

Research report



**well NETWORKED
top CONNECTED**

business models • smart services • strategies • solutions

Yordi Kremer
DATUM: 07/09/2023

Table of Contents

Table of Contents	1
1 Onderzoeksvragen	2
1.1 Hoofdvraag.....	2
1.2 Deelvragen	2
2 Onderzoek 1.....	6
2.1 Vraag	6
2.2 Hoe ben ik aan de slag gegaan?	6
2.2.1 Library Research - Literature Study	6
2.2.2 Library Research – Community research	7
2.2.3 Workshop Research - IT architecture sketching	8
2.2.4 Workshop Research – Prototyping.....	9
2.3 Resultaat	9
2.4 Qualiteit van het resultaat	9
3 Onderzoek 2.....	11
3.1 Vraag	11
3.2 Hoe ben ik aan de slag gegaan?	11
3.2.1 Field Research – Problem analysis	11
3.2.2 Field Research – Explore user requirements.....	11
3.3 Resultaat	12
3.4 Qualiteit van het resultaat	12
4 Onderzoek 3.....	13
4.1 Vraag	13
4.2 Hoe ben ik aan de slag gegaan?	13
4.3 Resultaat	14
4.4 Qualiteit van het resultaat	14

1 Onderzoeksvragen

1.1 Hoofdvraag

Hoe kan een integratie van third-party gezondheidsdata (apple, google) op een veilige manier in de huidige bedrijfssoftware gerealiseerd worden, zodat meer klanten van move4vitality gebruik kunnen maken van het delen van data met hun fysiotherapeut?

1.2 Deelvragen

- Welke gezondheids 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?

2 Onderzoek 1

2.1 Vraag

Welke gezondheids apps hebben de meeste integraties van populaire smartwatches?

2.2 Hoe ben ik aan de slag gegaan?

2.2.1 Library Research- Literature Study

Ik ben begonnen met het onderzoeken van de meest populaire/gebruikte smartwatches. Ik heb gekeken naar lijsten op verschillende websites en ben uitgekomen op 8 smartwatches die ik ga vergelijken met gezondheidsapps. Hierna heb ik de meest gebruikte gezondheidsapps op iOS en Android opgezocht. Na het raadplegen van meerdere bronnen ben ik uitgekomen op 13 apps

2.2.2 Workshop Research – Multi Criteria decision making

Om een duidelijk overzicht te krijgen van welke integraties met gezondheidsapps het beste kunnen worden gebruikt voor de Move4vitality app, heb ik een diagram gemaakt waarin ik de 8 smartwatches vergelijk die ik hiervoor heb gevonden met de 13 gezondheidsapps. Hier zal ik kijken welke integraties op Android en iOS de meeste dekking bieden in termen van connecties met smartwatches. In dit diagram worden alleen de directe verbindingen met de smartwatch-apps bekeken.

2.3 Criteria van het Resultaat

Een diagram waarin duidelijk wordt gemaakt wat de beste keuze(s) is/zijn als integratie met de Move4Vitality-app om de meeste dekking te hebben op iOS en Android met de populairste smartwatches.

2.4 Resultaat

Hier vindt u het diagram dat de vergelijking toont tussen de 8 meest populaire smartwatches en hun mogelijke integratie met de meest gebruikte gezondheidsapps.

Watches	FIT BIT	SAMSUNG	APPLE	GARMIN	POLAR	AMAZFIT	FOSSIL	MOBVOI
Apps		G						
Apple health (IOS)	X	V	V	V	V	V	V	V
Google fit (BOTH)	X	X	V	X	V	V	V	V
Samsung health (BOTH)	X	V	V	X	X	X	X	X
Health Connect (ANDROID)	V	V	X	X	X	X	X	X
Health Connect : Health tracker (IOS)	X	X	V	X	X	X	X	X
Fit Bit app (BOTH)	V	X	X	X	X	X	X	X
Garmin connect (BOTH)	X	X	V	V	X	X	X	X
Strava (BOTH)	V	V	V	V	V	V	V	V
Polar flow (BOTH)	X	X	V	X	V	X	X	X
Zepp (BOTH)	X	X	V	X	X	V	X	X
Fossil Smartwatches (BOTH)	X	X	V	X	X	X	V	X
Mobvoi App (BOTH)	X	X	V	X	X	X	X	V
Healthsync (ANDROID)	V	V	X	V	V	X	X	X

