Nicky Zakrzewski

3ICT



Fuel4less

Mobiele Apps 2: Kotlin

Inhoud

[Oogpunt van de gebruiker 2](#_Toc156310896)

[Wat is de Fuel4Less app? 2](#_Toc156310897)

[De home pagina 2](#_Toc156310898)

[De details pagina 3](#_Toc156310899)

[De navigatiebalk 3](#_Toc156310900)

[De login pagina 4](#_Toc156310901)

[De favorietenpagina 4](#_Toc156310902)

[Technische werking 5](#_Toc156310903)

[Models 5](#_Toc156310904)

[Firebase Realtime Database 5](#_Toc156310905)

[GasStation data klasse 5](#_Toc156310906)

[GasStationRepository 6](#_Toc156310907)

[ViewModels & Adapters 7](#_Toc156310908)

[GasStationViewModel 7](#_Toc156310909)

[GasStationAdapter 8](#_Toc156310910)

[Views 10](#_Toc156310911)

[MainActivity 10](#_Toc156310912)

[Home fragment 12](#_Toc156310913)

# Oogpunt van de gebruiker

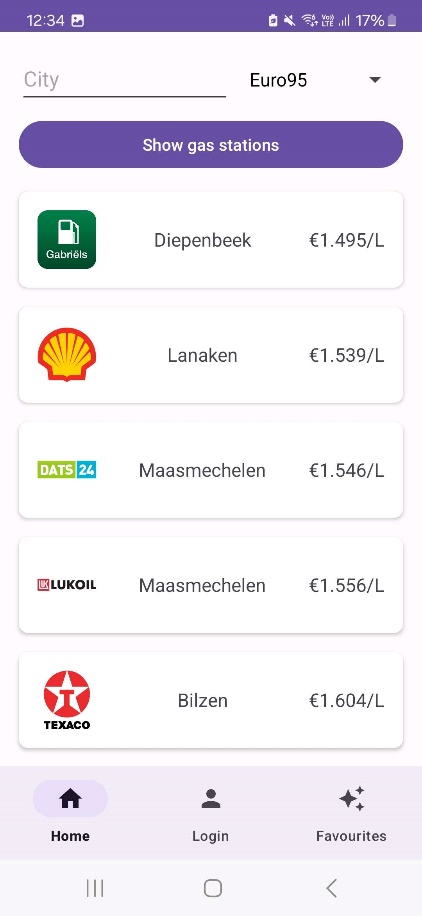
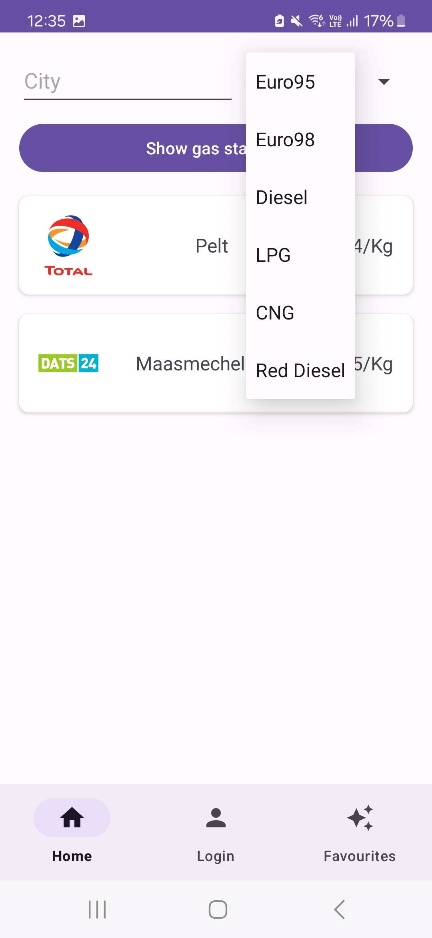
## Wat is de Fuel4Less app?

