

Technisch Ontwerp

Walk Smarter – INF1C – Project Innovate

Code Crusaders

Jesse van der Voet – Elmedin Arifi – Mart Velema – Lars
Kuijer – Bram Suurd – Bryan Potze – Lucas Lübbers



Inhoudsopgave

Inleiding	3
Hardwarecomponenten	4
Softwareprocessen	5
Applicatie Architectuur	5
API Architectuur	7
ERD	9
Data flow diagram	10
Testen	12
Postman Testing	12
GitHub-flows	12
Physical Testing	12
Ontwikkelomgeving	13
Gebruikte software	13
Dataopslag	13

Inleiding

Dit document dient als een gedetailleerd overzicht van de “Walk Smarter” applicatie. Een innovatieve mobiele applicatie die is ontwikkeld voor zowel Android- als iOS-platformen. Het biedt inzicht in de architectuur, functionaliteiten, en technologieën die zijn toegepast om een consistente en hoogwaardige gebruikerservaring te garanderen, ongeacht het gekozen platform.

Hardwarecomponenten

Er wordt gebruikt gemaakt van een server voor het verwerken van API-calls van de app en het opslaan van gebruikersgegevens, zoals vrienden, groepen en punten. Dit zijn de minimale vereisten van de server:

- **Besturingssysteem:** Ubuntu Linux
- **Hardwarecomponenten:**
 - 4 core CPU
 - 8 GB RAM
 - 50 GB SSD-opslag

Softwareprocessen

Applicatie Architectuur

- Inloggen:
 - Gebruikersinvoer valideren: Het proces begint met het valideren van de gebruikersinvoer op het inlogschermb, waarbij wordt gecontroleerd of zowel de gebruikersnaam als het wachtwoordveld zijn ingevuld.
 - Authenticatie verwerken: Nadat de gebruiker de inloggegevens heeft ingevoerd en op de inlogknop heeft geklikt, wordt het authenticatieproces gestart. Dit omvat het verzenden van de ingevoerde gebruikersnaam en wachtwoord naar de backend van de applicatie voor verificatie.
 - Backend-authenticatie: Op de backend van de applicatie wordt de ingediende gebruikersnaam en wachtwoord gevalideerd met behulp van gegevens in de database. Als de ingediende gegevens correct zijn, wordt de gebruiker doorgestuurd naar zijn homepage.
 - Foutafhandeling: Als de ingediende gegevens ongeldig zijn (bijvoorbeeld als de gebruikersnaam niet bestaat of als het wachtwoord niet overeenkomt), wordt een foutmelding weergegeven op het inlogschermb om de gebruiker op de hoogte te stellen van het probleem.
- Registreren:
 - Gebruikersinvoer valideren: Het proces begint met het valideren van de gebruikersinvoer op het registratieschermb, waarbij wordt gecontroleerd of alle vereiste velden zijn ingevuld en of het ingevoerde e-mailadres een geldig formaat heeft.
 - Gebruikersgegevens verwerken: Nadat de gebruiker alle vereiste informatie heeft ingevoerd en op de registratieknop heeft geklikt, worden de ingevoerde gegevens naar de backend van de applicatie verzonden voor verwerking.
 - Backend-validatie: Op de backend worden de ingediende gegevens gevalideerd en wordt gecontroleerd of de gebruikersnaam en het e-mailadres uniek zijn. Als de gegevens geldig zijn, wordt een nieuw gebruikersaccount aangemaakt en wordt de gebruiker doorgestuurd naar het inlogschermb.
 - Foutafhandeling: Als er tijdens het registratieproces problemen optreden (bijvoorbeeld als de gebruikersnaam al in gebruik is), wordt een foutmelding weergegeven op het registratieschermb om de gebruiker op de hoogte te stellen van het probleem.
- Accountpagina:
 - Gegevens ophalen: Het proces begint met het ophalen van de gebruikersgegevens uit de database, zoals de gebruikersnaam, profielfoto, e-mailadres, enzovoort.
 - Weergave opmaken: Nadat de gegevens zijn opgehaald, wordt de accountpagina dynamisch opgemaakt om de

