Technical Project Report - Android Module

EatAnywhere

Subject:	Computação Móvel
Date:	Aveiro, 27 January 2021
Authors:	84948: André Neves 85097: Dany Costa
Project abstract:	**EatAnywhere** is a detailed restaurant application that provides information about restaurants provided by the most exhaustive curated restaurant information API, Zomato, such as rating, cuisines, price, reviews and a few more. Besides those, the user can take advantage of a map that marks restaurants nearby and is also able to select his favorites restaurants and obtain directions to any location.

Table of contents:

1 Application concept

Nowadays when it comes to public services there are many applications that help people to be more informed and facilitate their daily activities. EatAnywhere is not different , as the name says, the aim is to help people eat anywhere they want using our services. But it's not just that, EatAnywhere allows their users to see detailed information about each restaurant what allows them to search the kind of restaurant they want (if they want fast food , sushi , traditional food , etc) and also know the restaurant quality not only by the ratings but also with the comments from other users. Besides that, our users have a list section where they can store all their favorite restaurants and , beyond that, they can also take full advantage of a map that allows them to be informed about restaurants near them and the respective directions.

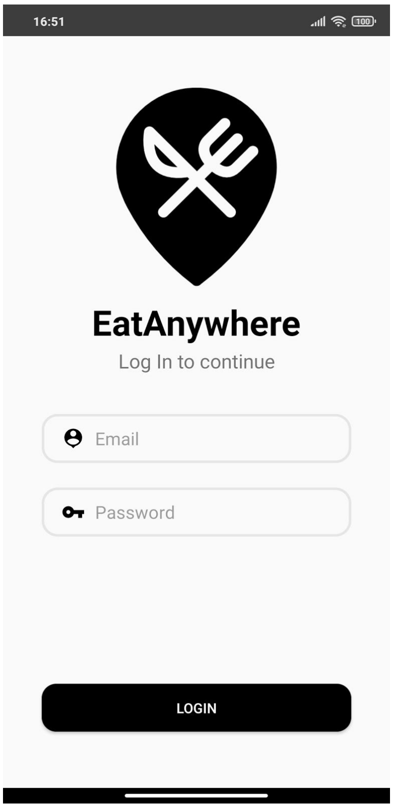
Basically EatAnywhere is an app designed for restaurant lovers who really like to be informed about all the restaurants near them giving all the information they need such as the restaurant cellphone, precise location , restaurant type , directions and users reviews. (rever melhor esta parte)

2 Implemented solution

Architecture Overview

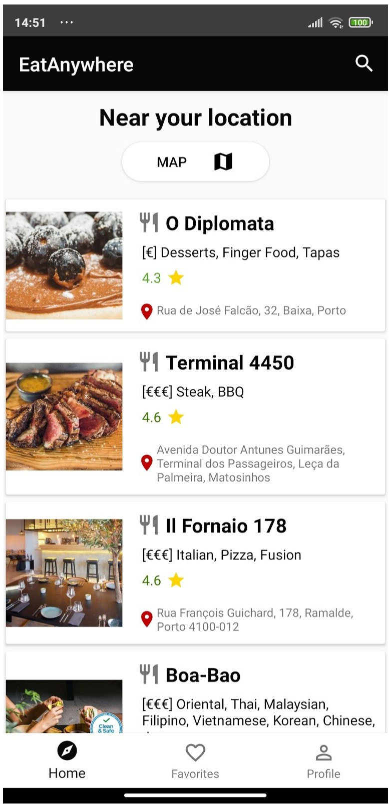
Our app is divided into 3 main fragments: Home fragment, Favorites fragment and Profile fragment. From each of these we launch new activities to show more information.

In the **home fragment**, we display a list of nearby restaurants using a LinearLayout RecyclerView, in order to be dynamic. These results are fetched from an API that will be explained later in this document, and are displayed only when the data is available. By clicking on an item, a new activity is launched with details about the restaurant selected, such as name, aggregated rating, total reviews, cuisines, price, address, phone number, directions and real reviews. In this page, we implemented 3 intents (2 from google maps and 1 dialer): show the address in google maps, trace a route to the destination from the current location (directions) and the phone dialer to call the restaurant's number. Also in this page, we are able to mark a restaurant as a favorite, which will be stored in the Firebase Firestore database.