De Brandstof app is een applicatie waarmee de gebruiker brandstofprijzen kan vergelijken. Men kan filteren op het brandstoftype en op plaatsnaam, zodat enkel relevante tankstations worden getoond. De app bestaat uit 3 hoofdpagina’s, namelijk de homepagina, de login pagina en de favorieten pagina. De homepagina toont een lijst van tankstations, de login pagina zorgt ervoor dat de gebruiker zich kan inloggen op zijn of haar account en de favorieten pagina toont opgeslagen tankstations. Als er op een tankstation van de lijst wordt getikt, kan men een gedetailleerde weergave zien op de details pagina.

## De homepagina

Om van start te gaan, moet de gebruiker de applicatie downloaden en installeren. Eens de gebruiker de app geïnstalleerd heeft, is de app meteen te herkennen aan het icoontje met een tankstation pomp. Op deze manier is het makkelijker om de app te onderscheiden en zo weet de gebruiker ook onmiddellijk dat de app gerelateerd is aan brandstof.

Op de foto links is het uiterlijk van de homepagina te zien wanneer de app wordt opgestart. Vanboven bevinden zich een zoekbalk, een dropdownlist en een zoekknop. In de zoekbalk kan men een plaatsnaam invoeren en bij de dropdownlist menu kan er een brandstoftype worden gekozen. Als de filters ingesteld zijn, kan de gebruiker op de “Show gas stations” knop drukken om vervolgens een lijst van tankstations te zien die aan de filters voldoen.

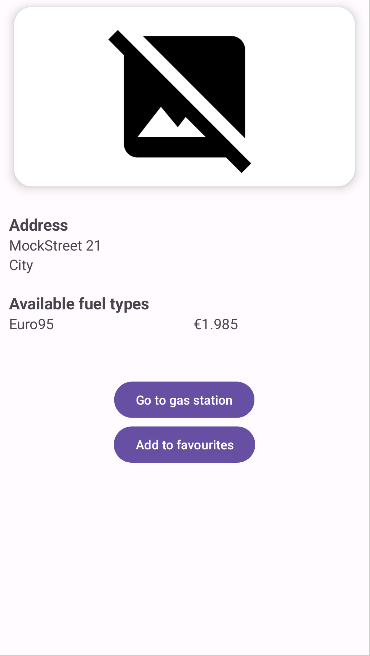
In de dropdownlist kan men kiezen uit de volgende brandstoftypes: Euro95 (E10), Euro98 (E5), Diesel (B7), LPG, CNG en Red Diesel (mazout). De dropdownlist staat standaard altijd op de eerste positie, namelijk Euro95.



