

Graduaatsproef

LiftLab

Graduaat programmeren

Wojciech Karwinski
20 december 2023

Inhoud

Inleiding.....	2
Gebruikte Technologie	3
Angular	3
TypeScript	4
Capacitor	4
Android Studio	4
C#	5
Microsoft SQL Server	5
Aanpak en motivatie	6
Resultaat	7
Conclusie	11
Referenties	12

Inleiding

LiftLab is een handige mobile fitness tracking-applicatie, deze applicatie is ontworpen om gebruikers op een eenvoudige manier te helpen bij het plannen, organiseren en volgen van hun fitnessroutines. Zoals elke mobile app, bestaat deze uit het frontend, en de backend.

Frontend

De frontend is ontwikkeld met **Angular**, wat zorgt voor een gebruiksvriendelijke en responsieve gebruikersinterface.

Voor de ontwikkeling en het testen op android smartphones is gebruik gemaakt van **Capacitor** en **Android Studio**.

Deze tools helpen de app te laten werken op verschillende mobiele toestellen, waardoor een optimale gebruikerservaring wordt gegarandeerd.

Backend

De backend is ontwikkeld met **C#** en gebruikt **Microsoft SQL Server** voor de opslag van gegevens.

Deze combinatie zorgt voor een efficiënte verwerking en veilige opslag van alle informatie met betrekking tot trainingen en oefeningen.

De kracht van C# in de backend maakt LiftLab betrouwbaar en robuust voor gegevensbeheer.

Gebruikte Technologie

Angular

Angular is een framework dat gebaseerd is op componenten en heeft een boomstructuur. Dit betekent dat een applicatie bestaat uit verschillende componenten. De hoofdcomponent van elke Angular-applicatie is de 'root'-component, die als primaire component dient. Binnen deze centrale component zijn 'child'-components te vinden, elk toegewezen aan een bepaald segment van de applicatie. Deze opstelling lijkt op een boom, waarbij de 'root' dient als de stam en de 'child'-componenten als takken. Parent components hebben de mogelijkheid om data te 'binden' en deze door te geven aan hun kinderen. Echter, de child-components weten niet direct waar deze gegevens vandaan komen. Als een kindcomponent gegevens terug wil sturen naar een oudercomponent, gebeurt dit via een 'emit' event. Dit maakt de componenten herbruikbaar, wat het programmeren van grote applicaties veel efficiënter maakt.

In Angular wordt voornamelijk gebruikgemaakt van **HTML** en **TypeScript**. HTML wordt gebruikt voor het ontwerp van de applicatie, terwijl TypeScript verantwoordelijk is voor de logica en de functionaliteit.

In elke applicatie die met Angular wordt gemaakt, ligt de kern binnen de Angular **components**. Deze componenten zijn van enorm belang omdat ze dienen als basis voor het definiëren van 'views'. In de context van Angular betekent een 'view' een verzameling elementen die zorgvuldig zijn uitgekozen en weergegeven door Angular, volgens de vooraf vastgestelde logica van de applicatie. Ze kunnen reageren op gebruikersinteracties en aanpassingen in de achterliggende logica van het programma, wat zorgt voor een interactieve en dynamische gebruikerservaring.

TypeScript

TypeScript is een programmeertaal ontwikkeld door Microsoft, die JavaScript uitbreidt met statische types, klassen, interfaces, ... Deze types helpen programmeurs om fouten te vermijden en maken de code duidelijker en beter onderhoudbaar. Tijdens het compileren wordt TypeScript omgezet in JavaScript zodat het in webbrowsers kan werken.

Angular gebruikt TypeScript om programmeurs te helpen bij het maken van krachtige en goed gestructureerde applicaties. TypeScript voor duidelijke structuren en principes van OOP, wat leidt tot betere kwaliteit en onderhoudsgemak van de code. Ook helpt TypeScript bij het vinden van fouten tijdens het ontwikkelproces, wat zorgt voor een efficiëntere en effectievere ontwikkelingsaanpak.