EatAnywhere

Log In to continue



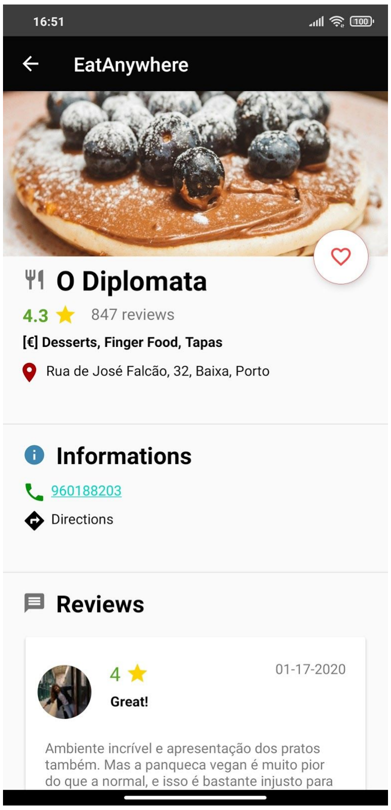
Near your location

O Diplomata
[€] Desserts, Finger Food, Tapas
4.3 ★
Rua de José Falcão, 32, Baixa, Porto

Terminal 4450
[€€€] Steak, BBQ
4.6 ★
Avenida Doutor Antunes Guimarães, Terminal dos Passageiros, Leça da Palmeira, Matosinhos

Il Fornaio 178
[€€€] Italian, Pizza, Fusion
4.6 ★
Rua François Guichard, 178, Ramalde, Porto 4100-012

Boa-Bao
[€€€] Oriental, Thai, Malaysian, Filipino, Vietnamese, Korean, Chinese,

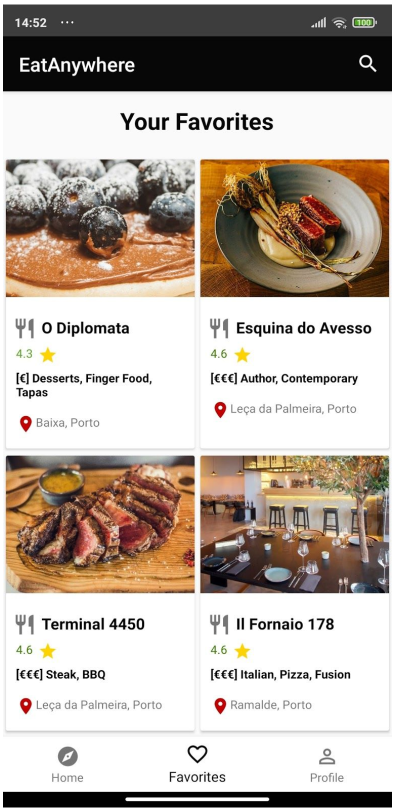


O Diplomata
4.3 ★ 847 reviews
[€] Desserts, Finger Food, Tapas
Rua de José Falcão, 32, Baixa, Porto

Informations
960188203
Directions

Reviews
4 ★ 01-17-2020
Great!
Ambiente incrível e apresentação dos pratos também. Mas a panqueca vegan é muito pior do que a normal, e isso é bastante injusto para

In the **favorites fragment**, we display the user's favorite restaurants on a GridLayout recycler view. Once an item is clicked, the same activity as before is launched to show its details. The floating action button is responsible for triggering the Firebase Firestore database operations, in this case adding/removing the restaurant to/from the user's favorites.