Dit diagram heb ik besproken met Roy, en samen zijn we tot de conclusie gekomen dat Apple Health de meeste dekking biedt op iOS, maar helaas geen Fitbit-ondersteuning mogelijk zou maken voor iOS-gebruikers. Voor Android was het ingewikkelder; Google Fit bood de meeste dekking voor Android, maar dan zouden we geen ondersteuning hebben voor Samsung, Fitbit en Garmin. Omdat Garmin al geïmplementeerd is, vervalt dit nadeel. Ook is de Google Fit Android API verouderd en zal deze aan het einde van 2024 worden stopgezet. Google werkt nu aan Health Connect op Android, dat tot nu toe een verbinding biedt met Fitbit en Samsung Health, wat precies is wat ontbreekt in Google Fit.

Als conclusie om de meeste dekking te garanderen, zijn we tot de volgende conclusie gekomen:

Voor de iOS-app hebben we alleen de Apple Health-integratie, en hiermee dekken we alles, behalve FitBit. Voor de Android-app hebben we de Google Fit-integratie, de Health Connect-integratie en de Garmin-integratie. We willen graag de Google Fit geïmplementeerd houden om een brede dekking te behouden tot het einde van 2024, in de hoop dat tegen die tijd de Health Connect-app van Google verder is uitgebreid. Op deze manier dekken we alle vereiste koppelingen

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

3 Onderzoek 2

3.1 Vraag

Hoe kan ik de API's van Apple Health en Google Health aanroepen?

3.2 Hoe ben ik aan de slag gegaan?

3.2.1 Library Research- Literature Study

Als eerste ben ik gaan uitvogelen wat apple en google health inhoud, bij apple heb je de [gezondheids app](#) en bij google heb je de [google fit app](#). Deze apps houden data bij zoals:

- Activiteiten
- Blood glucose
- Bloeddruk
- Lichaamsafmetingen
- Lichaamstemperatuur
- Hartslag
- Locatie
- Voeding
- Zuurstofverzadiging
- Slaap

Na dit overleg ben ik begonnen met het bestuderen van de documentatie van de [Google Fit API](#) en de [Apple Health API](#). Aangezien de integratie van deze API's aanzienlijk verschilt, en omdat Apple doorgaans beperkte toegang verleent tot hun software, was het duidelijk dat de Apple Health-integratie waarschijnlijk uitdagender zou zijn. Ik heb deze kwestie besproken met mijn stagebegeleider en we hebben besloten om ons in eerste instantie te richten op de Google Fit API.

Ik ben vervolgens de stappen gaan volgen zoals beschreven in de Google Fit Rest API-handleiding. Ik heb een OAuth 2.0-client-ID aangevraagd en een project aangemaakt waarin ik de Fitness API heb ingeschakeld. Tijdens dit proces kwam ik erachter wat 'scopes' zijn, aangezien je bij het aanmaken van je project specifieke toegangsniveaus moet definiëren voor de gegevens die je project van een Google-account wil gebruiken. Ik heb alle Fit-gegevens die ik nodig heb geselecteerd. Het is belangrijk op te merken dat deze scopes beperkt zijn, wat betekent dat ze toegang verlenen tot gevoelige informatie, en ik niet zomaar toestemming krijg om deze gegevens te gebruiken. Daarom zal ik bezoekers van de webpagina moeten vragen om toegang tot deze gegevens te verlenen. Ik heb besloten om [Google Sign-In](#) te gebruiken, aangezien dit automatisch het autorisatieverzoek voor je afhandelt.

Daarnaast heb ik de Google Fit-app gedownload om alvast mijn eigen gegevens te verzamelen, zodat ik later kan testen of de gegevens die ik ophaal daadwerkelijk correct zijn.

3.2.2 Library Research – Community research

Op dit moment kan ik succesvol inloggen met mijn Google-account en ontvang ik de profielinformatie. Echter, nu sta ik voor de uitdaging om toegang te krijgen tot gegevens van de beperkte 'scopes'. Met behulp van Google Sign-In op mijn website ontvang ik na een succesvolle gebruikersaanmelding een toegangstoken. Nu wil ik graag weten hoe ik dit toegangstoken met een POST-verzoek naar de Google Fit API kan sturen, zodat ik de benodigde gegevens kan verkrijgen.

