Semester Project: INFT2508 Cross-platform Application Development for Mobile Devices

Info

Presented by: **Demetrio Battaglia**
Student ID: **584801**
Date: **04.12.2022**
Please note that this document is best visualised in its HTML (exported) *format, although it is also provided in Markdown* (original) *and PDF* (worst).

Product Description & Features

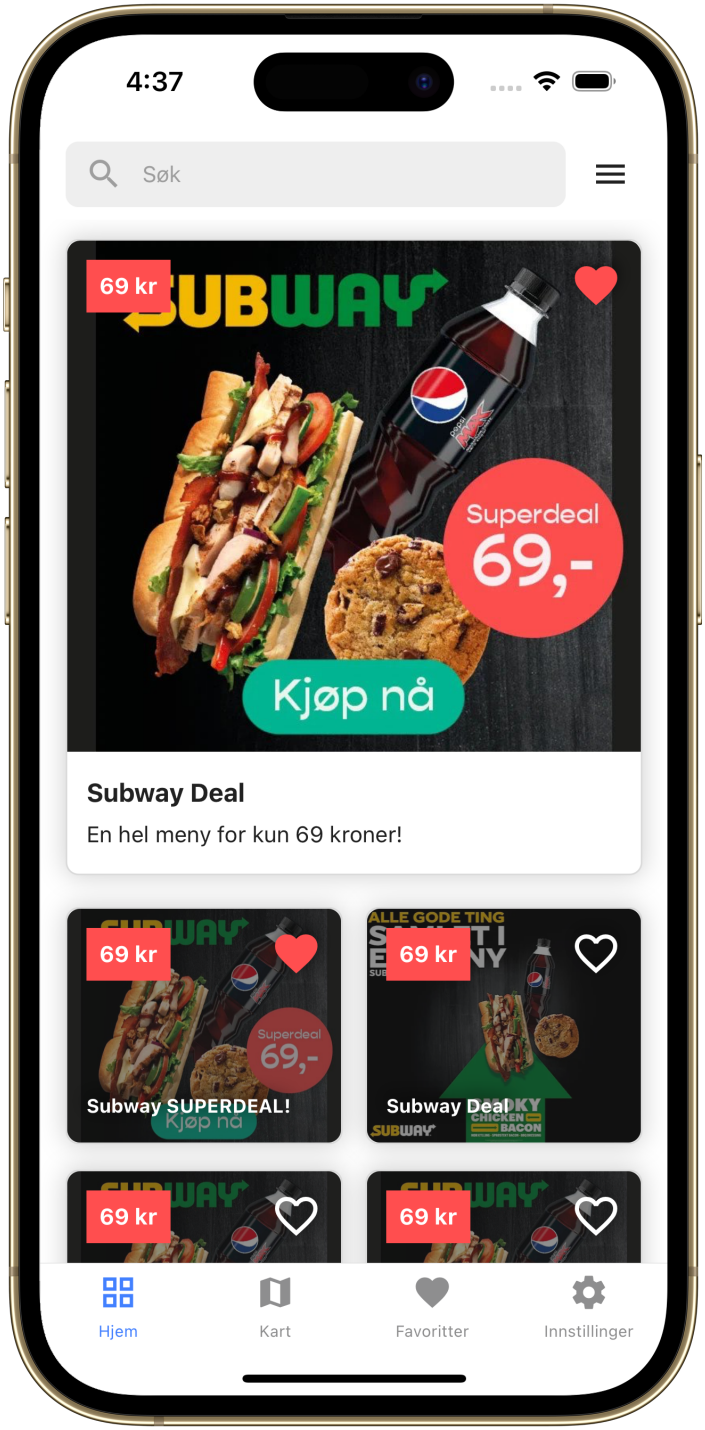
This app follows the project presented by the professor of an e-commerce platform where you are presented with products in a front page, which you can expand to see in detail.

An example is provided with the **Let's Deal** app, which is used here as inspiration.

Features include:

- a front page with products presented in a grid with elements of different sizes (filter-able)
- a details page for each product
- a map with all the geolocated products (filter-able)
- the ability to add a product to personal favourites
- the ability to change app language and appearance

Landing page


4:37
Søk
69 kr
SUBWAY
Superdeal
69,-
Kjøp nå
Subway Deal
En hel meny for kun 69 kroner!
69 kr
Superdeal
69,-
Subway SUPERDEAL!
Kjøp nå
69 kr
ALLE GODE TING
Subway Deal
69 kr
69 kr
Hjem
Kart
Favoritter
Innstillinger