14:52 ···

EatAnywhere

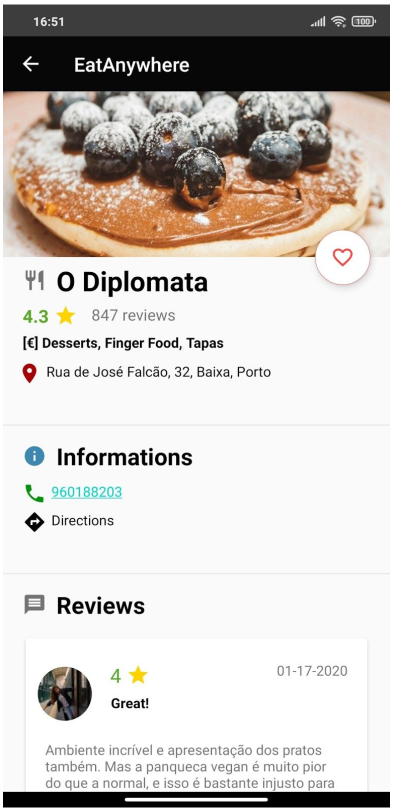
Your Favorites

O Diplomata
4.3 ★
[€] Desserts, Finger Food, Tapas
Baixa, Porto

Esquina do Avesso
4.6 ★
[€€€] Author, Contemporary
Leça da Palmeira, Porto

Terminal 4450
4.6 ★
[€€€] Steak, BBQ
Leça da Palmeira, Porto

Il Fornaio 178
4.6 ★
[€€€] Italian, Pizza, Fusion
Ramalde, Porto



16:51

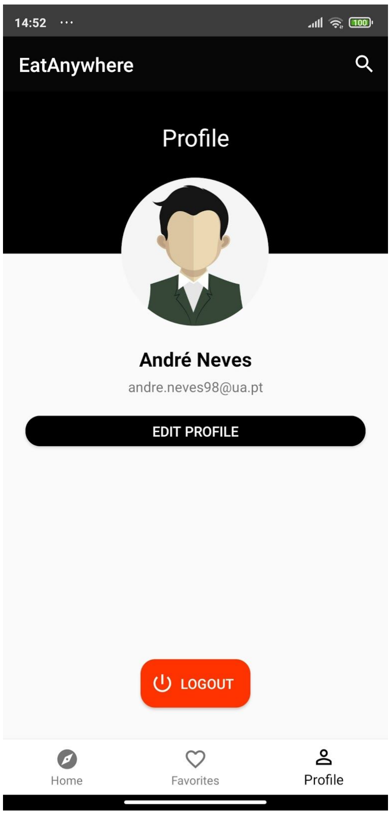
EatAnywhere

O Diplomata
4.3 ★ 847 reviews
[€] Desserts, Finger Food, Tapas
Rua de José Falcão, 32, Baixa, Porto

Informations
960188203
Directions

Reviews
4 ★ 01-17-2020
Great!
Ambiente incrível e apresentação dos pratos também. Mas a panqueca vegan é muito pior do que a normal, e isso é bastante injusto para

Finally, in the **profile fragment** is where the user can see and edit his information, as well as logout and leave the app. This information, like the restaurants', is stored in our database and can be changed.