- gebruikersgegevens weer te geven in de juiste velden en opmaak.
- Interactieve functies toevoegen: Deze pagina kan interactieve functies bevatten zoals het wijzigen van profielfoto, het bijwerken van persoonlijke informatie, of het bekijken van eerdere activiteiten.
 - Doelen(achievement)-pagina:
 - Gegevens ophalen: De prestatiegegevens van de gebruiker worden uit de database opgehaald, inclusief voltooide doelen, behaalde niveaus, verdiende badges, enzovoort.
 - Weergave opmaken: Nadat de gegevens zijn opgehaald, wordt de achievementpagina dynamisch opgemaakt om de prestatiegegevens van de gebruiker weer te geven op een visueel aantrekkelijke manier.
 - Interactieve functies toevoegen: De gebruiker kan interactieve functies hebben zoals het delen van prestaties op sociale media, het bekijken van gedetailleerde statistieken over behaalde doelen en het instellen van nieuwe doelen.
 - Gamification-elementen: Deze pagina kan gamification-elementen bevatten zoals beloningen, badges, en punten om gebruikers te stimuleren om doelen te behalen en deel te nemen aan de applicatie.

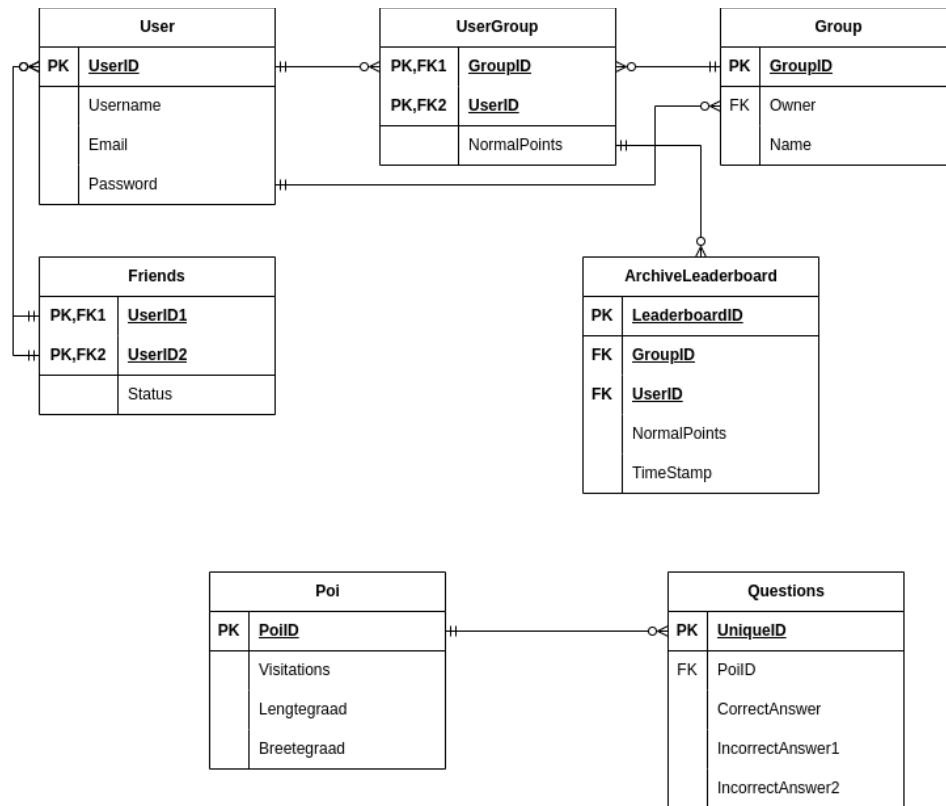
API Architectuur

Om de app werkend te maken, is er een API vereist op een externe server. Deze server verwerkt de API-calls en stuurt de output weer terug naar de app.

Voor onze applicatie moeten wij gebruik gaan maken van een zelf ontwikkelde API. De API gaat ervoor zorgen dat wij makkelijk en op een onderhoudbare manier met de data om kunnen gaan. Hier staan de belangrijkste processen beschreven.