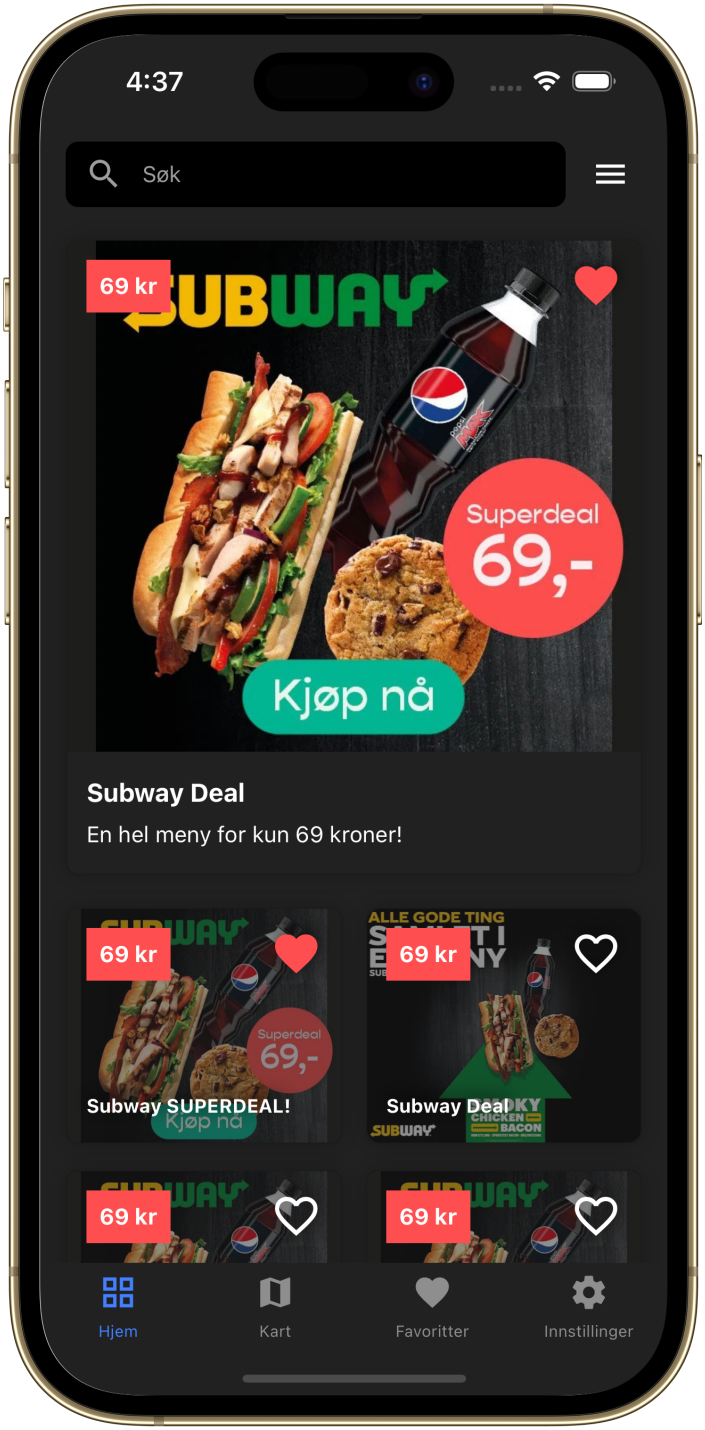


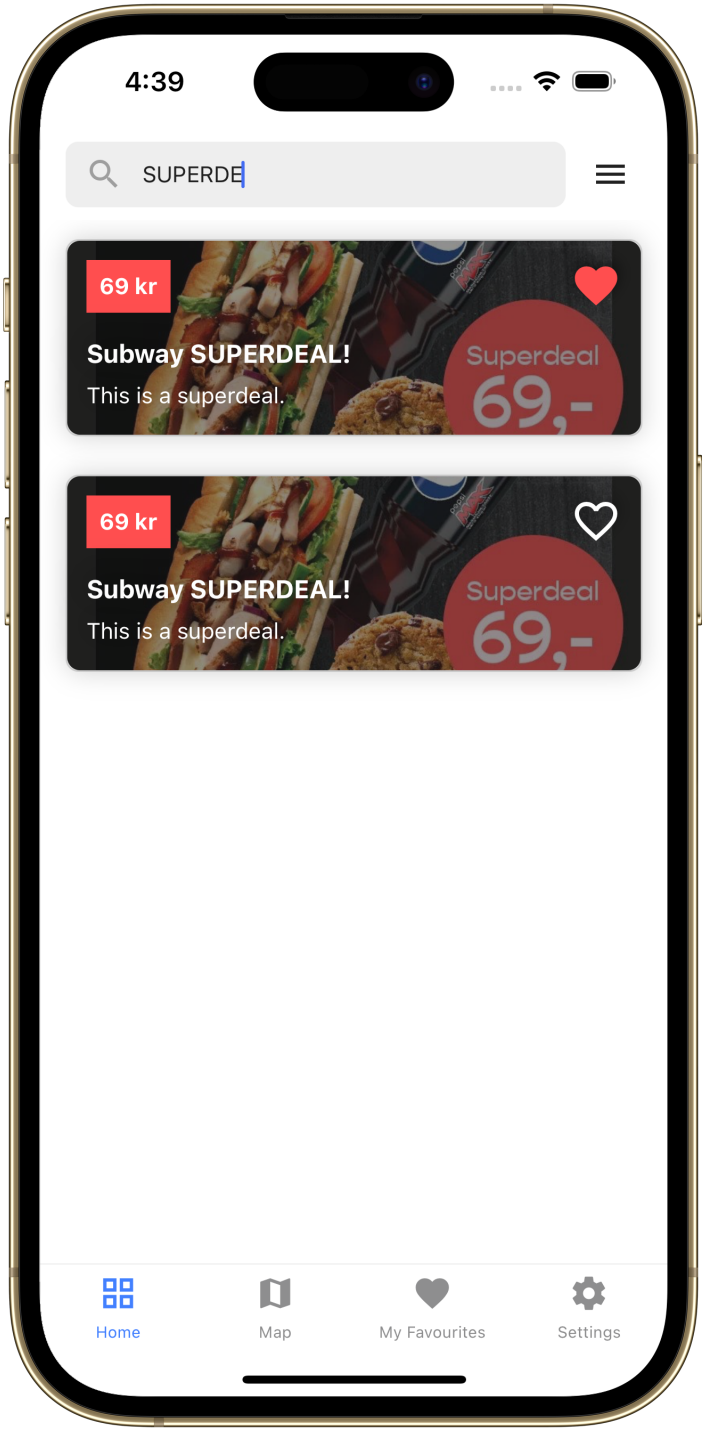
Figure 1. Landing page in both light and dark mode

As instructed, cards in landing page describing products can be presented in multiple forms:

- **Big**: This is the biggest card, the first one you can see in the screenshot
- **Twins**: This is the smallest card, it fits along with another one on the same row and presents the minimum information it can.
- **Small**: This is a medium card, which you cannot see in the first screenshot, because the cards are presented cyclically (more on this later)

The cycling of cards works like this: **one** big card, **four** twin cards and **two** small cards. Then a counter is resetted and the cycle starts again.