Ik heb geprobeerd informatie hierover te vinden, maar helaas kon ik niet veel relevante documentatie vinden. Daarom heb ik mijn [eigen vraag op Stack Overflow geplaatst](#). Helaas heb ik tot nu toe weinig reacties op mijn vraag ontvangen. Na uitgebreid zoeken ben ik echter gestuit op een [vraag van iemand anders](#) waarin nuttige informatie stond. Met deze informatie als leidraad ben ik begonnen met het maken van een GET-verzoek naar de Google API.

```
axios.get("https://www.googleapis.com/fitness/v1/users/me/dataSources/derived:com.google.step_count.delta:com.google.android.gms:estimated_steps/datasets/1694124000000000000-1694210400000000000")
```

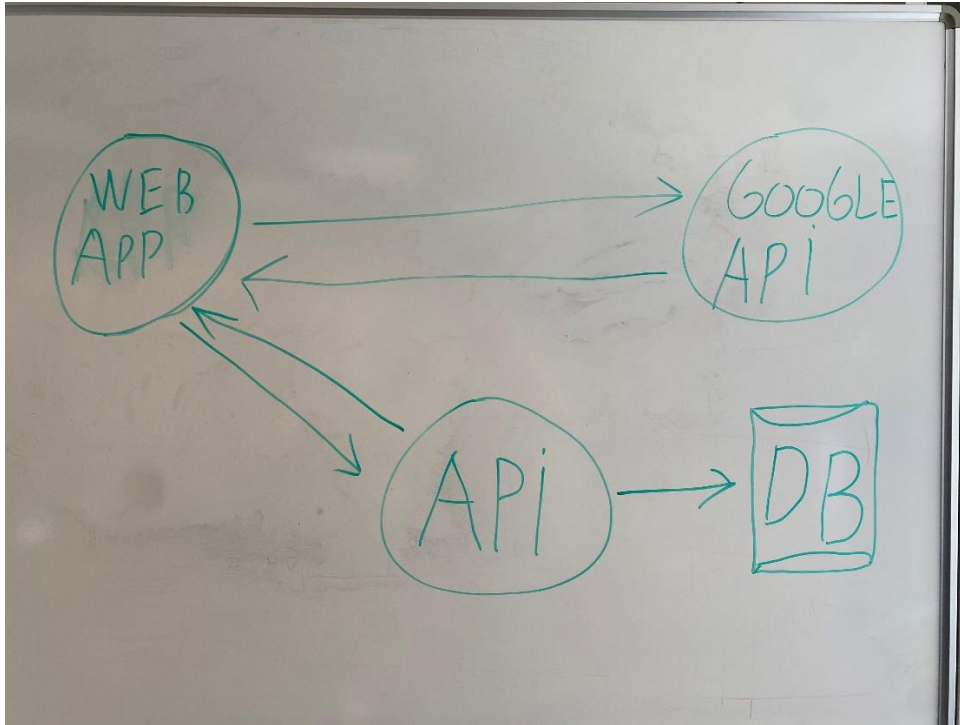
'me' verwijst naar de ingelogde gebruiker via Google. Vervolgens geef ik aan dat ik de gegevensbronnen van deze gebruiker wil bekijken, specifiek de geschatte stappentelling, en aan het einde geef ik met behulp van die nummers aan binnen welk tijdsbestek in milliseconden ik deze gegevens wil zien. Echter, ik bleef een '401 Unauthorized'-foutmelding ontvangen omdat ik geen token meestuurde. Toen herinnerde ik me dat ik tijdens mijn project in het derde semester van React een JWT-token moest meesturen, meestal via een 'Authorization'-header. Tot mijn verbazing werkte dit ook in dit geval.

```
function scopesRequest(){
  axios.get("https://www.googleapis.com/fitness/v1/users/me/dataSources/derived:com.google.step_count.delta:com.google.android.gms:estimated_steps/datasets/1694124000000000000-1694210400000000000", {
    headers: { Authorization: 'Bearer ' + token }
  })
  .then(function (response) {
    setStepRecords(response.data.point)
    calculateDailySteps();
  });
}
```