Capacitor

Capacitor is een bekende bibliotheek die webtechnologieën probeert te verbinden met native software development kits (SDK's). Het doel is om ontwikkelaars in staat te stellen één keer code te schrijven en deze eenvoudig te implementeren op verschillende platforms, waaronder iOS, Android en webapplicaties. Door Capacitor te gebruiken wordt applicatieontwikkeling efficiënter en eenvoudiger.

Android Studio

Android Studio is een IDE (Integrated Development Environment) die specifiek is ontworpen voor het bouwen van Android-applicaties, gebaseerd op IntelliJ IDEA. Android Studio helpt niet alleen bij het schrijven van code, zoals IntelliJ, maar biedt ook meer functies om uw productiviteit te verhogen bij het maken van Android-apps, bijvoorbeeld:

- Een flexibel systeem om verschillende versies van je app te beheren met behulp van Gradle.
- Een snelle en uitgebreide simulator waarmee je kunt zien hoe je app eruitziet en werkt zonder dat je deze op een echt apparaat hoeft te installeren.
- Uniforme omgeving voor ontwikkeling voor alle Android-apparaten, zoals telefoons, tablets en smartwatches.
- Instant Run om wijzigingen in actieve apps door te voeren zonder een nieuwe APK te maken

C#

C# is een object-georiënteerde programmeertaal die gebruikt kan worden om applicaties te maken die draaien in de .NET. Deze taal, ontwikkeld door Microsoft, is een evolutie van de C en C++ talen en is speciaal gemaakt om goed te werken met het .NET platform. C# is veelzijdig en kan gebruikt worden voor verschillende soorten applicaties, zoals Windows-applicaties, webapplicaties en webservices. Omdat C# een relatief recente taal is, bevat het veel functies die zijn overgenomen van andere talen, terwijl veel van hun problemen zijn opgelost. Met een goede balans tussen kracht en gebruiksgemak, is C# een sterke keuze voor moderne softwareontwikkeling.

Enkele belangrijke kenmerken van C# zijn:

- LINQ (Language Integrated Query): Stelt ontwikkelaars in staat om data queries direct in C# te schrijven, wat het werken met gegevens vereenvoudigt.
- Asynchrone Programmering (async/await): Vereenvoudigt het schrijven van asynchrone code, wat resulteert in betere prestaties en responsiviteit van applicaties.
- Lambda Expressies en Anonieme Functies: Zorgen voor kortere, meer leesbare code door het gebruik van beknopte functie-definities.
- Eigenschappen (Properties): Bieden een veilige manier om toegang te krijgen tot en te manipuleren van klassegegevens, wat leidt tot betere encapsulatie.
- Delegates en Events: Vergemakkelijken een losgekoppelde manier van event handling en communicatie tussen objecten.

Microsoft SQL Server

SQL is de afkorting voor Structured Query Language. Het is de taal die gebruikt wordt om te communiceren met databaseservers voor het opslaan, manipuleren en ophalen van gegevens die zijn opgeslagen in relationele databases. Relational Database Management Systems (RDBMS), zoals Oracle, Microsoft SQL Server, Microsoft Access, MySQL, ... maken gebruik van SQL. De gegevens in deze databases zijn gestructureerd in tabellen met rijen en kolommen. Elk systeem heeft zijn eigen versie van SQL met extra functies, maar ze vertonen allemaal een grote gelijkenis. Verschillende versies van SQL zijn bijvoorbeeld PL/SQL ontwikkeld en gebruikt door Oracle en T-SQL ontwikkeld en gebruikt door MS SQL Server enz. SQL speelt een belangrijke rol bij het mogelijk maken van veel dingen voor gebruikers om gegevens te beheren die zijn opgeslagen in relationeel databasebeheer systemen. Met SQL kunnen gebruikers gegevens in relationele databases opvragen, beschrijven en wijzigen. Ze kunnen ook databases en tabellen aanmaken of verwijderen, rechten instellen en nieuwe weergaven of opgeslagen procedures in de database maken.

Aanpak en motivatie

Mijn interesse in fitness heeft al lang een grote rol gespeeld in mijn leven. Deze passie was dan ook de drijvende kracht achter mijn beslissing om een fitnessapplicatie te ontwikkelen voor mijn graduaatsproef. Het idee om mijn hobby te combineren met mijn studie was voor mij een logische stap.