However, when filtering by search terms with the search bar above, only **small** cards are presented, whereas when filtering by category, the default cycle is used:



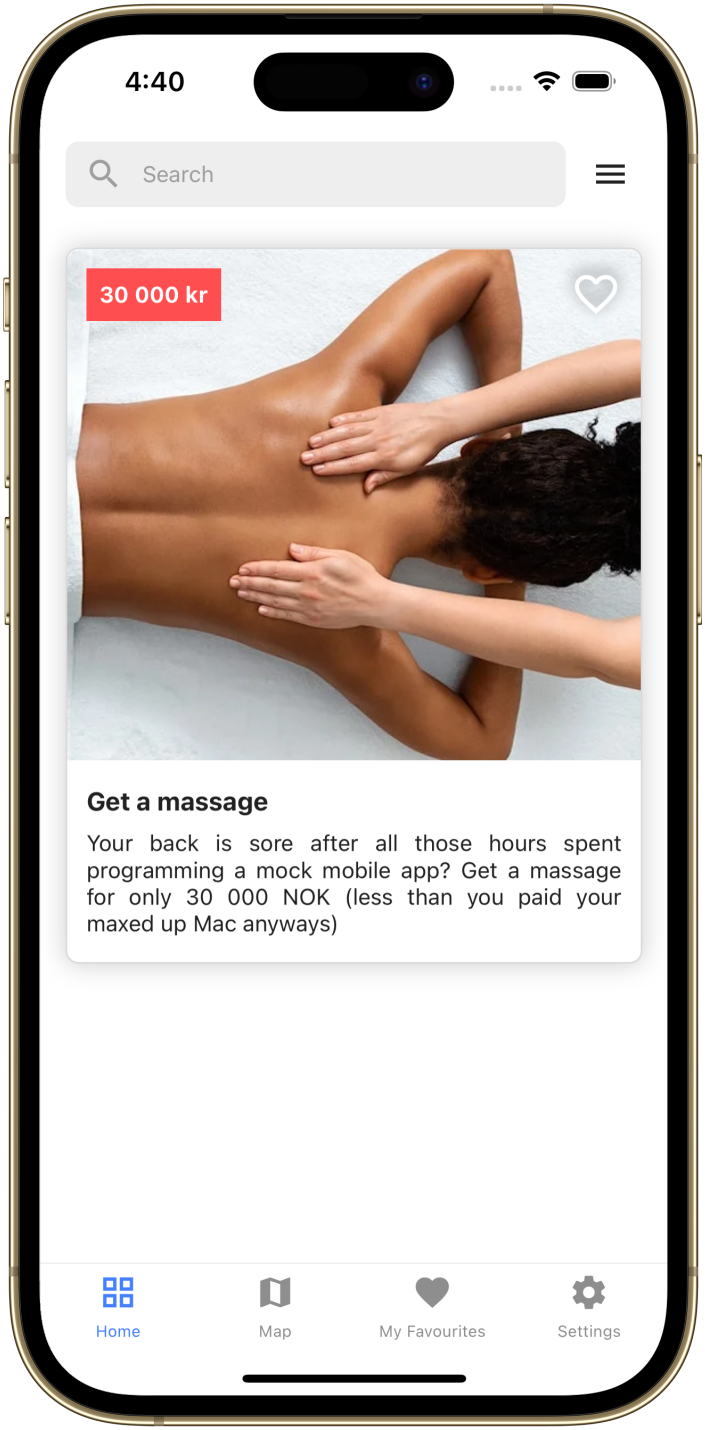
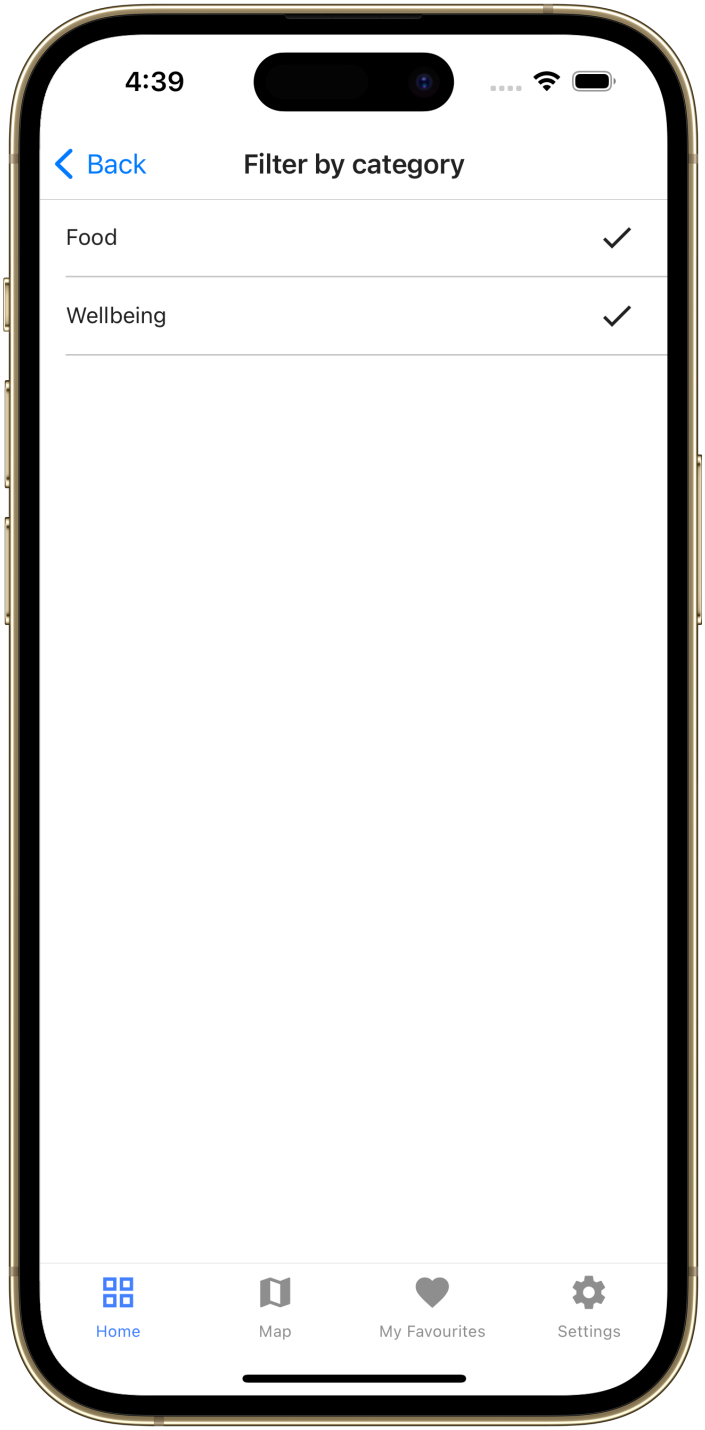


Figure 2. (a) Search by words. (b) Filter by category

Filtering by category

When pressing the hamburger button, a new view is presented with a list of categories to choose from. Multiple choice is allowed, in **figure 2a** all categories are selected, in **figure 2b** only **Wellbeing** is selected. The corresponding selections are shown in **figure 3**.