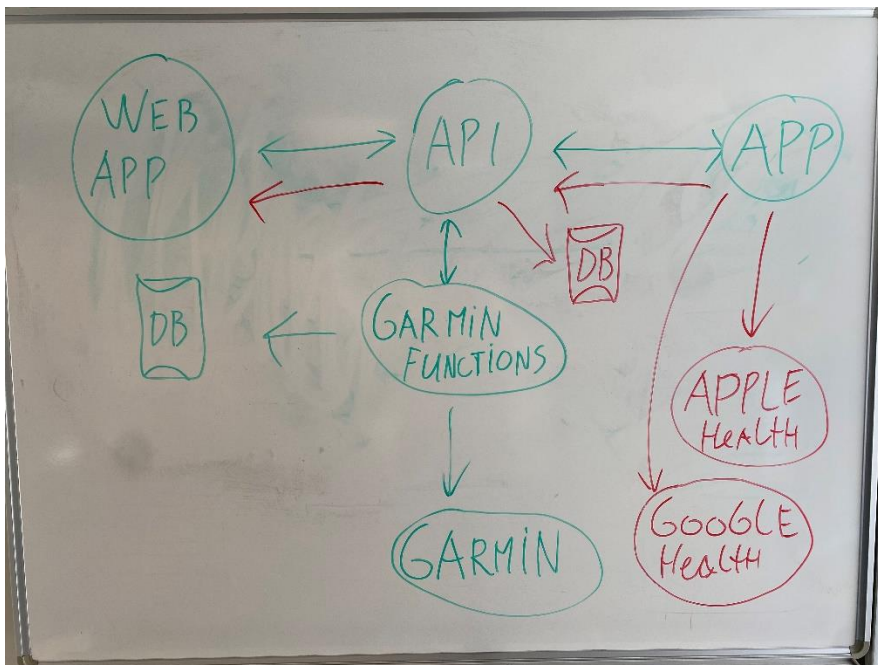
In eerste instantie riep ik de API aan en voerde ik berekeningen uit op de ontvangen gegevens om het totale aantal te verkrijgen. Later ontdekte ik echter dat je met behulp van een ander verzoek naar de [Google API](#) direct het totale aantal kunt verkrijgen.

3.2.3 Workshop Research - IT architecture sketching

Om een helder beeld te krijgen van de reikwijdte van het project, heb ik schetsen gemaakt op een whiteboard. Dit heeft me geholpen om duidelijkheid te krijgen over de aspecten waaraan ik zal werken, en het stelt mijn stagebegeleider ook in staat te zien dat ik het project begrijp. Dit is het systeem dat ik zal ontwikkelen in een testomgeving.



Dit is de gewenste uiteindelijke werking van het systeem.



3.2.4 Workshop Research – Prototyping

Mijn testomgeving zal dienen als een high-fidelity prototype waarin ik de mogelijkheden van de integratie kan verkennen en testen. Zodra ik tevreden ben met de staat van het prototype, zal ik het implementeren in het daadwerkelijke project

3.3 Qualiteit van het resultaat

Een prototype dat de Google API aanroept via de frontend-webpagina en vervolgens gegevens verzamelt.

3.4 Resultaat

Allereerst voer ik een POST-verzoek uit naar de Google Activity API en voer ik een aggregatie uit om het totaal te verkrijgen

```
function stepsRequest(){
    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": startTime,
        "endTimeMillis": endTime
    }, {
        headers: { Authorization: 'Bearer ' + token }
    })
    .then(function (response) {
        //console.log(response.data.bucket[0].dataset[0].point[0].value[0].intValue)
        setStepRecords(response.data.bucket[0].dataset[0].point[0].value[0].intValue)
    });
}
```

dataTypeName: Dit is de naam van het gegevenstype zoals te vinden op de Google API.

dataSourceId: Dit is de ID van de gegevensbron die je wilt verzamelen. Je kunt deze ID vinden door eerst een GET-verzoek uit te voeren om alle gegevensbronnen van een bepaald gegevenstype te bekijken.

```
axios.get("https://www.googleapis.com/fitness/v1/users/me/dataSources?dataTypeName=com.google.step_count.delta")
```

bucketByTime: Dit is de tijdsduur waarin je de gegevens wilt verzamelen.

startTimeMillis: Dit is het starttijdstip van de 'bucketByTime'.

endTimeMillis: Dit is het eindtijdstip van de 'bucketByTime'.

```
"bucketByTime": { "durationMillis": 86400000 },
                  "startTimeMillis": startTime,
                  "endTimeMillis": endTime
```

Vervolgens stuur ik de accesstoken mee met de autorisatie header

```
headers: { Authorization: 'Bearer ' + token }
```

en hierna sla ik het totaal wat ik terugkrijg van de api op in een locale variabele

```
setStepRecords(response.data.bucket[0].dataset[0].point[0].value[0].intVal)
```

Vervolgens controleer ik met behulp van een useEffect of de variabele is ingevuld.

```
useEffect(() => {
    if(stepRecords > 0){
        saveSteps();
    }
}, [stepRecords]);
```

Vervolgens stuur ik de data naar mijn API

```
function saveSteps(){
    axios.post('https://localhost:7212/api/Step', {
        DailySteps: stepRecords,
        StartTimeNanos: startTime,
        EndTimeNanos: endTime,
        userEmail: user.email
    })
    .catch(function (error) {
        console.log(error);
    });
}
```

4 Onderzoek 2

4.1 Vraag

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?

