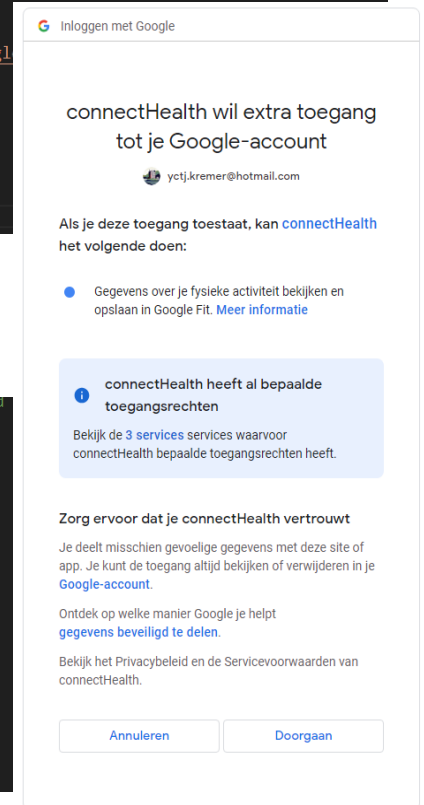
Google sign-in waarbij toestemming gevraagd word door scopes.

```
export const googleAuthentication = async () => {  
  //set the permissions to authorize access to.  
  GoogleSignin.configure({  
    scopes: ['https://www.googleapis.com/auth/fitness.activity.read', 'https://www.google.com/auth/fitness.location.read']  
  });  
  GoogleSignin.signIn().then((result) => {  
    console.log(result);  
  }).catch((error) => {  
    console.log(error);  
  })  
}
```

Vervolgens haal ik nu de dagelijkse stappen op van Google Fit en ben ik bezig met het ophalen van de afstand die een deelnemer heeft afgelegd.

Post Request voor dagelijkse stappen op te halen vanuit Google Fit

```
//post request that sends aggregate filters and retrieves steps between beginning of today and end of today.  
axios.post("https://www.googleapis.com/fitness/v1/users/me/dataset:aggregate",  
{  
  "aggregateBy": [{  
    "dataTypeName": "com.google.step_count.delta",  
    "dataSourceId": "derived:com.google.step_count.delta:com.google.android.gms:estimated_steps"  
  }],  
  "bucketByTime": { "durationMillis": 86400000 },  
  "startTimeMillis": startdate.getTime(),  
  "endTimeMillis": enddate.getTime()  
},  
{  
  headers: { Authorization: 'Bearer ' + (await GoogleSignin.getTokens()).accessToken }  
}).then((result) => {  
  setSteps(result.data.bucket[0].dataset[0].point[0].value[0].intValue);  
}).catch((error) => {  
  console.log(error);  
})
```



Deelnemer weergave van het toestemming verlenen voor Google Fit

Week 3 heb ik besteed aan verdere implementatie van Google Fit. Ik ben erin geslaagd dagelijks afgelegde afstanden van deelnemers op te halen, en deze worden nu samen met stappen en de huidige datum succesvol doorgestuurd naar de API en vervolgens opgeslagen in de database.

Post Request voor ophalen van totale afstand afgelegd door een deelnemer vanuit Google Fit

```
//post request that sends aggregate filters and retrieves total distance between beginning of today and end of today.  
axios.post("https://www.googleapis.com/fitness/v1/users/me/dataset:aggregate",  
{  
  "aggregateBy": [{  
    "dataTypeName": "com.google.distance.delta",  
    "dataSourceId": "derived:com.google.distance.delta:com.google.android.gms:merge_distance_delta"  
  }],  
  "bucketByTime": { "durationMillis": 86400000 },  
  "startTimeMillis": startdate.getTime(),  
  "endTimeMillis": enddate.getTime()  
},  
{  
  headers: { Authorization: 'Bearer ' + (await GoogleSignin.getTokens()).accessToken }  
}).then((result) => {  
  setDistanceInMeters(parseInt(result.data.bucket[0].dataset[0].point[0].value[0].fpVal))  
}).catch((error) => {  
  console.log(error);  
})
```

Naar aanleiding van feedback van mijn assessor heb ik deze laatste week van de sprint besteed aan mijn functioneel/technisch ontwerp om hier een logisch verhaal. In de app heb ik ook de mogelijkheid toegevoegd om te ontkoppelen van Google Fit & Health Connect, dit is essentieel om te voldoen aan de regelgeving omtrent gezondheidsdata. Ook heb ik een bug weten op te lossen die ervoor zorgde dat de connectie met de gezondheidsintegraties niet verbroken werden.

Mijn laatste week van het project heb ik besteed aan het verbeteren van al mijn documentatie. Ik heb mijn laatste stagebezoek gehad met mijn stagebegeleider, hier heb ik nog een eindpresentatie gegeven en dit verliep goed. Ik ga deze week nog alles voorbereiden en overdragen aan mijn stage, ook heb ik het Assessment form aan Roy doorgestuurd voor mijn laatste beoordeling vanuit stage.