Ik begon mijn project met het opstellen van een de businessmodellen. Dit hielp me om mijn ideeën te structureren en een duidelijk plan van aanpak te hebben. Het was essentieel om een solide basis te leggen voordat ik met de meer technische aspecten van het project aan de slag ging.

Vervolgens focuste ik me op de data laag, waarvoor ik ADO.NET gebruikte. Deze stap was cruciaal om een efficiënte en goed georganiseerde databank op te zetten. Na het opzetten van de databank ontwikkelde ik een REST API, die als een interface tussen de databank en de frontend van de applicatie diende.

Mijn volgende stap was het leren van TypeScript. Als een uitbreiding van JavaScript bood het de functionaliteit die ik nodig had voor mijn project. Het aanleren van TypeScript was een interessante en uitdagende ervaring, die me goed van pas kwam bij de latere ontwikkeling van de frontend.

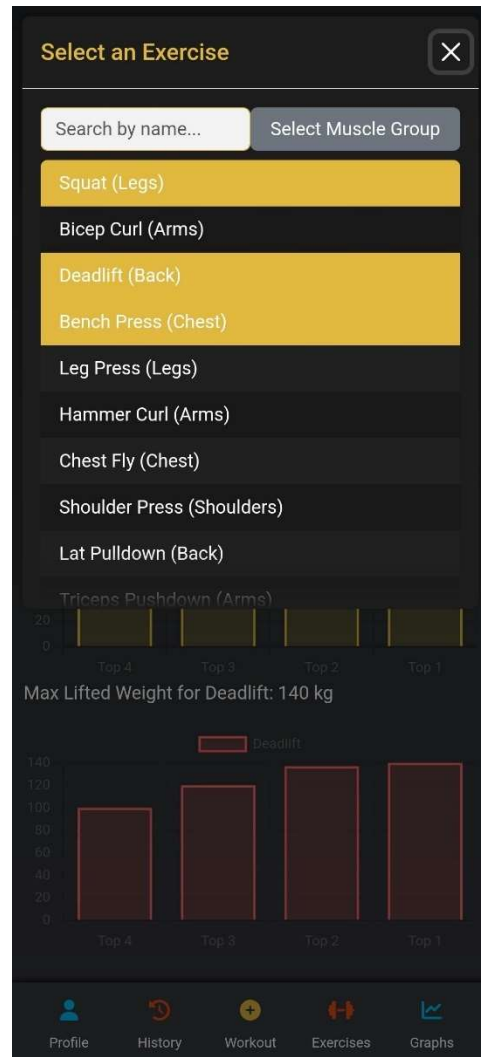
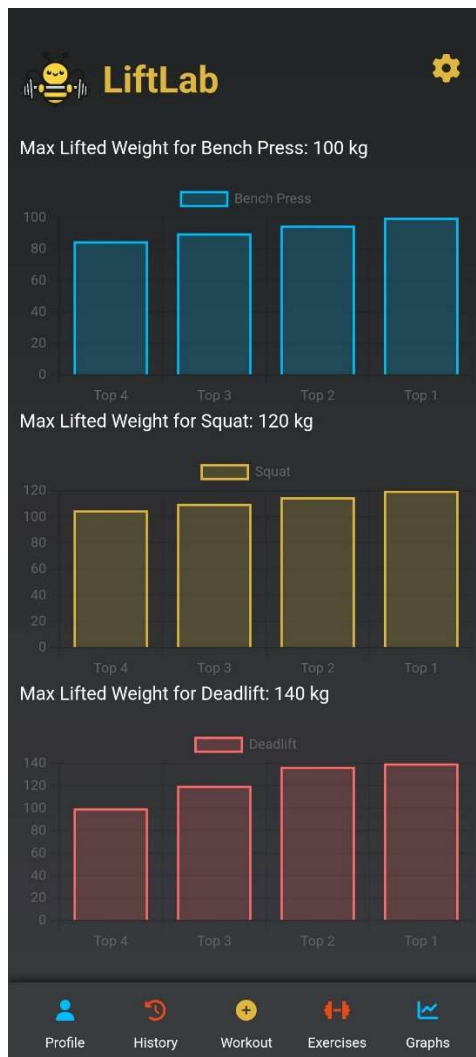
Uiteindelijk heb ik me gericht op Angular voor het ontwikkelen van de frontend. Het was echt interessant om te zien hoe mijn applicatie stap voor stap tot leven kwam. Het proces van het ontwikkelen van de gebruikersinterface was zowel creatief als technisch uitdagend.