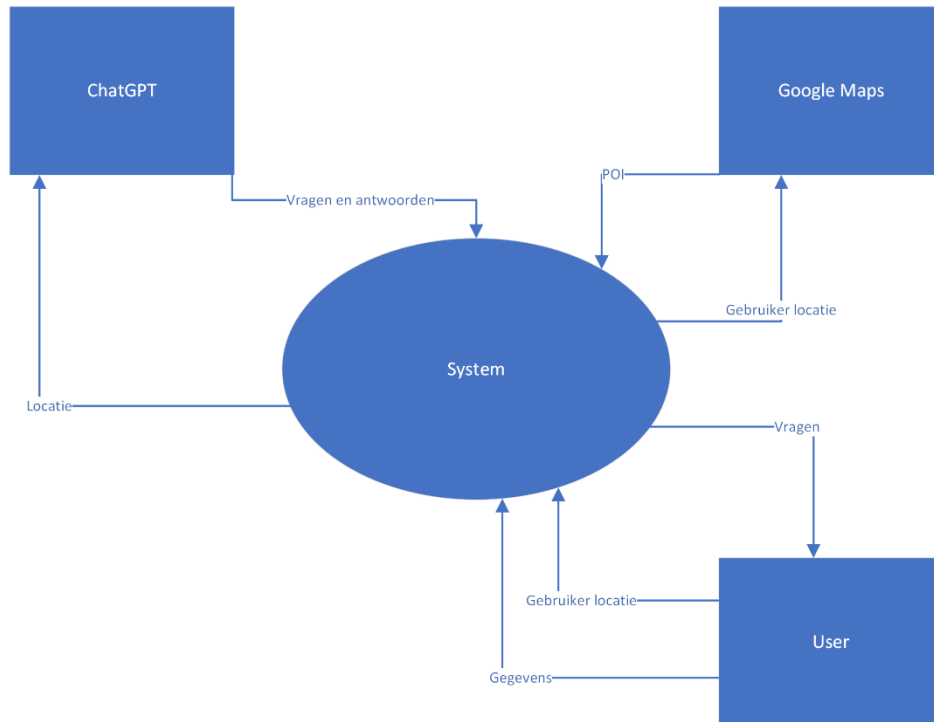
- Inlog API-call:
 - Wanneer een gebruiker wil inloggen moet een gebruiker de volgende data meegeven:
 - Gebruikersnaam
 - WachtwoordAls de gebruiker bestaat en het goede wachtwoord heeft ingevuld dan zal de API reageren met http code 200. In elk ander geval zal er een http code 403 teruggestuurd worden.
- Google Maps Service:
 - De Google Maps service data:
 - API Token
 - Longitude
 - LatitudeAls de Google Maps service een betrouwbare map en betrouwbaar points of interest (POI) wil laten zien dan stuurt de server een API-call naar de Google Maps service om vervolgens de map terug te sturen met bepaalde belangrijke markers.
- Point of interest questions service:
 - Wanneer een gebruiker dicht bij een POI is dan wordt er bij de server API een verzoek gemaakt naar de database. Dit vraagt om de volgende data:
 - Point of interest naam
 - Point of interest lengtegraad
 - Point of interest breedtegraad
 - De vraag
 - 1x correct antwoord
 - 2x verkeerd antwoordAls een POI geen vraag heeft opgeslagen in de database wordt er vanuit de server API een call gemaakt naar de Google Maps API om de POI's naam en coördinaten terug te krijgen. De server API stuurt de data vervolgens naar de OpenAI API die dan een vraag genereert met één goed antwoord en twee verkeerde antwoorden. De server API verwerkt de data in de database en stuurt vervolgens de vraag naar de applicatie toe.
- ChatGPT API vraag generatie:
 - Als een POI geen vraag heeft wordt er informatie naar de OpenAI GPT-4 API gestuurd. GPT-4 zoekt meer informatie op om op basis van die informatie educatieve vragen te creëren. Als de vraag gegenereerd is wordt die met een bepaald format teruggestuurd naar de server API om geformatteerd te worden voor de database.