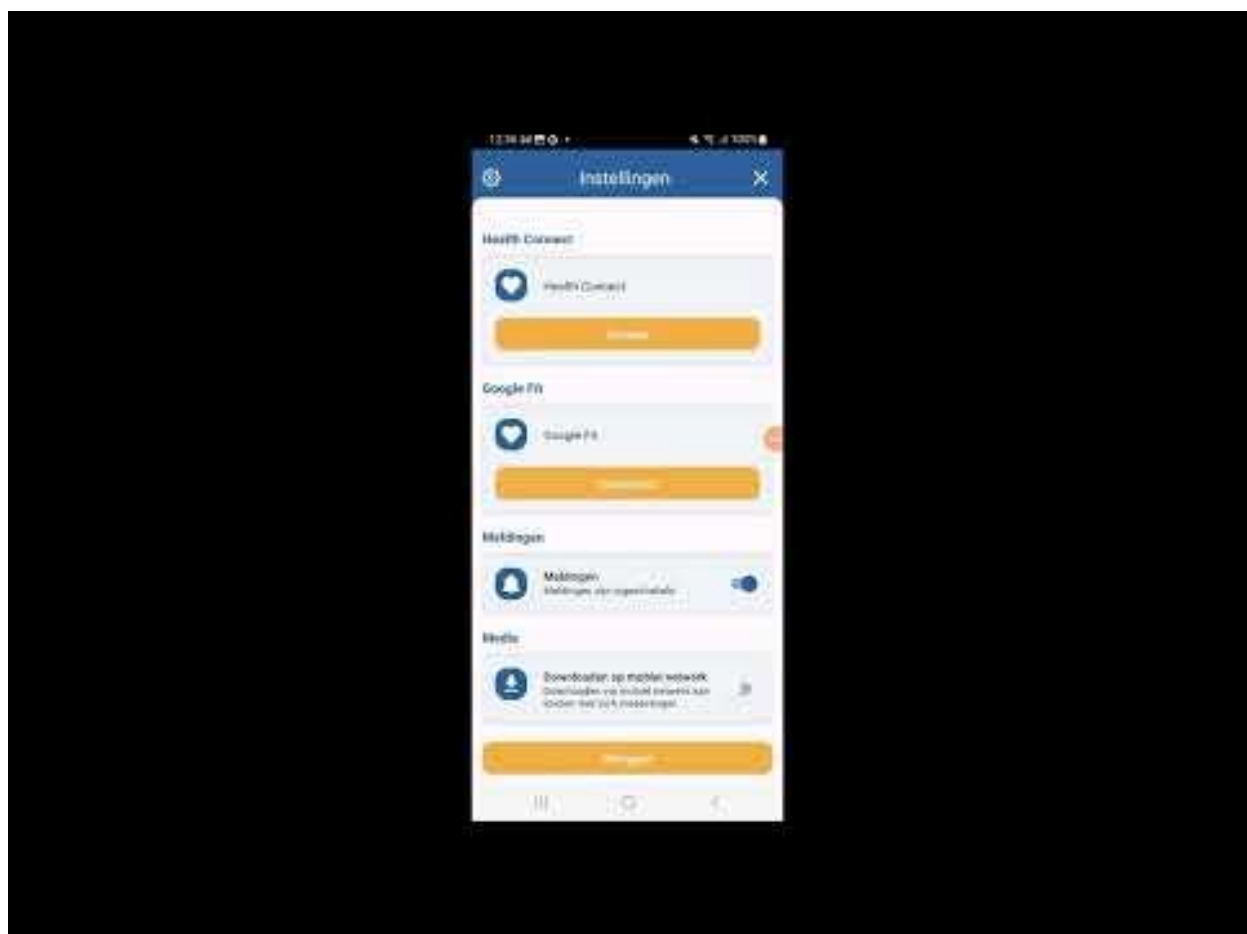
Conclusie

Dit stageproject heb ik kunnen afronden met de succesvolle implementatie van Google Fit en Health Connect in het huidige systeem van Move4Vitality. In eerste instantie was het doel om ook nog Apple Health te realiseren maar door gebrek aan tijd en benodigde technische apparatuur zoals een macbook was dit niet mogelijk. Ondanks dit heb ik de gehele android kant van de Applicatie kunnen afmaken.

Door dit project is Move4Vitality nu onafhankelijk van welke activiteiten tracker een deelnemer heeft. Een deelnemer kan op android zijn Move4Vitality app nu verbinden met alle Wear OS, Fitbit OS en Garmin OS watches.

Hieronder vind u een demo van hoe de deelnemer een connectie kan leggen met de externe gezondheidsapps via de Move4Vitality app.