## De details pagina

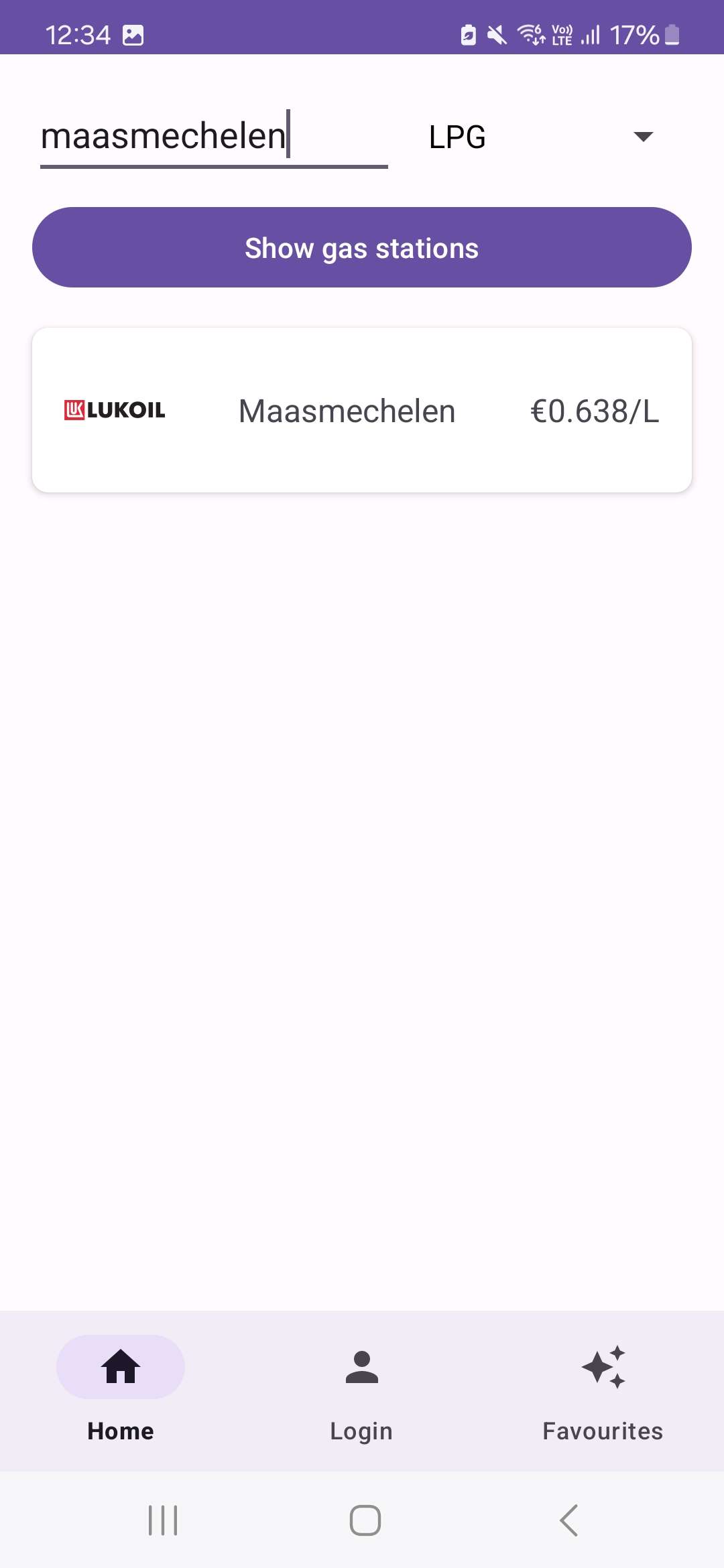
Wanneer de gebruiker een tankstation tegenkomt waar die in geïnteresseerd is, kan hij/zij drukken op het item uit de lijst. Nadat er op het item is geklikt, wordt er een gedetailleerde weergave getoond van het tankstation. Vanboven ziet men het logo en onder het logo volgt bijkomende informatie, zoals het adres van het tankstation en alle beschikbare brandstoffen (met de prijzen erbij).

Helemaal van onder zijn er ook nog 2 knoppen, de “Go to gas station” knop en de “Add to favourites” knop. De eerste knop opent de lokale navigatie app en stelt een route in naar het adres van het tankstation. De tweede knop voegt het tankstation toe aan de lijst van favoriete tankstations, zodat deze gemakkelijk geraadpleegd kan worden. Deze pagina is een potentiële uitbreiding naar de toekomst toe.



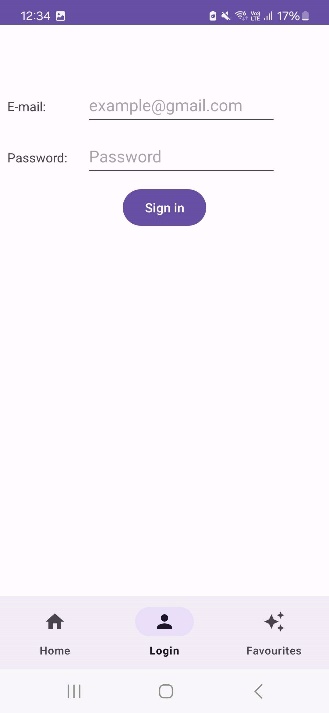
## De navigatiebalk

Onderaan kan men de navigatiebalk zien. Deze is altijd zichtbaar, ongeacht de pagina waar de gebruiker zich op bevindt. In deze navigatiebalk zijn er 3 opties, namelijk de homepagina, de loginpagina en de favorietenpagina. Er kan van pagina worden gewisseld door op het icoontje van de pagina te drukken.