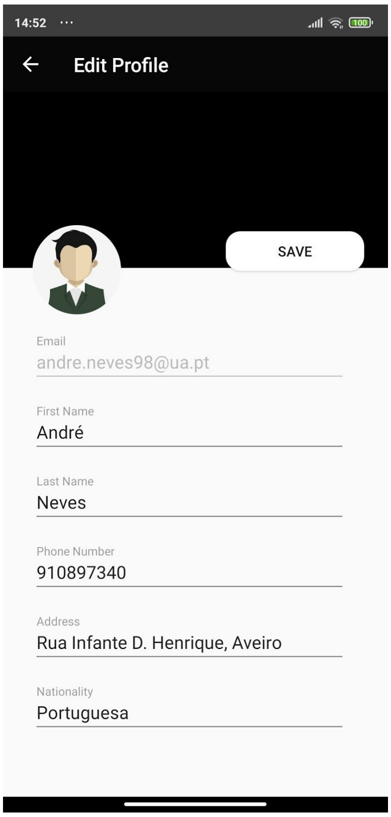


14:52 ···

EatAnywhere

Profile

André Neves
andre.neves98@ua.pt



14:52 ···

Edit Profile

Email
andre.neves98@ua.pt

First Name
André

Last Name
Neves

Phone Number
910897340

Address
Rua Infante D. Henrique, Aveiro

Nationality
Portuguesa

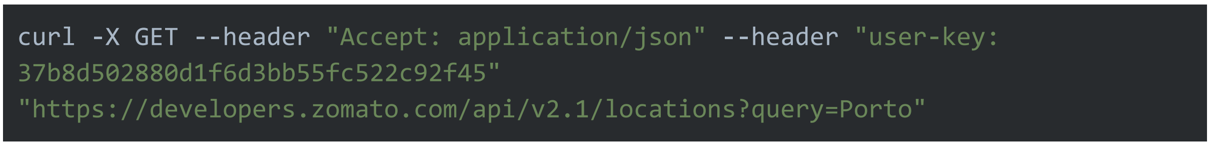
API and Sensors

Zomato API

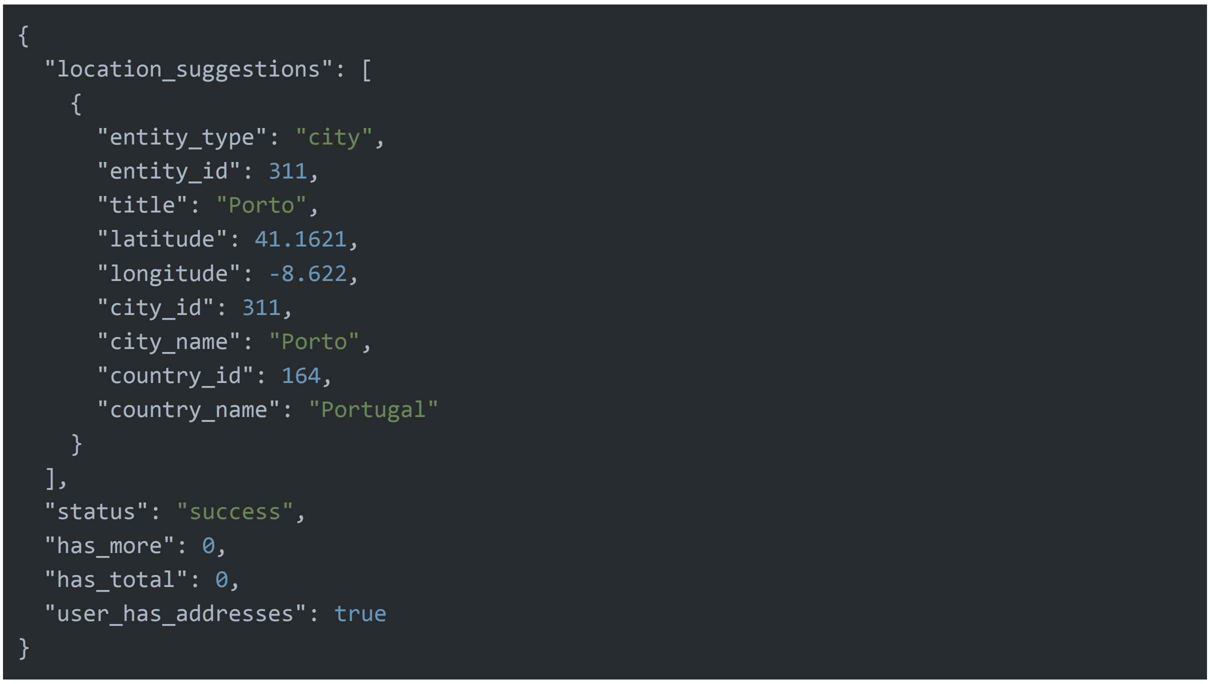
In order to fetch the restaurants' informations, we used the Zomato API which provides the most exhaustive curated restaurant information and it's free (1000 calls/day). The communication was done using Retrofit, as recommended at practical classes. We faced some challenges retrieving the data caused by the structure of the data we were being given, which we have solved by creating several classes in order to correctly map the nested JSONs.

We use 3 queries for our data: one that searches a given location and returns 2 required fields for the next query (entity_id and entity_type), that represent a city ID and type (e.g. Porto: entity_id=311, entity_type=city); these fields are used in the second query for more precise results, which searches restaurants in that location; the third and last query is responsible for fetching reviews of a given restaurant. Overall, our requests follow this structure:

1. *Location query*



```
curl -X GET --header "Accept: application/json" --header "user-key:
37b8d502880d1f6d3bb55fc522c92f45"
"https://developers.zomato.com/api/v2.1/locations?query=Porto"
```



```
{
  "location_suggestions": [
    {
      "entity_type": "city",
      "entity_id": 311,
      "title": "Porto",
      "latitude": 41.1621,
      "longitude": -8.622,
      "city_id": 311,
      "city_name": "Porto",
      "country_id": 164,
      "country_name": "Portugal"
    }
  ],
  "status": "success",
  "has_more": 0,
  "has_total": 0,
  "user_has_addresses": true
}
```