4:39
Back
Filter by category
Food
Wellbeing
Home
Map
My Favourites
Settings

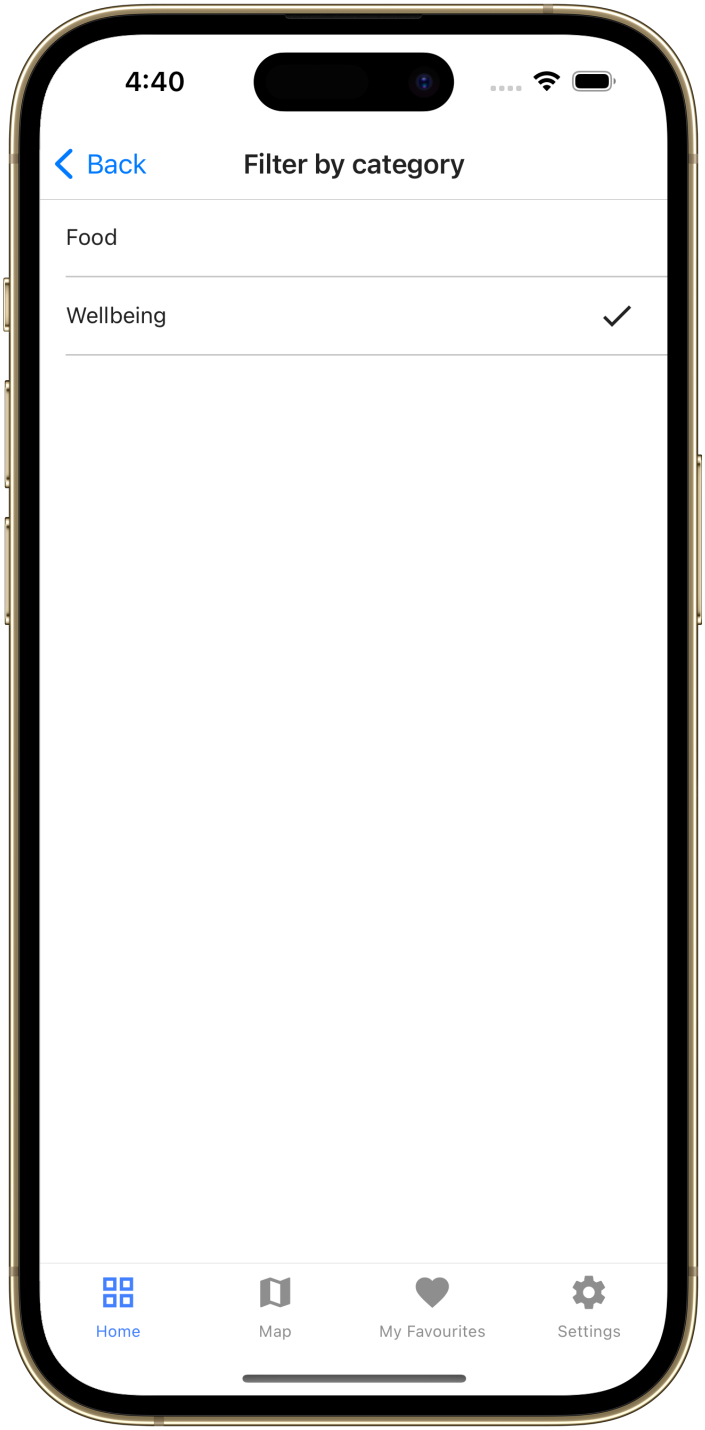


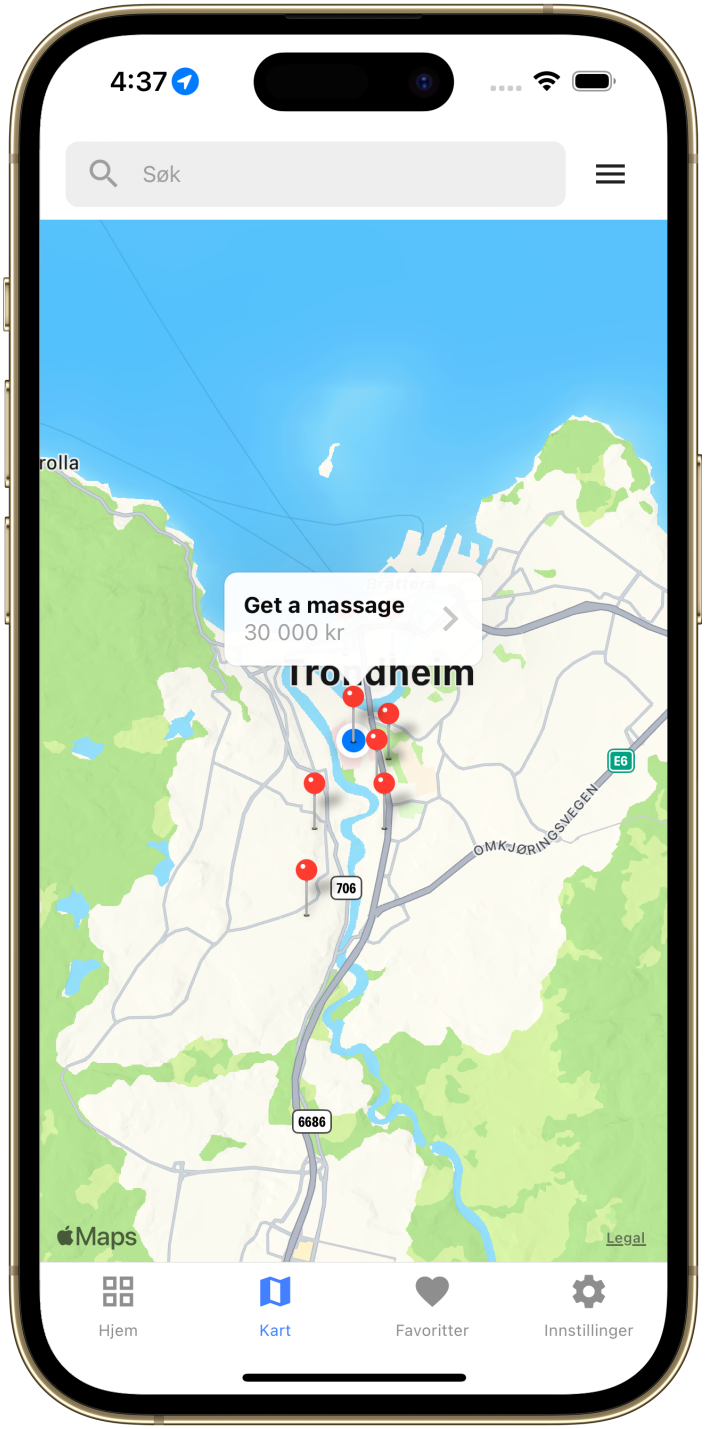
Figure 3. (a) All categories visible. (b) Filter by *Wellbeing*.