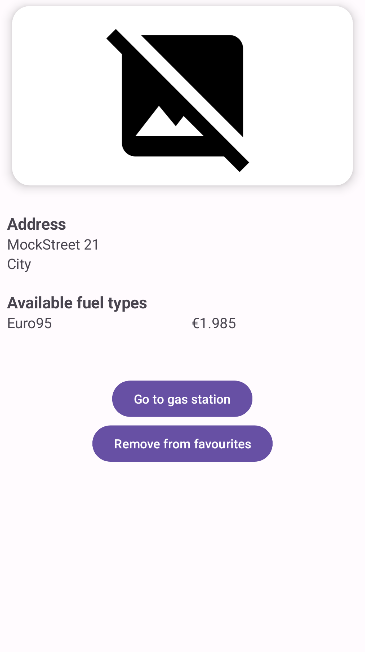
## De login pagina

De loginpagina geeft de mogelijkheid om een email adres en een wachtwoord in te voeren. Als er dan op de “sign in” knop wordt gedrukt, is de gebruiker ingelogd en kan die zijn/haar opgeslagen tankstations raadplegen. Mogelijke uitbreidingsfuncties naar de toekomst toe zouden bijvoorbeeld accountsinstellingen kunnen zijn, waardoor de login pagina een accountpagina wordt.



## De favorietenpagina

De favorieten pagina lijkt veel op de homepagina, maar bevat geen filters en zoekknop. Men kan opgeslagen tankstations raadplegen in de vorm van een lijst. Als er op een tankstation wordt getikt, komt de details pagina tevoorschijn met de gegevens van het gekozen tankstation. De “Add to favourites” knop wordt vervangen door de “Remove from favourites” knop. Deze pagina is ook een potentiële uitbreiding.



# Technische werking

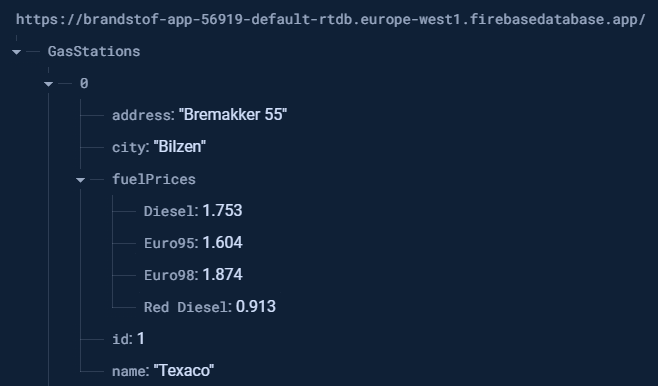
De applicatie is gemaakt volgens het MVVM-model, wat staat voor Model, View en ViewModel. Dit is de gebruikte structuur die de UI scheidt van de business logica.

## Models

### Firebase Realtime Database

Fuel4Less maakt gebruik van een Firebase Realtime Database. Het originele idee was om een API te gebruiken die brandstofprijzen van verschillende tankstations laat zien, maar er was geen beschikbare API die gratis is. Het alternatief is dus een database met statische gegevens. De Firebase Realtime Database is opgebouwd in de vorm van een JSON-formaat, dit is dus geen klassieke SQL-table omgeving. Doordat dit een JSON-formaat heeft, lijkt het ook alsof er gebruik wordt gemaakt van een API.

De structuur van de database bestaat uit een “tabel” genaamd GasStations. In deze tabel zitten JSON-objecten. Deze objecten bevatten de eigenschappen id, name, city, address en fuelPrices. Het id is een integer dat de “gegevensrij” identificeert. Elk tankstation heeft zijn eigen id. Bij name, city en address zijn de gegevens vanzelfsprekend en van het type string. De kolom fuelPrices is een associatieve array, waarbij het brandstoftype (string) de sleutel is en de prijs (double) het gegeven is. Dit is altijd de prijs per liter, behalve bij CNG. Daar is het de prijs per kilogram.