2. *Search query*

```
curl -X GET --header "Accept: application/json" --header "user-key:
37b8d502880d1f6d3bb55fc522c92f45"
"https://developers.zomato.com/api/v2.1/search?entity_id=311&entity_type=city"
```

```
{
      "restaurant": {
        "R": {
          "res_id": 18165844,
          "is_grocery_store": false,
          "has_menu_status": {
            "delivery": -1,
            "takeaway": -1
          }
        },
        "apikey": "37b8d502880d1f6d3bb55fc522c92f45",
        "id": "18165844",
        "name": "Terminal 4450",
        "url":
"https://www.zomato.com/porto/terminal-4450-leça-da-palmeira?utm_source=api_basic_user&u
tm_medium=api&utm_campaign=v2.1",
        "location": {
          "address": "Avenida Doutor Antunes Guimarães, Terminal dos Passageiros, Leça
da Palmeira, Matosinhos",
          "locality": "Leça da Palmeira",
          "city": "Porto",
          "city_id": 311,
          "latitude": "41.1875441294",
          "longitude": "-8.6989925802",
          "zipcode": "",
          "country_id": 164,
          "locality_verbose": "Leça da Palmeira, Porto"
        },
        "switch_to_order_menu": 0,
        "cuisines": "Steak, BBQ",
        "timings": "12:30 a 15:00, 19:30 a 23:00 (Seg, Ter, Qua, Qui, Dom), 12:30 a
15:00, 19:30 a 24:00 (Sex-Sáb)",
        "average_cost_for_two": 70,
        "price_range": 4,
        "currency": "€",
        "highlights": [
          "Lunch",
          "Serves Alcohol",
          "Debit Card",
          "Dinner",
          "Cash",
          "Credit Card",
          "Sports TV",
          "Fullbar",
          "Wine",
          "Serves Super Bock Selecção 1927",
          "Wifi",
          "Gin Bar",
          "Wine By Glass",
          "Indoor Seating",
          "Table booking recommended"
        ],
        "offers": [],
        "opentable_support": 0,
        "is_zomato_book_res": 1,
        "mezzo_provider": "ZOMATO_BOOK",
        "is_book_form_web_view": 0,
        "book_form_web_view_url": "",
        "thumb":
"https://b.zmtcdn.com/data/res_imagery/18165844_RESTAURANT_3a7207e33875eccb81b8d94a9b602
5eb.jpg?fit=around%7C200%3A200&crop=200%3A200%3B%2A%2C%2A",
        "user_rating": {
          "aggregate_rating": "4.6",
          "rating_text": "Excellent",
          "rating_color": "3F7E00",
          "rating_obj": {
            "title": {
              "text": "4.6"
            },
            "bg_color": {
              "type": "lime",
              "tint": "800"
            }
          },
          "votes": 2219
        },
        "all_reviews_count": 924,
        "photos_url":
"https://www.zomato.com/porto/terminal-4450-leça-da-palmeira/photos?utm_source=api_basic
_user&utm_medium=api&utm_campaign=v2.1#tabtop",
        "photo_count": 2469,
        "menu_url":
"https://www.zomato.com/porto/terminal-4450-leça-da-palmeira/menu?utm_source=api_basic_u
ser&utm_medium=api&utm_campaign=v2.1&openSwipeBox=menu&showMinimal=1#tabtop",
        "featured_image":
"https://b.zmtcdn.com/data/res_imagery/18165844_RESTAURANT_3a7207e33875eccb81b8d94a9b602
5eb.jpg",
        "medio_provider": "",
        "has_online_delivery": 0,
        "is_delivering_now": 0,
        "store_type": "",
        "include_bogo_offers": true,
        "deeplink": "zomato://restaurant/18165844",
        "is_table_reservation_supported": 1,
        "has_table_booking": 1,
        "book_url":
"https://www.zomato.com/porto/terminal-4450-leça-da-palmeira/book?utm_source=api_basic_u
ser&utm_medium=api&utm_campaign=v2.1",
        "events_url":
"https://www.zomato.com/porto/terminal-4450-leça-da-palmeira/events#tabtop?utm_source=ap
i_basic_user&utm_medium=api&utm_campaign=v2.1",
        "phone_numbers": "919851933",
        "all_reviews": {
          "reviews": [
            {
              "review": []
            },
            {
              "review": []
            },
            {
              "review": []
            },
            {
              "review": []
            },
            {
              "review": []
            }
          ]
        },
        "establishment": [
            "Casual Dining"
        ],
        "establishment_types": []
      }
    },
```

3. *Reviews query*

```
curl -X GET --header "Accept: application/json" --header "user-key: 37b8d502880d1f6d3bb55fc522c92f45" "https://developers.zomato.com/api/v2.1/reviews?res_id=18420894"
```

```
{
  "reviews_count": 707,
  "reviews_start": 0,
  "reviews_shown": 1,
  "user_reviews": [
    {
      "review": {
        "rating": 5,
        "review_text": "Excelente atendimento.. espaço acolhedor.. a comida cinco estrelas... a repetir muitas vezes... um italiano quer merece a distinção no serviço prestado",
        "id": 51514912,
        "rating_color": "305D02",
        "review_time_friendly": "6 days ago",
        "rating_text": "Insane!",
        "timestamp": 1610153097,
        "likes": 0,
        "user": {
          "name": "Diana Thais",
          "zomato_handle": "",
          "foodie_level": "Big Foodie",
          "foodie_level_num": 7,
          "foodie_color": "ffae4f",
          "profile_url":
"https://www.zomato.com/users/diana-thais-50952545?utm_source=api_basic_user&utm_medium=
api&utm_campaign=v2.1",
          "profile_image":
"https://b.zmtcdn.com/data/user_profile_pictures/7e9/1650390d58b798082bb7e0467d0a57e9.jp
g?fit=around%7C100%3A100&crop=100%3A100%3B%2A%2C%2A",
          "profile_deeplink": "zomato://u/50952545"
        },
        "comments_count": 0
      }
    }
  ]
}
```

In terms of sensors, we used real location to use with the map's intent.

Google Maps API & Places API

In order to take advantage of a personalized map implemented in our system we used Google Maps API that detects user current location. To achieve our main goal we had to use Places API to make a http request with user current location and receive a JSON file with all restaurants information near him as we can see in the code below.