Selecting no categories is also possible, and it's the same as selecting all of them, or rather "not filtering by category".

Map view

This was an optional feature, but a rather interesting one, especially for the use of markers in the map. Furthermore, this feature allows one to understand what's behind **permissions** (geolocation) and the difference between using **MapKit** and **Google Maps** (an API key is needed for GMaps).

An example of a selected product in the map is provided below, in both light and dark mode.



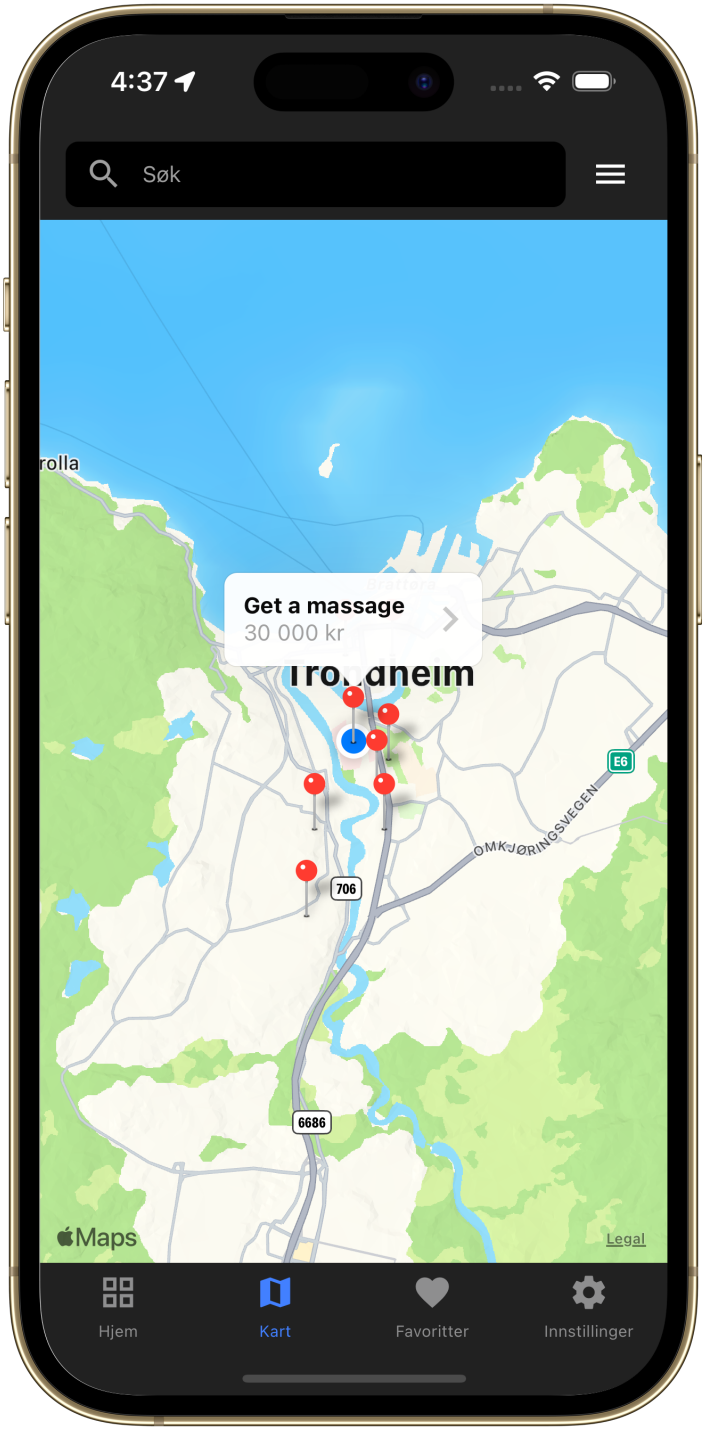


Figure 4. Map view of all the products in the database

Pressing on the callout of a marker will lead to the product page, described in the next section.

Filtering by words or category works on here too.

Product details

One of the major features requested for this app is to show products in detail in a page dedicated to each one of them. Below are two screenshots of the details of a product, for light and dark mode.