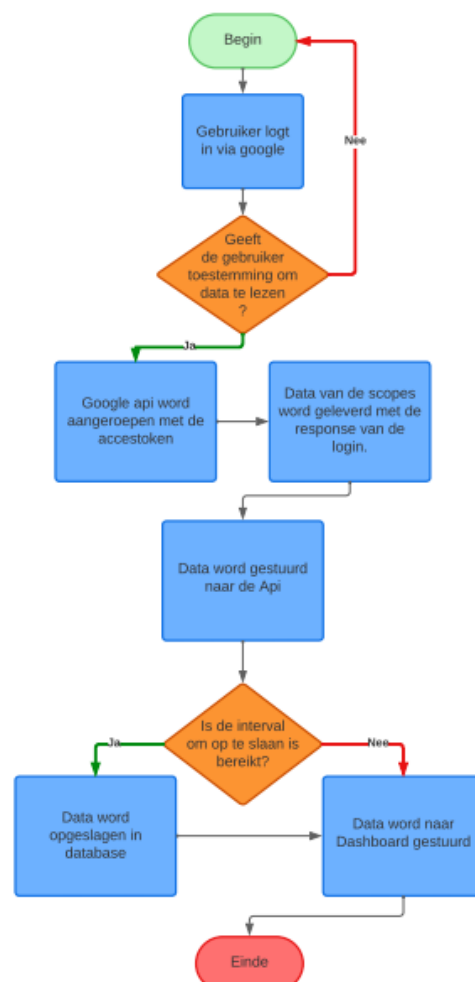
4.2 Hoe ben ik aan de slag gegaan?

4.2.1 Field Research – Problem analysis

Ik heb een gesprek gehad met mijn stagebegeleider en mijn collega Ruben, die heeft gewerkt aan het Move4Vitality-project. Zij hebben me uitgelegd hoe het huidige systeem werkt en waar mogelijke verbeteringen voor mijn project zich bevinden. Ruben heeft met mij de technische vereisten van het project besproken, waaronder de programmeertalen en benodigde software. In eerste instantie overwogen mijn stagebegeleider en Ruben om de Google API aan te roepen via mijn eigen API-backend, vergelijkbaar met de aanpak voor de Garmin-functionaliiteit. Echter, ik kwam met het idee om dit mogelijk via de frontend te realiseren. Hierdoor zouden we de extra Garmin API kunnen elimineren, omdat Garmin ook kan worden gekoppeld aan Apple en Google Health.

4.2.2 Field Research – Explore user requirements

Daarna heb ik een flowchart opgesteld om een visuele representatie te creëren van hoe gebruikers door de applicatie zouden navigeren.



Met behulp van deze flowchart heb ik een lijst van gebruikersverhalen samengesteld om de taken van elke gebruiker te definiëren. Ik heb vervolgens Planning Poker toegepast om een nauwkeurige schatting te maken van de complexiteit van deze functionaliteiten, zodat ik ze kon opnemen in mijn projectplanning. Daarnaast heb ik use cases ontwikkeld voor mijn gebruikersverhalen om te begrijpen hoe gebruikers in aanraking komen met deze functionaliteiten. Deze diagrammen en documentatie heb ik verwerkt in een Functioneel en Technisch ontwerp als resultaat

4.3 [Qualiteit van het resultaat](#)

Een functioneel & technisch ontwerp die de functionele en technische eisen van het project bevatten.