<https://youtu.be/McFnWKF1744>



Aanbevelingen

Move4Vitality App Verifiëren

Op het moment als de deelnemer een connectie legt met Google Fit en de deelnemer inlogt komt er een beveiligings melding die zegt "Google heeft deze app niet geverifieerd". Vervolgens wil Google de deelnemer allarmen dat voorzichtig gedaan moet worden, omdat de app voor gezondheidsgegevens vraagt maar dus niet gecontroleerd is of deze app wel echt veilig is.

Dit is geen goede blik voor de deelnemer dus ik zou dit adviseren als een van de eerst volgende stappen.

Dit kan gedaan worden door in de Google Cloud het project "Google Android Developer" te publishen, Vervolgens te klikken op prepare for verification knop. Na het klikken hierop krijg je een scherm met alle informatie over je app deze moet je dan conrolleren en opslaan.

Hierna moet worden uitgelegd hoe de scopes in dit project worden gebruikt in de app. En als laatste moet er een demo video worden opgestuurd net zoals de video weergeven bij de Conclusie. Na het voltooien van deze stappen kan de app opgestuurd worden voor verificatie. Na succesvolle verificatie zal deze beveiligings melding verdwijnen. [Hier kunt u de stappen vinden voor het verifiëren van een app.](#)

Apple Health Implementatie

Op het moment is alleen een Garmin connectie mogelijk op de Move4Vitality app aan de IOS kant. Dus de implementatie van Apple Health in de Move4Vitality app zou een grote vooruitgang zijn en zo is ook de IOS kant van de app beschikbaar voor connecties met alle Wear OS activiteitentrackers. De enigste activiteiten tracker die niet op IOS geïmplementeerd kan worden is Fitbit want dit is een directe concurrent van Apple.

Persoonlijke reflectie

Ik ben heel erg tevreden over mijn resultaten rondom dit project. Tijdens dit project heb ik met veel nieuwe dingen gewerkt, NoSql databases, Azure, Clean Architecture voor c#, react native, Google Fit API, Health Connect API etc, en zo dus ook heel veel dingen geleerd. Interessante ontdekkingen gedaan zoals erachter komen dat er heel veel structuren zijn om je API in te schrijven en het nut van Asynchronous functions. Ook ben ik erachter gekomen dat ik een paar fouten heb gemaakt zoals het maken van mijn Proof of concept met andere programma's, structuren en frameworks, waardoor ik uiteindelijk dubbel werk heb moeten verrichten bij het overzetten naar het huidige systeem. Nog een grote fout die ik had gemaakt was niet de mogelijke toekomstige problemen inschatten van een implementatie, zoals het alleen implementeren van Google Fit terwijl dit op zichzelf geen goede dekking voor activiteitentrackers zou bieden.

Ik heb dit project ook goed kunnen werken aan mijn persoonlijke doelen zoals meer feedback vragen voor mijn documentatie/implementaties wat mij heel erg heeft geholpen om betere resultaten op te leveren.

Ook wilde ik graag deze stage mijn Project afronden, nu heb ik dit kunnen voltooien voor zover mogelijk was en is Roy tevreden met mijn resultaat. Natuurlijk is een project nooit 100% af en kan er altijd iets verbeterd worden. Ik ben erg blij dat ik nu snap hoe je een connectie kan leggen met externe API's, in de huidige markt word dit vaak gebruikt om extra werk te voorkomen en dit lijkt mij daarom een handige skill om te beheersen.

Producten

Al deze producten staan in de Producten folder in mijn portfolio en deze zijn ook downloadbaar als u dit document opent in .docx door op het object te klikken. (niet downloadbaar in pdf)

Mid Term Presentatie

Een Powerpoint presentatie die ik heb gepresenteerd halverwege mijn stage om mijn opdracht duidelijk te maken aan mijn medestudenten en mijn assessors.



MidTermPresentation.pdf

Project Plan

Het project plan is opgesteld bij het begin van dit project om een duidelijke planning te creëren, alle stakeholders te definiëren en de aanpak van het project te beschrijven.



ProjectPlan.pdf

Scrum Document

Het Scrum document is een document waarbij elke sprint is bijgehouden waarin ik reflecteer elke week wat ik heb bereikt en waar ik tegenaan liep ook is er voor elke sprint een burndown chart en een burnup chart weergegeven om het verloop van de sprint in beeld te brengen.



ScrumDoc.pdf

Research Report

Het Research report is een document waarin al mijn research vragen worden beantwoord die ik heb behandeld tijdens dit project.



researchreport.pdf

Functioneel & Technisch Ontwerp

Het Functioneel & Technisch ontwerp bevatten allerlei diagrammen/designs die ik gemaakt heb tijdens dit project om duidelijkheid voor mijzelf/stakeholders te creëren.



functioneel_Technisch_Ontwerp.pdf