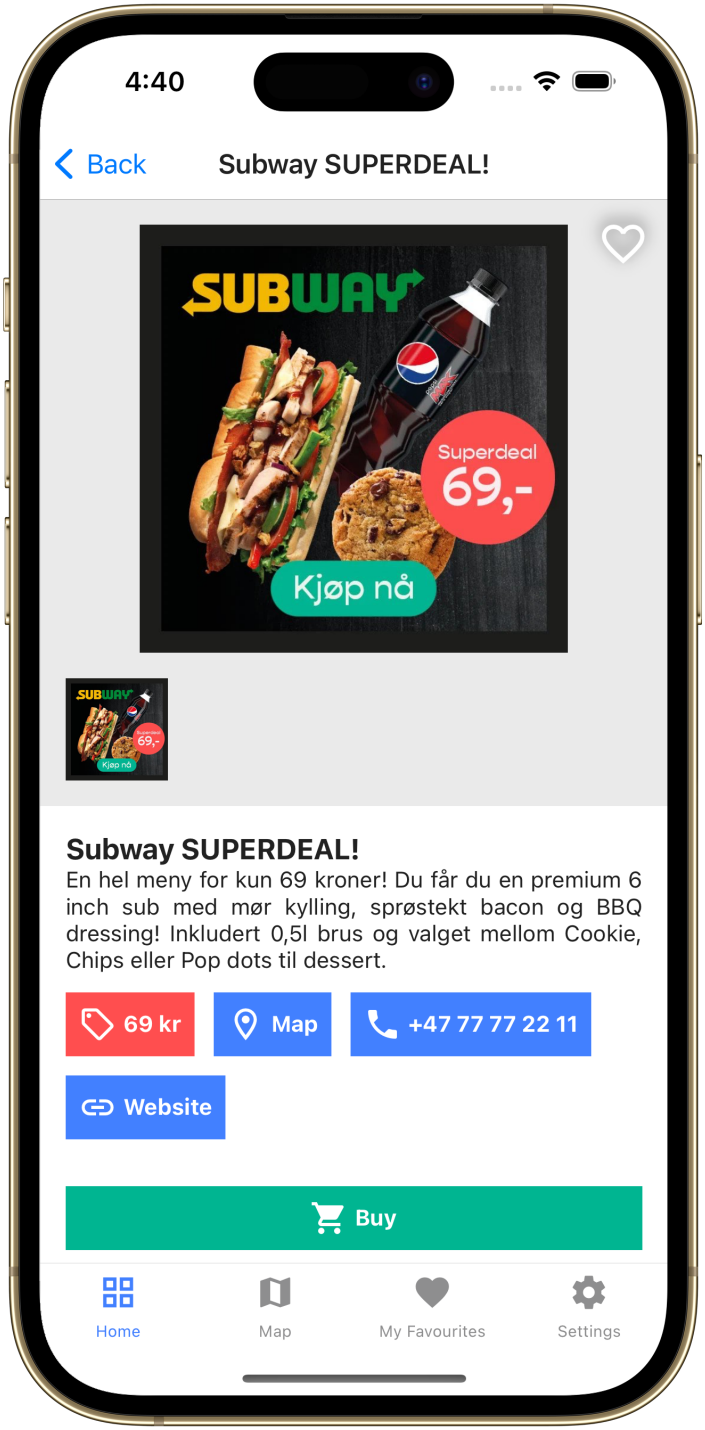

4:40
Back
Subway SUPERDEAL!
SUBWAY
Superdeal
69,-
Kjøp nå
Subway SUPERDEAL!
En hel meny for kun 69 kroner! Du får du en premium 6 inch sub med mør kylling, sprøstekt bacon og BBQ dressing! Inkludert 0,5l brus og valget mellom Cookie, Chips eller Pop dots til dessert.
69 kr
Map
+47 77 77 22 11
Website
Buy
Home
Map
My Favourites
Settings

Figure 5. Product details with long description, price, contact information, location (pressing on **Map** will focus the product on the map) and the option to buy.

Pressing on the telephone or website buttons will respectively initiate a call or open the system browser to the linked website.

Pressing the **Buy** button will do something different based on the app settings and system capabilities: on iOS, Touch ID and Face ID offer a reliable way of protecting data, but are mutually exclusive. If the device supports it, one can choose to "protect" purchases with either of those biometrics in the settings, like shown in **figure 6**.


4:38
Back
Subway Deal
SUBWAY
Superdeal
69,-
Face ID
Subway Deal
En hel meny for kun 69 kroner! Du får du en premium 6 inch sub med mør kylling, sprøstekt bacon og BBQ dressing! Inkludert 0,5l brus og valget mellom Cookie, Chips eller Pop dots til dessert.
69 kr
Mappa
Compra
Home
Mappa
Preferiti
Impostazioni


4:39
Back
Subway SUPERDEAL!
SUBWAY
Superdeal
69,-
Kjøp nå
Purchase failed
Your authorisation method did not confirm your identity.
Confirm
Subway SUPERDEAL!
En hel meny for kun 69 kroner! Du får du en premium 6 inch sub med mør kylling, sprøstekt bacon og BBQ dressing! Inkludert 0,5l brus og valget mellom Cookie, Chips eller Pop dots til dessert.
69 kr
Map
Buy
Home
Map
My Favourites
Settings