ERD

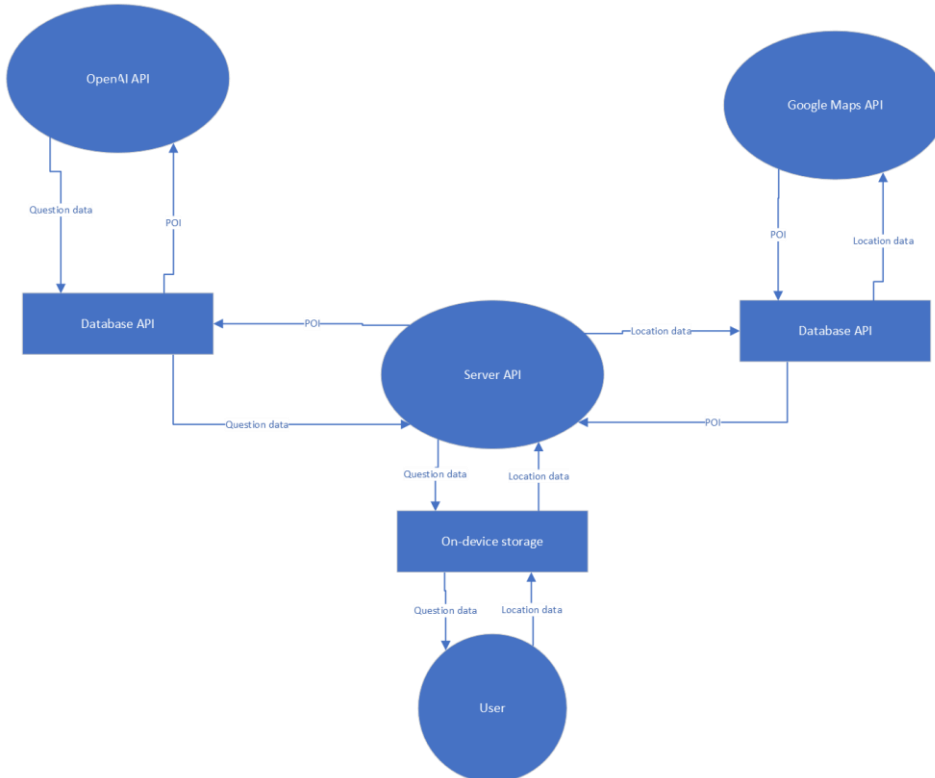


Data flow diagram

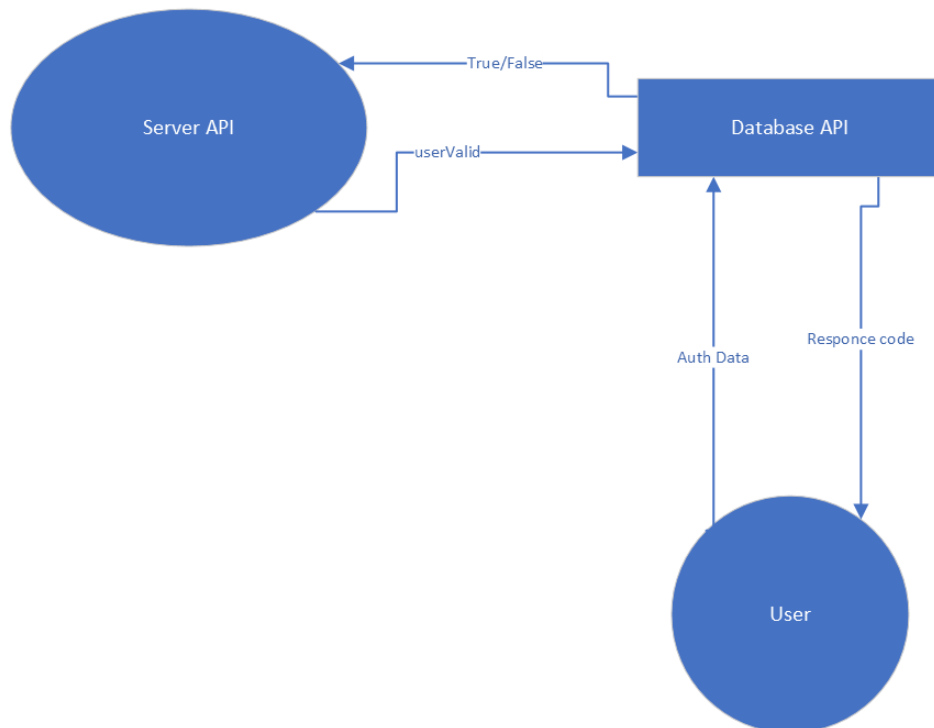
DFD-0 Algemene applicatie



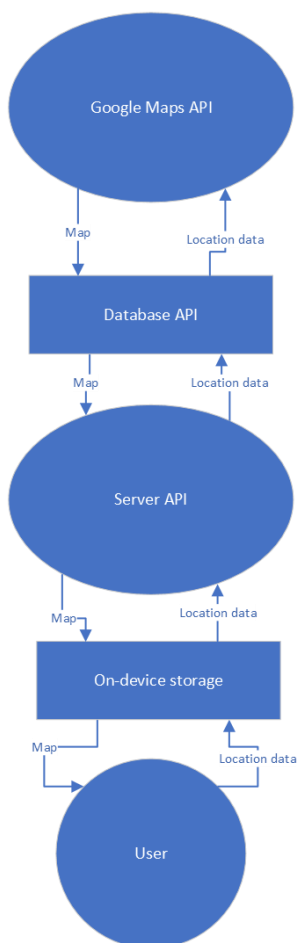
DFD-1 Vragen



DFD-1 Gebruiker authenticatie



DFD-1 Kaart locatie



Testen

Postman Testing

We zullen de applicatie uitgebreid testen met Postman om de API's te controleren op:

- Authenticatie
- Data-integriteit
- Responsiviteit

Dit zorgt ervoor dat de API's correct werken en aan de eisen voldoen.

GitHub-flows

We zullen GitHub-flows gebruiken om ons ontwikkelproces te stroomlijnen en de juiste codeconventies te handhaven, inclusief:

- Pull requests
- Code reviews om de codekwaliteit te waarborgen en fouten te minimaliseren.

Fysieke Testen

We zullen de applicatie uitgebreid testen door zelf gebruik te maken van de applicatie en naar verschillende points of interest te lopen. Wij controleren op:

- Correcte vragen bij POI's
- Accurate map van omgeving
- Bijhouden van punten

Ontwikkelomgeving

Gebruikte software

Voor het ontwikkelen van onze applicatie is er bepaalde software vereist. Hierin wordt kort uitgelegd welke software er nodig is.

Visual Studio Code: Visual Studio Code is een source-code editor die makkelijk te leren is en snel opgezet kan worden. Visual Studio Code is ideaal omdat het licht is, maar ook heel krachtig met het ontwikkelen van code. Visual Studio Code heeft overigens ook goede ondersteuning voor veel programmeertalen en extensies die de ontwikkelomgeving verbeteren.