### GasStation dataklasse

De items van de database worden na ontvangst omgezet in objecten van het type GasStation. Deze objecten bevatten de nodige properties van de JSON-objecten zodat het omzetten makkelijker verloopt. FuelPrices is een mutableMap waarin het brandstoftype de sleutel is en de prijs/liter de data is.

Afbeelding met tekst, Lettertype, lijn, schermopname

Automatisch gegenereerde beschrijving

### GasStationRepository

De repository is waar de gegevens uit de database worden opgehaald en beschikbaar gesteld aan de rest van de applicatie. Eerst wordt er een database reference gemaakt zodat er een instance gemaakt wordt van de database. Omdat dit in JSON-formaat is, wordt er gevraagd om de children van GasStations. Daarnaast is er ook nog de gasStations variabele, waarin alle tankstations worden opgeslagen in de vorm van een lijst bestaande uit GasStation objecten. Bij de initialisatie wordt de functie loadGasStations opgeroepen. Deze functie is verantwoordelijk voor het laden van gegevens van gasstations vanuit de database. Ten slotte is er nog de getAllGasStations functie, die wordt geërfd van de IGasStationRepository interface. Deze functie geeft simpelweg de lijst van alle GasStation objecten terug.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

## ViewModels & Adapters

### GasStationViewModel

Deze ViewModel haalt alle gegevens op uit de repository met behulp van de getAllGasStations functie. Er komen 2 nieuwe functies bij, namelijk de getGasStationByFuelType en getGasStationsByFuelTypeAndCity. Deze tweede functie wordt aangeroepen vanuit de home fragment om alle gefilterde tankstations op te halen. Als er geen plaatsnaam is ingegeven in de zoekbalk, wordt automatisch de eerste functie opgeroepen en worden de tankstations enkel gefilterd op brandstof. Als er wel een plaatsnaam is ingevoerd, dan wordt er gefilterd op plaatsnaam en brandstof. Vervolgens wordt er een lijst teruggestuurd.

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

GasStationViewModelFactory

De GasStationViewModelFactory is een factory klassse die wordt gebruikt om een instantie van GasStationViewModel te maken. Het accepteert IGasStationRepository als argument in de constructor. Wanneer een nieuw ViewModel nodig is, wordt er gecontroleerd of het gevraagde type overeenkomt met GasStationViewModel. Zo ja, wordt een nieuw ViewModel gemaakt met de meegegeven repository. Zo niet, wordt een fout gegenereerd.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

GasStationViewHolder

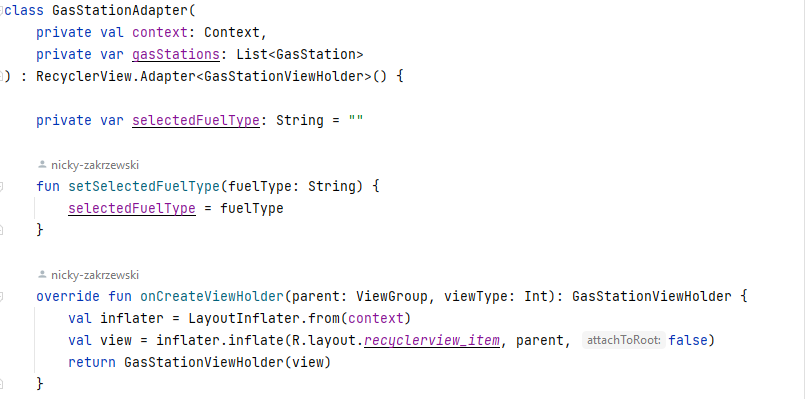
Deze ViewHolder wordt gebruikt om de weergave-elementen van een gasStation item efficiënt vast te houden en toegankelijk te maken voor hergebruik in de recyclerView.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