4.4 [Resultaat](#)

5 Onderzoek 3

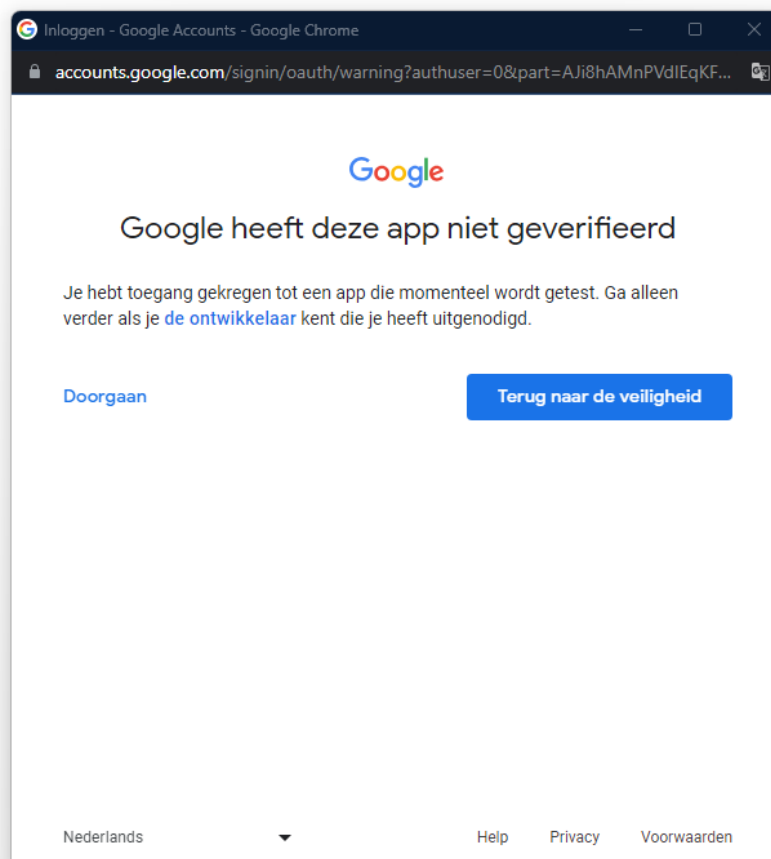
5.1 Vraag

Hoe moet omgegaan worden (uit technisch en regelgevings oogpunt) met privé/gezondheidsdata?

5.2 Hoe ben ik aan de slag gegaan?

5.2.1 Literature study – Library Research

Als eerste ben ik gaan onderzoeken hoe ik toegang kan krijgen tot de beveiligde gegevens van een Google-gebruiker via de Google API. Dit gebeurt door middel van het aanvragen van 'scopes'. Gezondheidsdata van een gebruiker valt onder de categorie 'restricted scope', wat betekent dat je de autorisatie van de gebruiker nodig hebt om deze data te kunnen lezen. Je kunt ook je applicatie laten verifiëren door Google. Om dit te doen, moet je applicatie voldoen aan het [Google Fit Developer and User Data Policy](#). Hierin staat beschreven hoe je moet omgaan met gevoelige gebruikersinformatie en welke beperkingen je moet naleven, zoals het niet aanvragen van scopes die je niet nodig hebt en het niet onnodig verzamelen van gebruikersgegevens. Als je applicatie niet geverifieerd is, kun je nog steeds gegevens ophalen, maar de gebruiker krijgt dan een beveiligingswaarschuwing te zien in je applicatie, zoals hieronder weergegeven:



5.2.2 Expert Interview – Library Research

Ik heb een lijst opgesteld met alle beschikbare activiteits- en gezondheidsgegevens die Google aanbiedt. Vervolgens heb ik hierover overlegd met mijn stagebegeleider om te bepalen welke gegevens ik daadwerkelijk in de applicatie ga gebruiken, met als doel het aantal benodigde 'scopes' te beperken tot wat strikt noodzakelijk is. Mijn stagebegeleider heeft me ook geadviseerd om een lijst te maken van de vereiste taken om de verificatie van Google voor de applicatie te verkrijgen.

5.2.3 Data Analytics – Lab Research

Ik heb ook contact opgenomen met Ruben om toegang te krijgen tot de gegevens van de huidige koppeling met Garmin. Dit geeft me inzicht in wat ze momenteel gebruiken om aan fysiotherapeuten te tonen en hoe ik deze gegevens eventueel in de toekomst kan integreren.

5.2.4 Explore User Requirements- Field Research

Ik ga in gesprek met mijn stagebegeleider, Ruben om duidelijkheid te krijgen over hun plannen voor een dergelijke applicatie. Dit zal me helpen bepalen hoe ik de gegevens wil verwerken en presenteren aan de fysiotherapeut.

5.3 Qualiteit van het resultaat

Een overzicht van de technische en functionele maatregelen die er genomen moeten worden om goed met gezondheidsdata om te kunnen gaan.

5.4 Resultaat