Resquest:

https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=39.6712975,-8.6568081&radius=6000&type=restaurant&key=API_Key

Response:

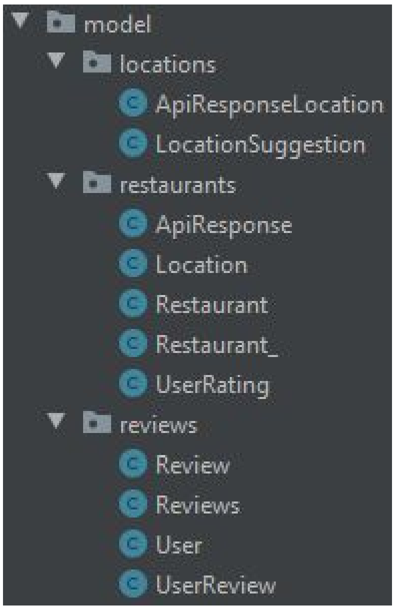
```
{
   "html_attributions" : [],
   "next_page_token" :
"ATtYBwLiedLguKMUoGKmW5y86wNpSq9VxvhtwVIEcUsceyW8QBbe7_NNWjUjPmvBkMGW9qjv_EJ7G34gx4EgF86
LfL_0QxlHMINmMeduIXWiBO-gnkf-kNvfMrGyDfVen1PSx9So6WBaBd4W5kkpnhZb96X_UjZ43iQHu9wBthZX3po
fX6E_xVEcdllXbXLs8Ep3yqwlJBwRHKJl0730hvP92HDtjLrt6nrJP67E_CV-o9NK9yjn4GxX93_rZodexLp0THi
uUyEn5NQy-FvI7MljDWCU1rqM_fh4ZF_gG7--8gknEmM5QVy1FrBw2YLcu6MIDklTZlmRLxJLVYA6xEsZcbCX6WC
gMe4GxJhUur_mMk3zCsR6lJVRMQaS5EEKGtUqYuu7ICMVRCdbXjx-vhFxhTPeQ3m9d7WZt-YGELnSzIN71vzQGbM
_BdngHAld",
   "results" : [
      {
         "business_status" : "OPERATIONAL",
         "geometry" : {
            "location" : {
               "lat" : 39.6310031,
               "lng" : -8.676138099999999
            },
            "viewport" : {
               "northeast" : {
                  "lat" : 39.6323961802915,
                  "lng" : -8.674482919708497
               },
               "southwest" : {
                  "lat" : 39.6296982197085,
                  "lng" : -8.677180880291502
               }
            }
         },
         "icon" :
"https://maps.gstatic.com/mapfiles/place_api/icons/v1/png_71/lodging-71.png",
         "name" : "Hotel Fátima",
         "opening_hours" : {
            "open_now" : true
         },
         "photos" : [
            {
               "height" : 1119,
               "html_attributions" : [
                  "\u003ca
href=\"https://maps.google.com/maps/contrib/104427945437439882485\"\u003eHotel
Fátima\u003c/a\u003e"
               ],
               "photo_reference" :
"ATtYBwLPQIhhp7r-eu_Vv3Mgz1e0aFck-QLo94Lcxek6CFOuvkWLy2WajfZMDoXrMrls13Ch0_ca-52uB_g4XmL
EhxjoA8JIZDcgDLPhVupOdJ2AvaAM3bm4IlQaBZ2utgasmCGKeCYFJHIYmuW32bDigUu3qETceHBO774b7CukVqK
55ilO",
               "width" : 1990
            }
         ],
         "place_id" : "ChIJWWJq4TudGA0RV9pWxqeofsQ",
         "plus_code" : {
            "compound_code" : "J8JF+CG Fátima, Portugal",
            "global_code" : "8CFHJ8JF+CG"
         },
         "rating" : 4.3,
         "reference" : "ChIJWWJq4TudGA0RV9pWxqeofsQ",
         "scope" : "GOOGLE",
         "types" : [ "lodging", "restaurant", "food", "point_of_interest",
"establishment" ],
         "user_ratings_total" : 681,
         "vicinity" : "Rua João Paulo II, Fátima"
      },
```

We use this information to create a list of markers so that when a user goes to the **Map section** he can see all restaurants near him marked.

This JSON file contains better information about each restaurant near each user but we opted for Zomato API because using Places API we didn't have enough permissions to make the number of queries we want. (it's limited).