### GasStationAdapter

De adapter zorgt ervoor dat de gegevens aan de recyclerview kunnen worden doorgegeven. Deze erft van de GasStationViewHolder. De selectedfueltype wordt van de fragment doorgegeven aan de adapter. Voor elk item van de lijst wordt gekeken wat de naam is van het tankstation en op basis daarvan wordt de afbeelding van het juiste logo toegewezen. De plaats en de prijs worden ook doorgegeven aan de hand van deze gegevens. Er wordt nog gecontroleerd of het brandstoftype cng is, want dan is de prijs namelijk per kg en niet per liter. Met de setData functie wordt er geïnformeerd dat er iets is veranderd aan de data.



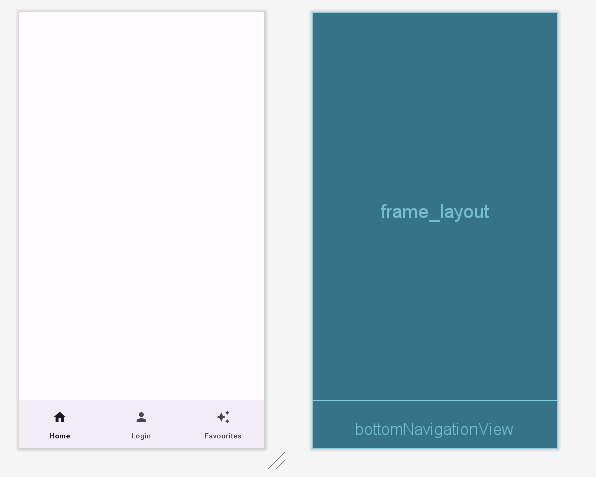


## Views

De views in deze applicatie bestaan uit fragments. Dit zijn herbruikbare stukken van de view. Deze fragments worden getoond met behulp van de MainActivity en de bottom nav bar. In het verslag zal voornamelijk uitgelegd worden over de home fragment, aangezien de login en favourites fragments nog niet volledig ontwikkeld zijn.

### MainActivity

De main activity is opgebouwd uit 2 belangrijke elementen, namelijk de bottom navigation view en de frame layout. De navigation is dus altijd te zien en in de frame layout komen de fragments tevoorschijn.





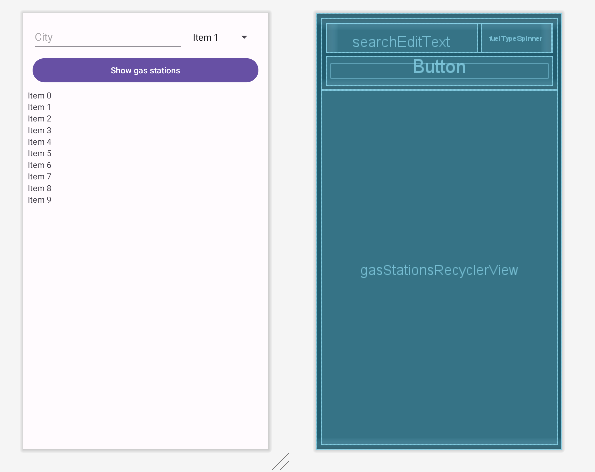
In de main activity klasse staat simpele code die ervoor zorgt dat de fragments worden gewisseld wanneer er gebruik wordt gemaakt van de bottom nav bar.