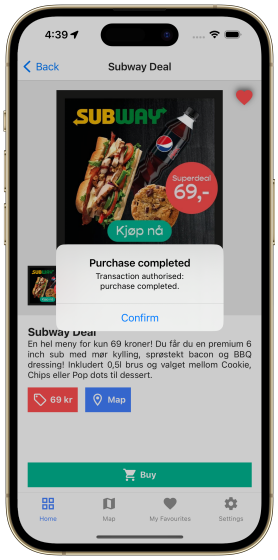
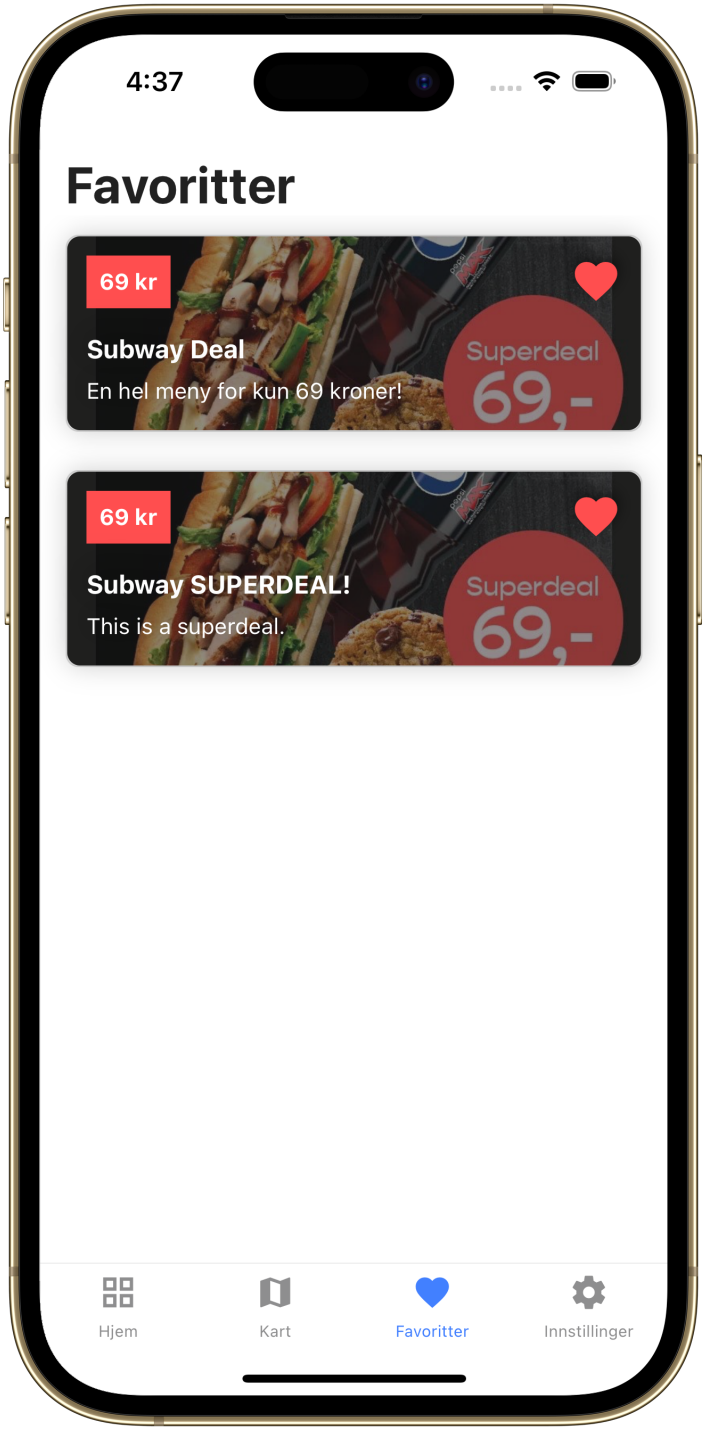

4:39
Back
Subway Deal
SUBWAY
Superdeal
69,-
Kjøp nå
Purchase completed
Transaction authorised: purchase completed.
Confirm
Subway Deal
En hel meny for kun 69 kroner! Du får du en premium 6 inch sub med mør kylling, sprøstekt bacon og BBQ dressing! Inkludert 0,5l brus og valget mellom Cookie, Chips eller Pop dots til dessert.
69 kr
Map
Buy
Home
Map
My Favourites
Settings

Figure 6. From left to right, Face ID request, Face ID fail, Face ID success, Settings selection (None / Touch ID / Face ID), Settings selection failed (unsupported technology selected).

My Favourites

One of the core features is also to be able to mark products as **favourites**. Doing so will add them to a view called "My Favourites". In each product card you can see an heart overlaying the product image, pressing it will add it to favourites.


4:37
Favoritter
69 kr
Subway Deal
En hel meny for kun 69 kroner!
Superdeal
69,-
69 kr
Subway SUPERDEAL!
This is a superdeal.
Superdeal
69,-
Hjem
Kart
Favoritter
Innstillinger

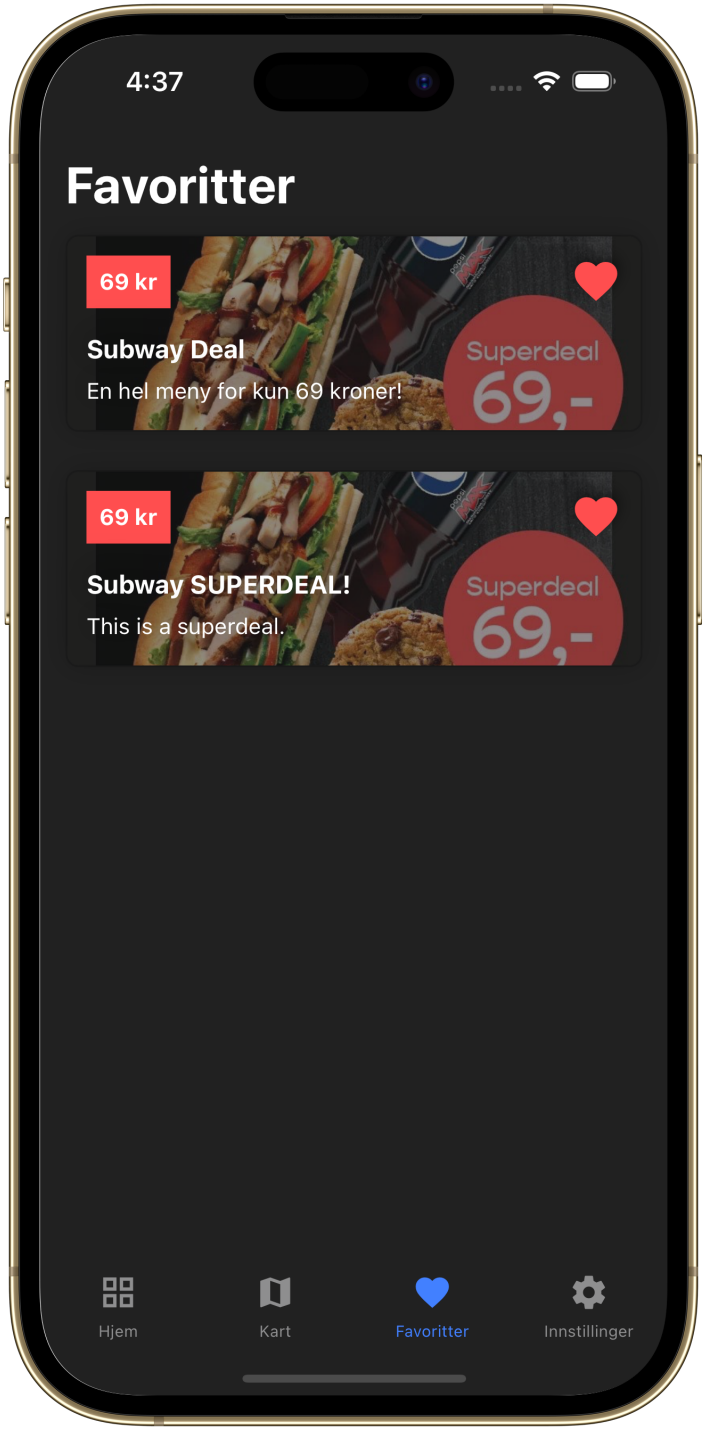
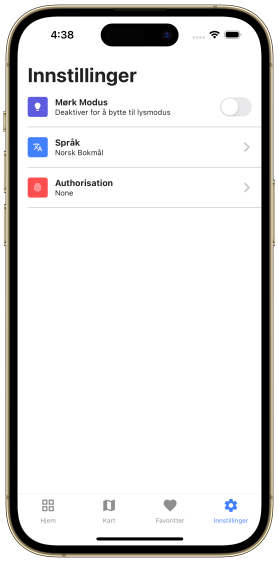