De keuze voor dit project kwam voort uit mijn persoonlijke interesse in fitness en mijn wens om mijn vaardigheden in softwareontwikkeling te verdiepen. Voordat ik aan dit project begon, was frontend ontwikkeling niet mijn sterkste kant. Dit project was daarom een uitgelezen kans om mijzelf te ontwikkelen en te verbeteren in dit gebied. Het ontwikkelen van deze applicatie was een manier om mijn passie voor fitness te integreren met mijn studie, wat het project voor mij extra bijzonder maakte. Door me te verdiepen in Angular en TypeScript, kon ik mijn vaardigheden uitbreiden en versterken, wat een waardevolle ervaring was.

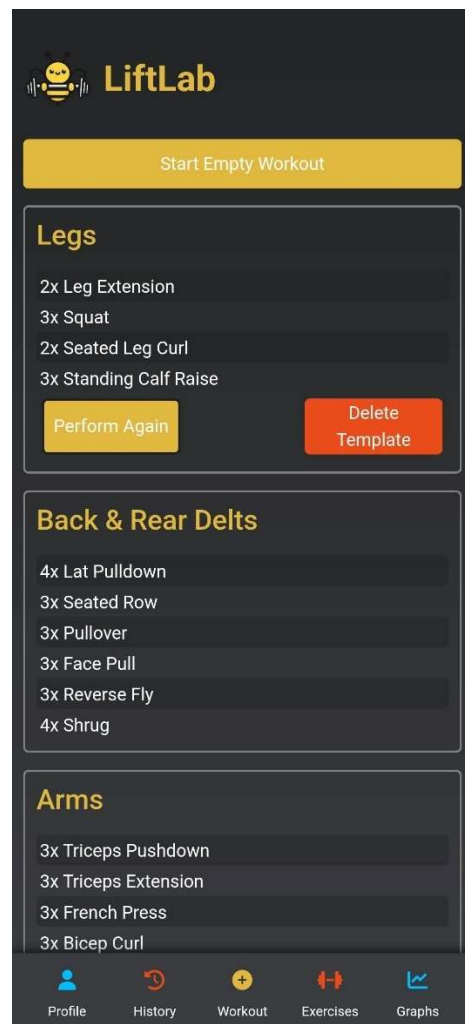
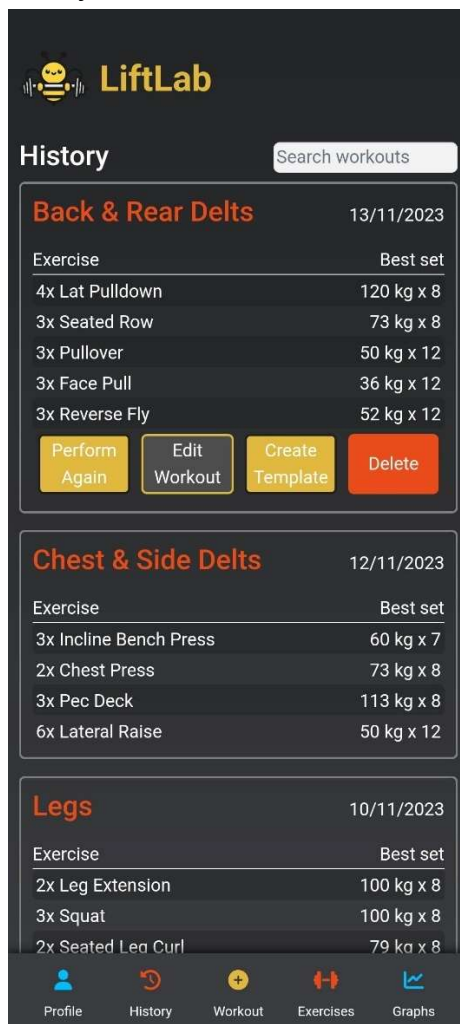
Resultaat

Het resultaat van het project is een grotendeels voltooide mobiele applicatie met diverse functionaliteiten.