Afbeelding met tekst, schermopname, Lettertype, document

Automatisch gegenereerde beschrijving

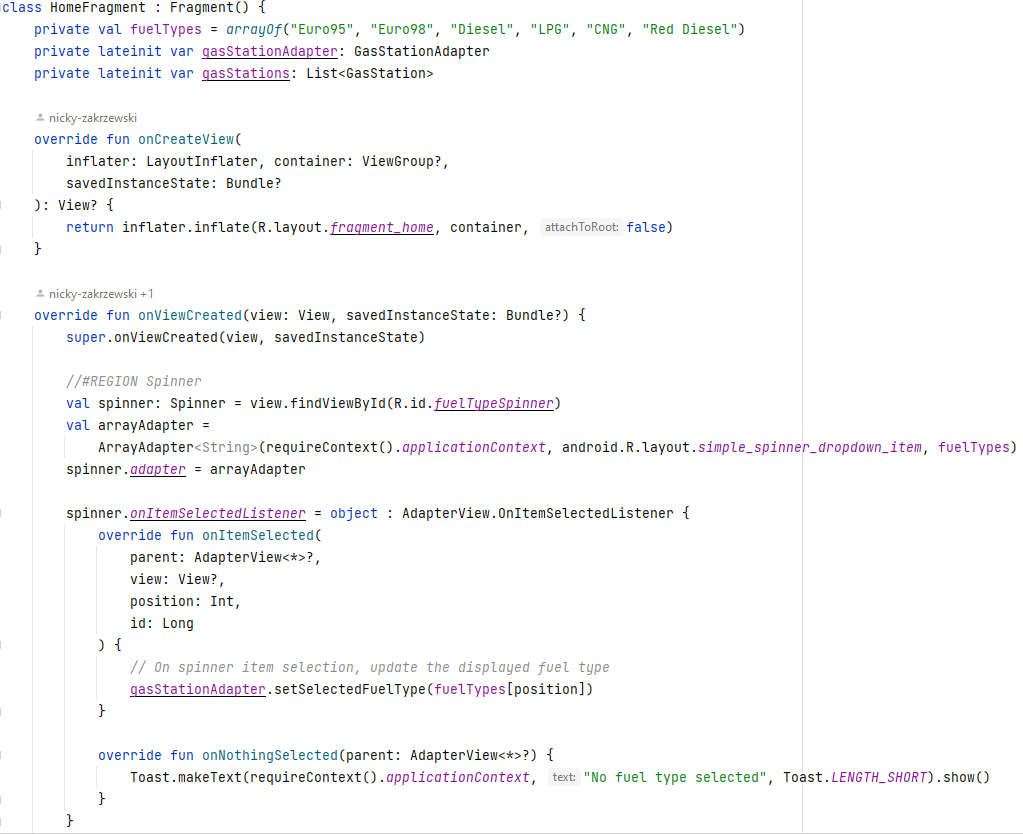
### Home fragment

De home fragment bestaat uit 4 elementen, de searchEditText om de plaat op te zoeken, de fuelTypeSpinner om de brandstof te kiezen, de searchGasStationButton om de gefilterde gegevens op te halen en de gasStationsRecyclerview om de lijst van tankstations te tonen.





In de klasse HomeFragment gebeuren er verschillende dingen. De lijst van fueltypes is erin opgeslagen en er worden ook variabelen gemaakt voor de GasStationAdapter en voor gasStations. De adapter is nodig om gegevens in de recyclerview te kunnen steken. Bij de inflater gaan we de frame layout opvullen met de home fragment. Er wordt gekeken welk item van de spinner is geselecteerd (het is in de vorm van een array). Wanneer op een brandstof wordt getikt, is dat het nieuwe geselecteerde brandstof type.



De GasStationAdapter wordt geïnitieerd met een lege lijst. Er wordt ingesteld van welke adapter de recyclerview gebruik moet maken. Eens dit allemaal ingesteld is, wordt er een listener op de show gas stations button gezet. Als de button ingedrukt wordt, worden de repository, viewmodelfactory en het ViewModel ingesteld. Vervolgens wordt aan het ViewModel gevraagd om tankstations op te halen met de ingevoerde plaatsnaam en brandstof. De setData functie informeert de adapter over de nieuwe lijst van tankstations en vuurt de notifyOnDataChanged af zodat de gegevens ververst kunnen worden.

Afbeelding met tekst, schermopname, Lettertype, document

Automatisch gegenereerde beschrijving

De items zijn een simpele cardviews met links het logo, in het midden de plaatsnaam en rechts de prijs/liter.

Afbeelding met tekst, schermopname, Lettertype, Rechthoek

Automatisch gegenereerde beschrijving