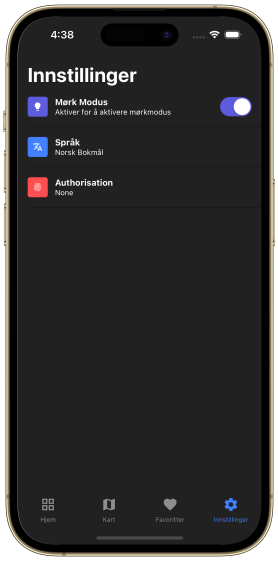
Figure 7. My Favourites page (localised in Norwegian Bokmål)

Pressing each card (here presented always as **small** type) will open the product details page just described.

Settings

Finally, the settings view will allow us to set **Light / Dark Mode**, **Language**, and **Authentication** method for purchases (described in **Product details** section).


4:38
Innstillinger
Mørk Modus
Deaktiver for å bytte til lysmodus
Språk
Norsk Bokmål
Authorisation
None
Hjem
Kart
Favoritter
Innstillinger


4:38
Innstillinger
Mørk Modus
Aktiver for å aktivere mørkmodus
Språk
Norsk Bokmål
Authorisation
None
Hjem
Kart
Favoritter
Innstillinger

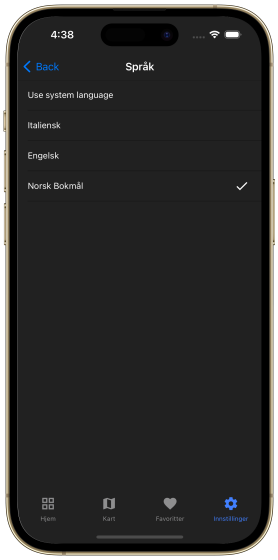

4:38
Back
Språk
Use system language
Italiensk
Engelsk
Norsk Bokmål
Hjem
Kart
Favoritter
Innstillinger

Figure 8. From left to right: Norwegian Light Mode Settings; Norwegian Dark Mode Settings; Norwegian Language Selection (Dark); Italian Language Selection (Light); Italian Dark Mode Settings.

Design decisions and main components

```
.
├── components/
│   ├── FilterBar.js
│   ├── ProductCard.js
│   ├── Tag.js
│   └── Title.js
├── data/
│   ├── public/img/
│   └── db.json
├── localisation/
│   ├── en-GB.json
│   ├── it-IT.json
│   ├── localisation.js
│   └── nb-NO.json
├── views/
│   ├── Home.js
│   ├── MapView.js
│   ├── MyFavourites.js
│   ├── ProductDetails.js
│   ├── Settings.js
│   └── SettingsDetailed.js
├── App.js
├── AppState.js
├── Globals.js
├── index.js
└── palette.js
```

The code can be divided in **General Components**, **Views** and **Views Sub-Components**.

The **General Components** are located in the `components/` : those are the components that are mostly re-used in multiple different views or components (e.g. `Title` is used in `Settings` and `MyFavourites` , `Tag` is used in `ProductCard` and `ProductDetails`).

The **Views** are components that are used as different pages in the app:

- **Home**: Here we have a stack navigator with three sub-views
  - **Landing page**
  - **ProductDetails**: shows when a card is clicked
  - **CategorySeection**: shows when clicking on the hamburger to filter by category
- **MapView**: Stack navigator with two sub-views
  - **MapPage**: the map itself
  - **ProductDetails**: shows when a callout is clicked
- **MyFavourites**: Stack navigator with two sub-views
  - **Favourites list**
  - **ProductDetails**: shows when a product is clicked

- **Settings**: a stack view with three sub-views
  - **Settings list**
  - **Language selection**
  - **Authentication selection**

The **Views Sub-Components** are components that are made re-usable and sometimes exported when they are mostly or exclusively used within the view itself (e.g. `ImageGallery` in `ProductDetails`).

These sub-components have been made not to pollute too much the main reusable components, that mostly constitute important parts of the app.

Additionally, I have made use of `createContext` to create a global state within the app that stores the current settings and allows us to update the app appearance and language without closing and reloading the app.

As for what concerns UI Design, I've taken mostly inspiration from the Let's Deal app layout, and adapted it to the project requisites.

I have used cards instead of a simple grid of products because it helps picturing what the assignment asks for: different sizes.

The colour palette is mostly taken from Let's Deal.

Launching the app

Data server

The app requires a server to be running, so you'll have to start it with:

```
npm run mockapi
```

This will start a json-server on all interfaces (**0.0.0.0**) instead of **localhost**, because the Android Emulator will contact **10.0.2.2** to access the host machine. Whereas iOS Simulator will contact **localhost** because it shares the same virtual network as the host machine. This behaviour is accounted for in `Globals.js`. Default port is **3000**.

The server is already set up with some data about products and favourites. It also contains some images for the products taken from the Let's Deal website.

These images are property of the respective companies that ran the ads on the website.

Metro bundler

```
npm start
```

iOS

```
npm run ios
```

Android

```
npm run android
```

Or, if you have multiple JDK in your system, you can use

```
npm run android-jdk11
```

As the AVD requires Java 11.

Please note that Android will require a Google Maps API key in the app configuration before running. You can edit the API key in `android/app/src/main/AndroidManifest.xml` at the line that looks like this:

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="[ API KEY HERE ]" />
```

Please also note that iOS does not require an API key for MapKit.