Data models

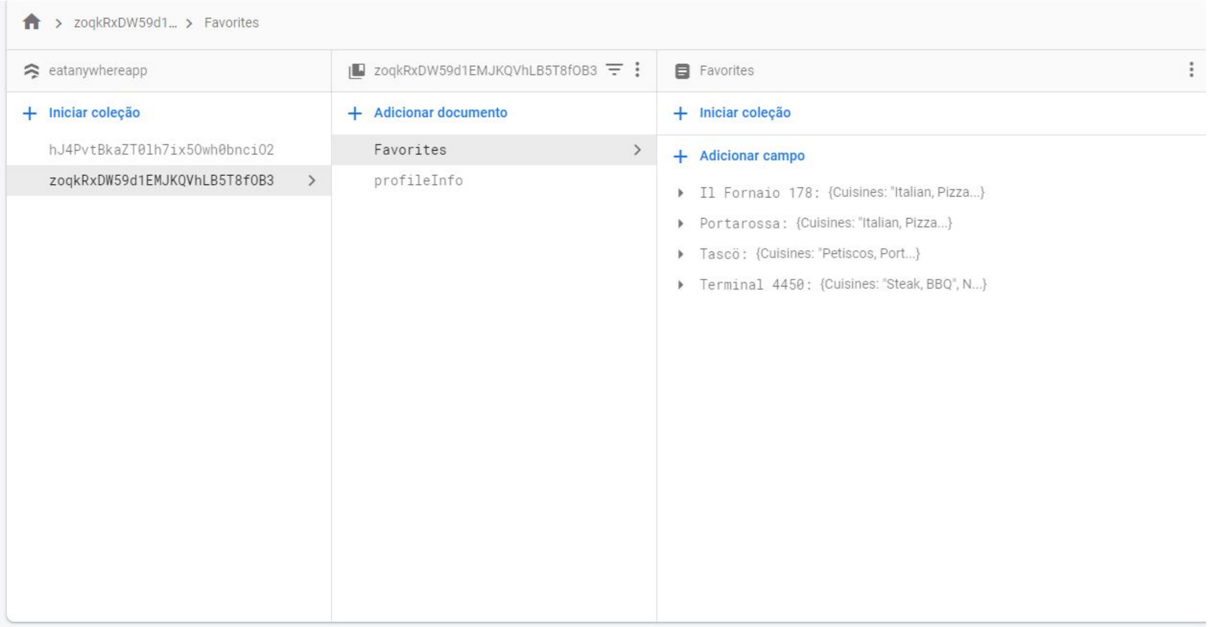
Because of the nested JSONs, we were forced to define all these classes, otherwise we couldn't use the data properly.



```
model
  locations
    ApiResponseLocation
    LocationSuggestion
  restaurants
    ApiResponse
    Location
    Restaurant
    Restaurant_
    UserRating
  reviews
    Review
    Reviews
    User
    UserReview
```

Firebase

Firebased was used as our database first to authenticate a user and then to save profile info for each user and also a list of favorite restaurants.



