A picture containing text

Description automatically generated

ecycan

developed by team 6 – Itamar Shor, Yuval Alon and Omer Gil

Encouraging environmental care by making it easy to find colorful recycling bins.

****

# Overview

**Server**

Bins’ router

Users’ router

Bins’ DB agent

Users’ DB agent

Bins’ DB

Users’ DB

**Client**

Map View

My Account

Recycle Info

UI

APIs

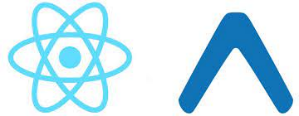
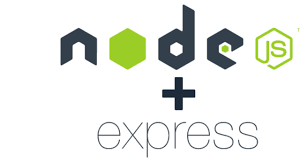
OpenCage geo translation

Google Maps

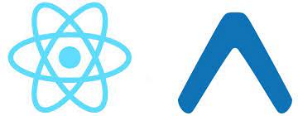
Google Sign-in

Server communication

REST API agent



**React Native**



**Expo**

**+**

**React native**

**Server**

**Server**

**Expo**

**React native**

User DB document:

|  |
| --- |
| User |
| * name * email * score * avatar * token * google user id |

Bin DB document:

|  |
| --- |
| Point |
| * coordinates =  [longitude, latitude] |

|  |
| --- |
| Bin |
| * bin type * street name * location * missing reports |

As shown in the above block diagram, we used React Native with Expo for the frontend and NodeJS with express for the backend. For the databases we used mongo DB.

# Main APIs

* **Geo-translation APIs** (using OpenCage geocode API):
  + GET: ServerDomain/location/coordinates.  
    Translate address to coordinates. The address is given as query parameter.
  + GET: ServerDomain/location/address.  
    Translate coordinates to address. The coordinates are given as query parameters.
* **Bins related APIs**:
  + GET: ServerDomain/bins/getBin/<type>/.  
    Retrieve all the bins of a certain type from the database.
  + GET: ServerDomain/bins/getBin/<distance>/.

Retrieve all the bins within <distance> radius from a location. The location (coordinates) is given as query parameter.

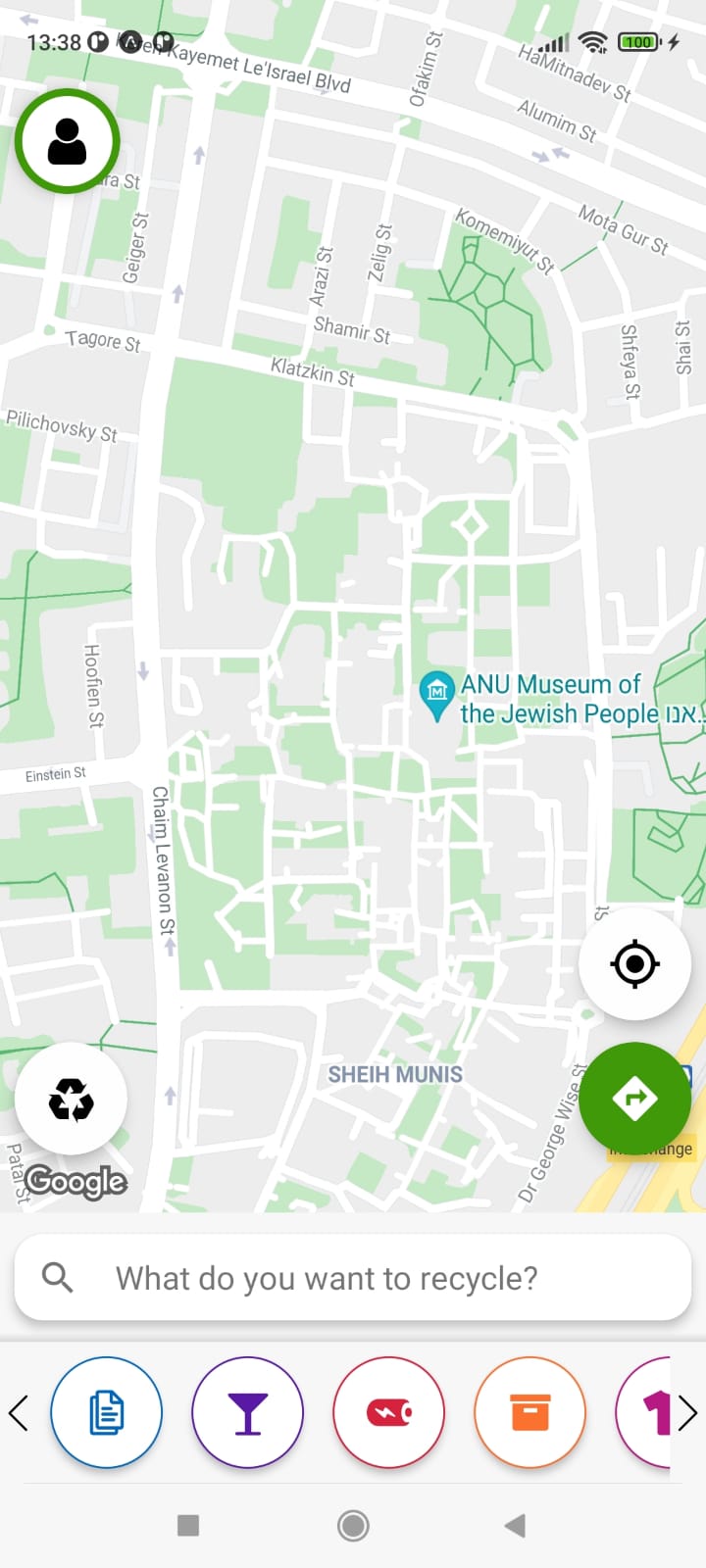
* + POST: ServerDomain/bins/reportMissingBin.  
    Report missing bin and update the database accordingly. The report is made by passing the bin (unique) database id as query parameter. Note – a bin will only be deleted after 10 users reports.
* **Users related APIs**:

All the requests, but login, require a logged-in user. To make sure the request is valid, it must contain an Authorization header with the token returned from the login request. Without the header or a valid token, the request will fail.

* + GET: ServerDomain/users/me.  
    Get the user’s full name, recycling scores and avatar from the database.
  + POST: ServerDomain/users/login.  
    The input is the Google user id generated in the frontend with **Google sign-in API**.  
    Before the requests is being processed, the Google user id is validated by a middleware which uses **Google auth API**. The login request generates a new token with **JWT** and sends it back to the user.
  + POST: ServerDomain/users/logout.  
    Deletes the user’s token.
  + PATCH: ServerDomain/users/me.  
    Modify the user’s data. Only the avatar and score are modifiable.

# Usage Instructions

* **When opening the app**:



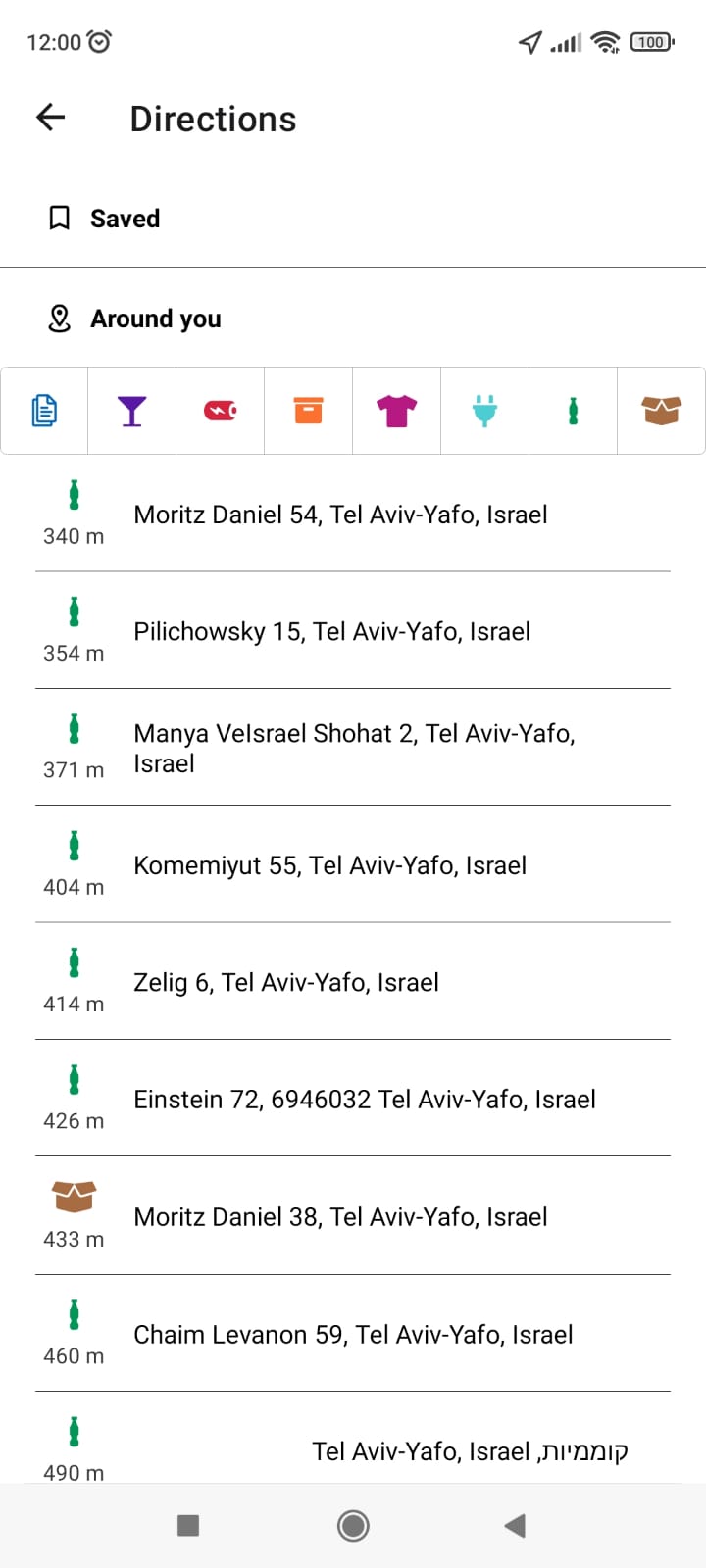
Go to My Account

Go to Recycle Info page

Select bin type to show on the map

Navigate to closest bin (by type)

* **When clicking on the navigation button**:

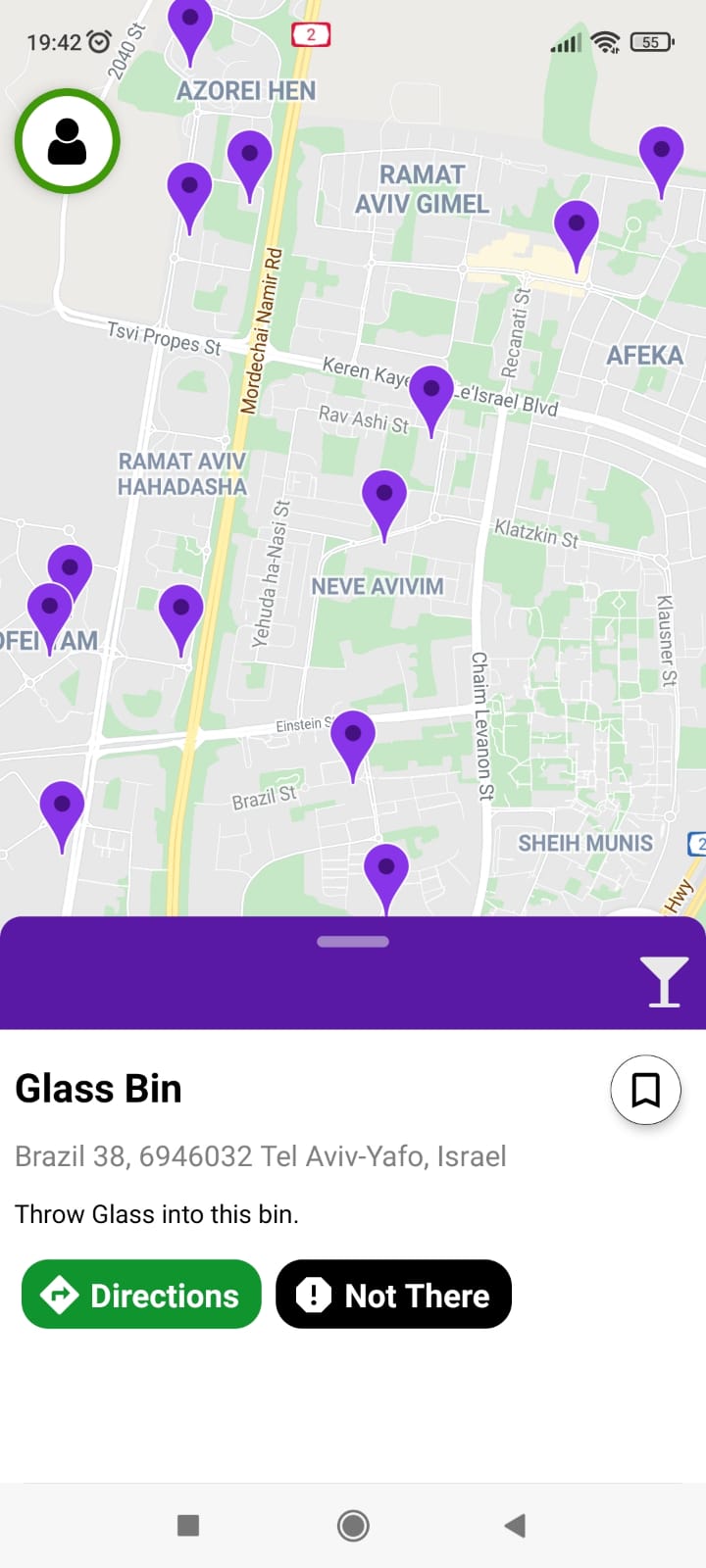


Navigate to closest bin (by type)