Op de **profielpagina** worden drie grafieken gepresenteerd. Gebruikers kunnen hier kiezen uit hun oefeningen om de topsets weer te geven, wat inzicht biedt in hun beste prestaties. Deze functie stimuleert gebruikers om hun voortgang te volgen en te streven naar verbetering.



De **geschiedenispagina** toont een overzicht van de voltooide workouts. Hier kunnen gebruikers ervoor kiezen om een eerder voltooide workout opnieuw uit te voeren, een bestaande workout te bewerken, een nieuwe workout template te maken of een workout te verwijderen. ↓



Op de workout-pagina worden verschillende workout-templates getoond. Gebruikers kunnen vanuit deze pagina een lege workout starten of kiezen voor een vooraf gemaakte template.

Tijdens een workout kunnen gebruikers een pauze-timer starten, elke set van een oefening aanpassen, zoals het gewicht, het aantal herhalingen, de RPE (Rate of Perceived Exertion) en aangeven of de set is voltooid. Het is ook mogelijk om sets of oefeningen te verwijderen of extra sets of oefeningen toe te voegen. Zodra een workout is voltooid, wordt deze direct weergegeven op de geschiedenispagina

⌚ 00:00

FINISH

Back & Rear Delts :

Lat Pulldown

SET	KG	REPS	RPE	
1	60	12	<div></div>	<input checked="" type="checkbox"/>
2	80	12	<div></div>	<input checked="" type="checkbox"/>
3	100	10	<div></div>	<input checked="" type="checkbox"/>
4	120	8	<div></div>	<input type="checkbox"/>

Add Set

Seated Row

SET	KG	REPS	RPE	
1	59	8	<div></div>	<input type="checkbox"/>
2	66	8	<div></div>	<input type="checkbox"/>
3	73	8	<div></div>	<input type="checkbox"/>

Select an Exercise

Search by name... Select Muscle Group

Push-up (Chest)

Pull-up (Back)

Squat (Legs)

Bicep Curl (Arms)

Plank (Core)

Deadlift (Back)

Lunges (Legs)

Tricep Dip (Arms)

Sit-up (Core)

Bench Press (Chest)