zoqkRxDW59d1… > Favorites

eatanywhereapp

hJ4PvtBkaZT0lh7ix50wh0bnci02

zoqkRxDW59d1EMJKQVhLB5T8fOB3

zoqkRxDW59d1EMJKQVhLB5T8fOB3

Favorites

profileInfo

Favorites

Il Fornaio 178: {Cuisines: "Italian, Pizza...}

Portarossa: {Cuisines: "Italian, Pizza...}

Tascö: {Cuisines: "Petiscos, Port...}

Terminal 4450: {Cuisines: "Steak, BBQ", N...}

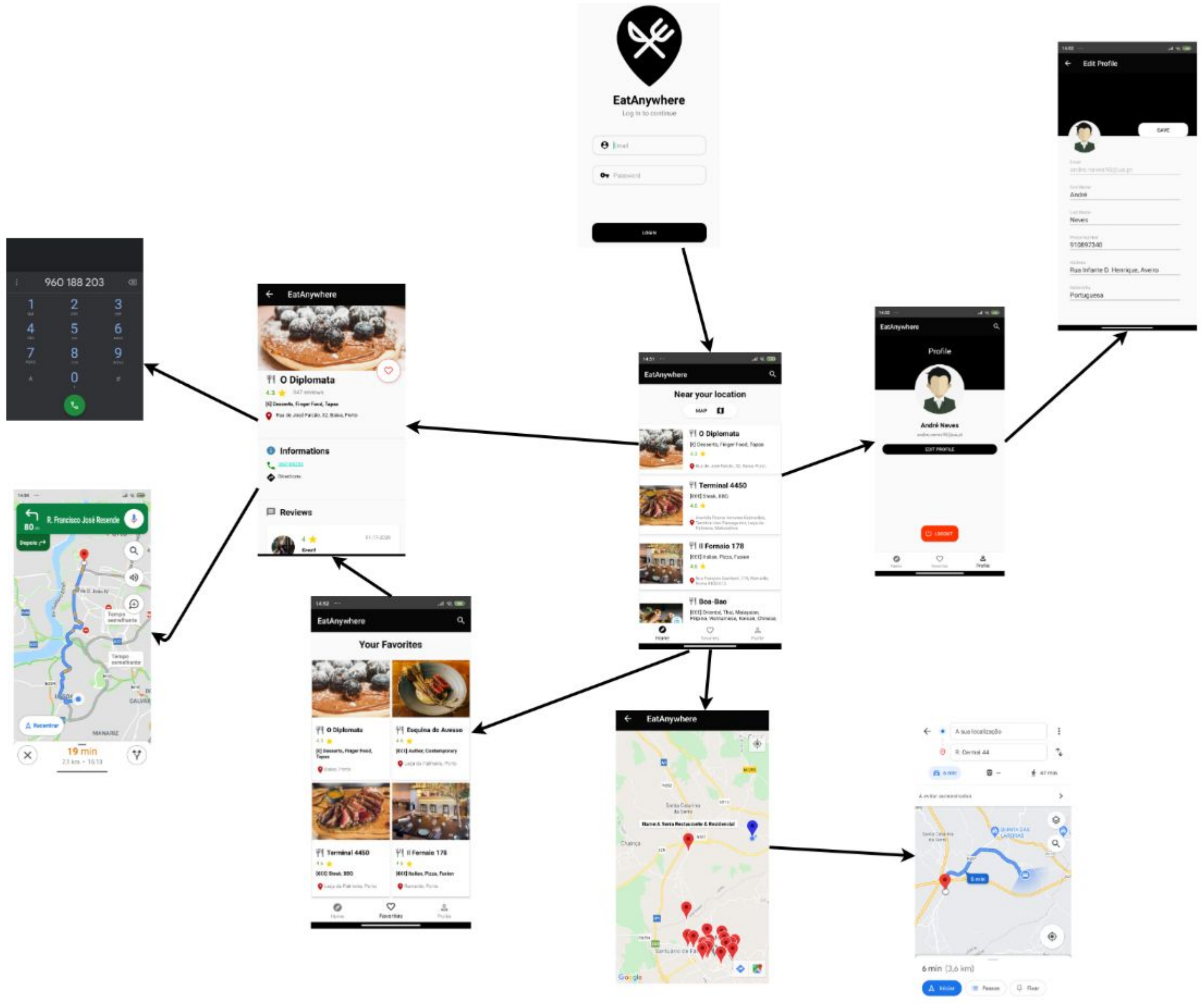
Fig 2 - Firebase structure

Each collection represents the user ID and in each user ID we have two documents one to save the profile Info and another one to save the list of a user's favorite restaurants.

- **ProfileInfo**
    - Stores personal information about the user that can be changed in the application
    - If a user doesn't have information it will appear a pop up that will inform the user to add his profile information
    - Displayed in profile section
    - Profile information contains:
        - Address
        - First name and last name
        - Phone number
        - Nationality
        - Email

- **Favorites**
    - Stores favorite restaurants added by the user
    - User can remove or add them in favorites page or home page
    - Each restaurant represents a list that contains all the information about it including a list of reviews
    - Used in favorites page and in each restaurant details to check if restaurant belongs to favorite list or not

Implemented interactions



Project Limitations

In this topic we can cover points such as time, APIs and research. First of all, given the current situation we are facing and taking into account every course unit plus thesis, our work load is much higher than normal, meaning that we consider that our time was very limited. Second of all, limitations concerning the google maps API are out of our control, but still was a feature limitation, in some way. And last, the time that we spent researching solutions to our problems and tutorials took a huge part of our effort. The features that are missing and we had planned were sorting by cost, rating and searching restaurants on a user specified location. Nevertheless, we are happy with the final result as we present a simple UI with a minimalistic design and a perfectly working app with external APIs, database and sensors integration.

3 Conclusions and Resources

Lessons learned

- Google Maps problems:
    - To search automatically restaurants(and mark them) nearby using our current location we had to create an account with billing options because we didn't have enough permissions without that. We tried with Google Maps Places API, Directions API , Geolocation API , Maps static API but all without success. To accomplish that we made a http request to Google Maps that retrieves a json with multiple information about the restaurants nearby.
- Zomato API problems:
    - Correctly mapping the data into objects.
- Design problems:
    - A lot of effort is put into design. Multiple difficulties but we were able to solve them all.

Key project resources

- Code repository: https://github.com/andreneves98/EatAnywhereApp.git
- Ready-to-deploy APK: in repo
- Video Demo: https://www.youtube.com/watch?v=MzAC6-Pyj4s

Reference materials:

- Zomato API: https://developers.zomato.com/api?lang=pt
- Firebase Firestore Database: https://firebase.google.com/docs/firestore
- Google Maps API: https://developers.google.com/maps/documentation/android-sdk/start

Contributions:

- André Neves: 60%
- Dany Costa: 40%