Filter according to type.

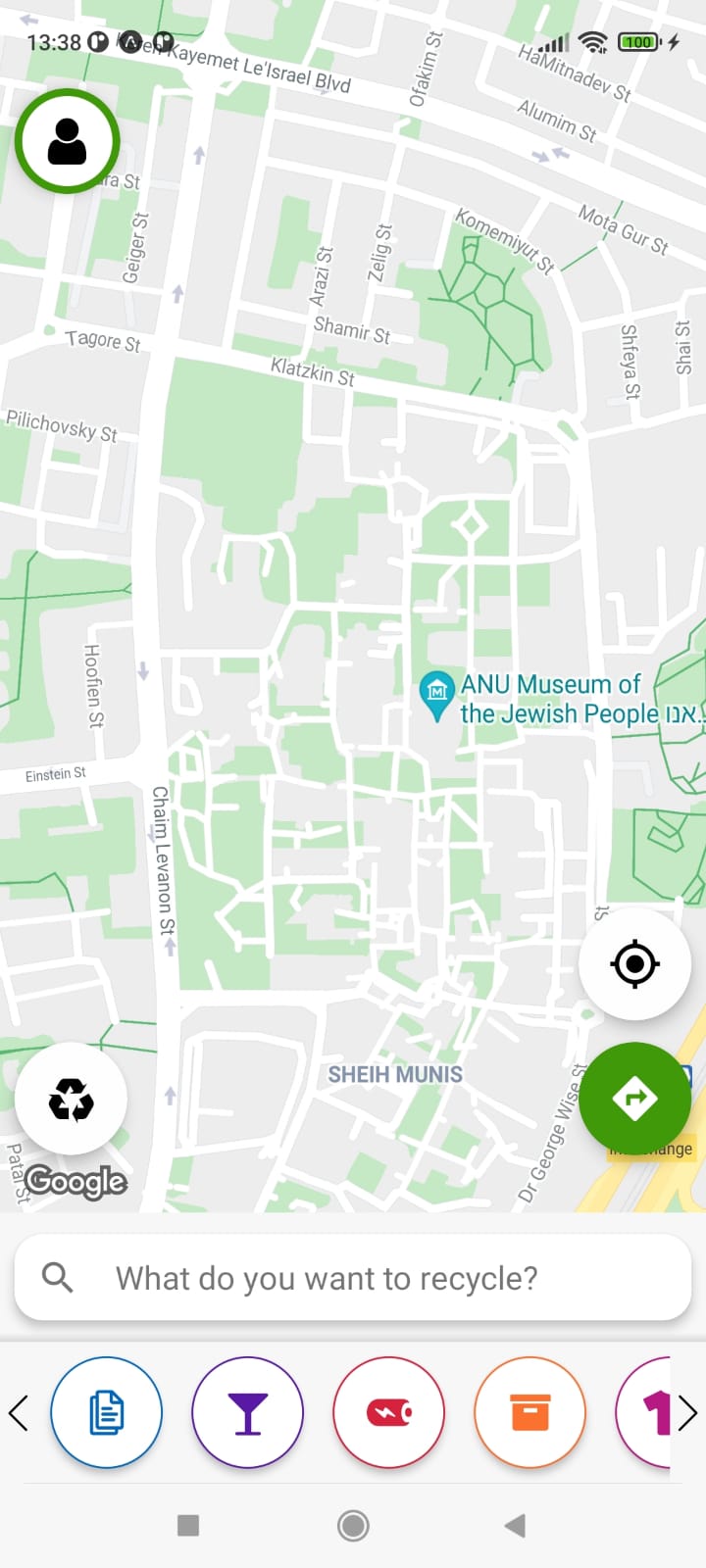
List of all the nearby bins by distance

* **When clicking a marker on the map:**



Report missing bin. If enough users report it – then it will be removed from the DB.

Get directions to this bin.



Go to My Account