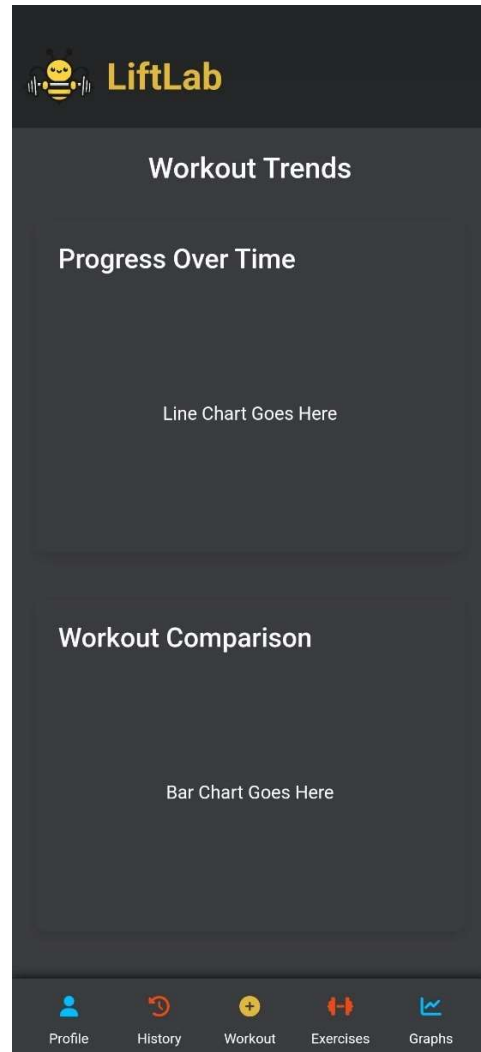
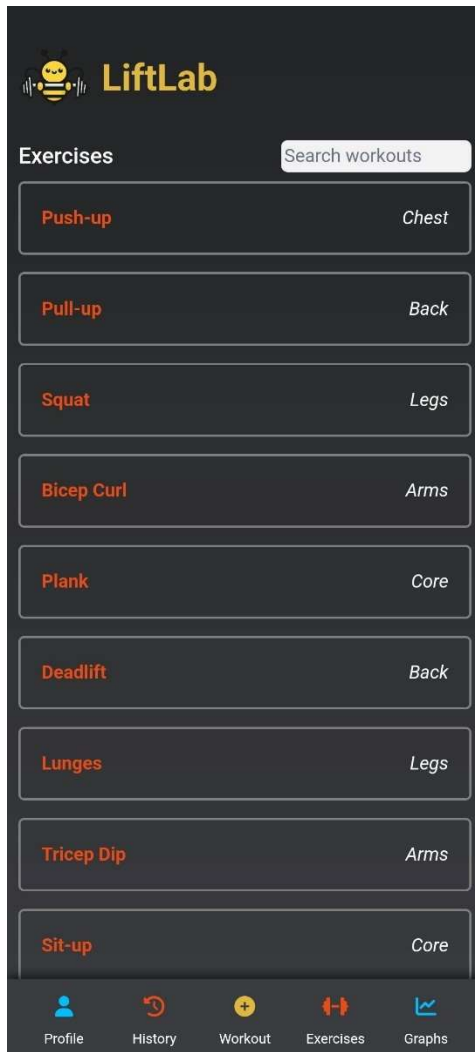
2	52	12	<div></div>	<input checked="" type="checkbox"/>
3	52	12	<div></div>	<input checked="" type="checkbox"/>

Add Set

Add Exercise

Cancel Workout

Op de exercises- pagina wordt een uitgebreide lijst van beschikbare oefeningen gepresenteerd. Een van de geplande, maar niet gerealiseerde functies, was het toevoegen van zelfgemaakte oefeningen. ↓



Hoewel de grafiekenpagina niet voltooid is, was het oorspronkelijke doel om geavanceerde analysefuncties te bieden, zoals het maken van voorspellingen en het vergelijken van verschillende workouts, evenals het bijhouden van de voortgang over tijd.

Conclusie

De ontwikkeling van de LiftLab-applicatie was een leerrijke en uitdagende ervaring. Het project combineerde mijn persoonlijke interesse in fitness met mijn passie voor softwareontwikkeling, wat resulteerde in een uniek en functioneel product. Door het bouwen van deze applicatie heb ik waardevolle vaardigheden en kennis opgedaan, vooral in Angular en TypeScript, en heb ik ervaring opgedaan met het testen van software met tools zoals Capacitor en Android Studio.

Hoewel niet alle geplande functies zijn voltooid, zoals de grafiekenpagina, zijn de resultaten van dit project toch bevredigend. De applicatie bevat nu een profielpagina, een gedetailleerde geschiedenis van workouts, een workout-menu en een oefeningenpagina. Deze onderdelen bieden een solide basis voor gebruikers om hun trainingen effectief te plannen en te analyseren.

De grootste uitdaging lag in het ontwikkelen van de frontend, aangezien dit niet mijn sterkste kant was. Dit project bood echter een uitstekende kans om deze vaardigheden te ontwikkelen en mijn begrip van gebruikersinterface-ontwerp te verbeteren.

Al met al was het proces van het ontwikkelen van de LiftLab-applicatie zowel verrijkend als uitdagend. Het project heeft niet alleen mijn technische vaardigheden verbeterd, maar ook mijn vermogen om complexe problemen op te lossen en zelfstandig een project te beheren. Dit vormt een waardevolle basis voor mijn toekomstige carrière in softwareontwikkeling.

Referenties

https://docs.oracle.com/cd/E10405_01/appdev.120/e10379/ss_oracle_compared.htm

<https://angular.io/guide/architecture>

<https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver16>

<https://learn.microsoft.com/en-us/dotnet/>

<https://www.typescriptlang.org/docs/>