PocketBase: PocketBase is een open-source backend-as-a-service (BaaS) met de belangrijkste functies zoals een real-time database, gebruikersverificatie en bestandsopslag. Het is vanwege het gebruiksgemak en de implementatie in één bestand ideaal voor snelle backends voor prototypes of kleine apps.

Android SDK: De Android SDK is nodig om op Android de applicatie te laten werken op Android gebaseerde apparaten.

Xcode (alleen op Apple apparaten): Xcode is de enige IDE (Integrated Development Environment) waarin de applicatie kan worden getest op IOS-apparaten.

Flutter SDK: De Flutter SDK is nodig om de applicatie op te bouwen.

Dart: Dart is de programmeertaal die wordt gebruikt in Flutter.

Dataopslag

GitHub URL: <https://github.com/NHL-P4-INF1C/WalkSmarter>

Git/GitHub: Wij gebruiken GitHub als primaire opslag bron vanwege de betrouwbaarheid en de schaalbaarheid van GitHub. Met GitHub wordt ervoor gezorgd dat iedereen op een gestructureerde manier zijn werk kan afmaken en opslaan.

PocketBase: PocketBase wordt gebruikt om data die nodig is op te slaan. Om API verzoeken te minimaliseren worden de vragen over POI's opgeslagen. Verder wordt PocketBase gebruikt om gebruikers te authenticiseren en om de data op te slaan van gebruikers, zoals hun score en groepen.

