



BYTE MY CODE

# WHERE IS THE POWER ?

Demo 4





# Overview

Where is the power, is a mobile and desktop app that navigates users through these dark times in South Africa. Loadshedding is inconveniencing a lot of South Africans, especially on the roads. Where is the power aims to assist South Africans avoid traffic, plan their day, find areas that have electricity, inform communities of surprise power outages in local areas and more.

# Meet the team



**Andreas Visagie**



**Amber-Leigh Lezar  
(Team Lead)**



**Daniel Radloff**



**Jaco Malan**



**Tumi Pare**



# Contribution



**Lourens Snyman**

University of Pretoria Geography, Geoinformatics & Meteorology

Advised and provided spatial data  
for suburbs



# Technology Choices



**Figma** - To design wireframes and mockups And for planning purposes.



**Angular Ionic** - Develop Front-End for mobile and desktop web apps.



**Mapbox** - Will be used for having a Realtime map.



**Rust Rocket** - API, Is type safe and guarantees speed in runtime



**MongoDB** - A flexible database system for spatial data



**GitHub** - Used for version controlling the repository.



**Firebase Hosting** - Hosts the frontend application



**AWS Hosting** - Hosting the application backend

# Use Cases

*As expected from the client*

**Real time map overlay**

**Historical Data**

**User reporting**

**Statistics dashboard**

*WOW FACTORS*

# Research

## Geospatial Data

- Not readily available.
- API's like google maps only offer to a ward level.
- AfriGIS offers a paid solution to get suburb level data.
- The data that was obtained, is from 2011 (OUTDATED).

## Loadshedding Schedule

- ESP API, is very expensive (Licensing is an issue).
- Municipalities have different ways of display data (pdf, excel and so on).
- Matching Geo-spatial data with loadshedding data is no easy task (Especially with suburbs from 2011)



# Why MongoDB

**Geo-Spatial data**

**Geo-Spatial indexing**

**Document based (works well with Rust)**

**Large Document Sizes**

**Working with dynamic data**

**Ties in well with our (microservice) architecture**

# Wow Factors

**Saved places**

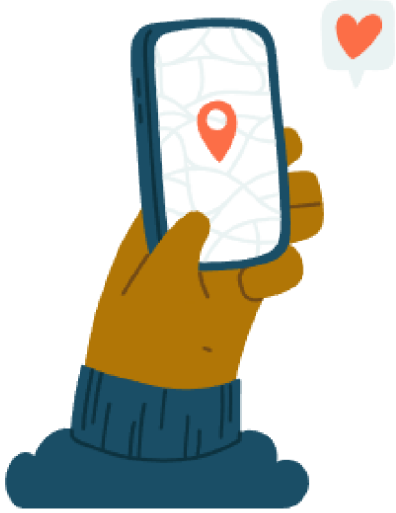
**Swagger API**

**Loadshedding schedule**

**Navigation**

# Saved Places

Saved Places



Saved places is only available to registered users.

[GO TO PROFILE](#)

Navigation bar: Navigate, Saved, Schedule, Statistics, Profile

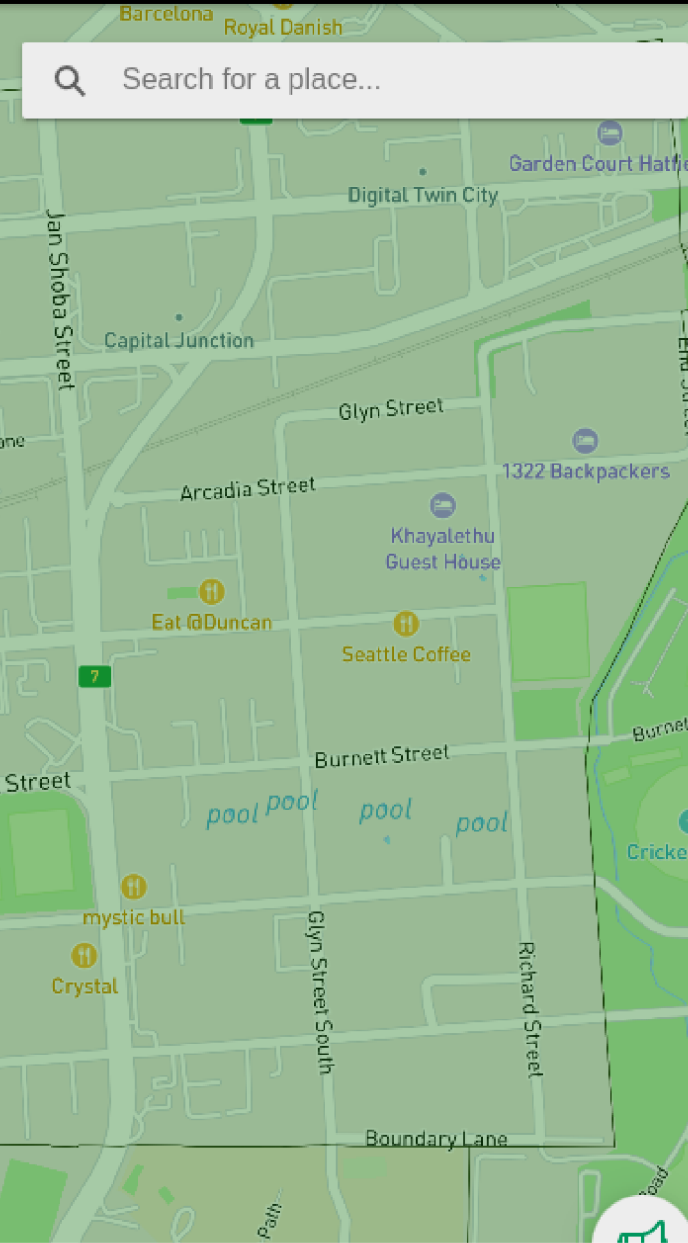
Saved Places

Search for a place...

You have no saved places

Navigation bar: Navigate, Saved, Schedule, Statistics, Profile

Search for a place...



**Hatfield**  
0083, Pretoria, Gauteng, South Africa

[SAVE](#) [DIRECTIONS](#)

Navigation bar: Navigate, Saved, Schedule, Statistics, Profile

Saved Places

Search for a place...

Hatfield, 0083, Pretoria, Gauteng, South Africa

Navigation bar: Navigate, Saved, Schedule, Statistics, Profile



# Swagger API

<https://witpa.codelog.co.za/swagger-ui/>

The screenshot displays the Swagger UI interface for an API. It features a list of endpoints grouped into sections: **loadshedding**, **Map Data**, **Schedule Data**, and **Suburb Statistics**. The **Suburb Statistics** endpoint is selected, showing its details. The endpoint is a **POST** request to `/api/fetchSuburbStats`. It has no parameters and a required request body of type `application/json`. An example request body is shown as `{ "suburbId": 1245 }`. A **Try it out** button is visible in the top right of the endpoint details. The **Responses** section is partially visible at the bottom.

POST /api/auth

loadshedding

GET /api/fetchCurrentStage

Map Data

POST /api/fetchMapData

Schedule Data

POST /api/fetchScheduleData

POST /api/fetchTimeForPolygon

Suburb Statistics

POST /api/fetchSuburbStats

Parameters

No parameters

Request body required

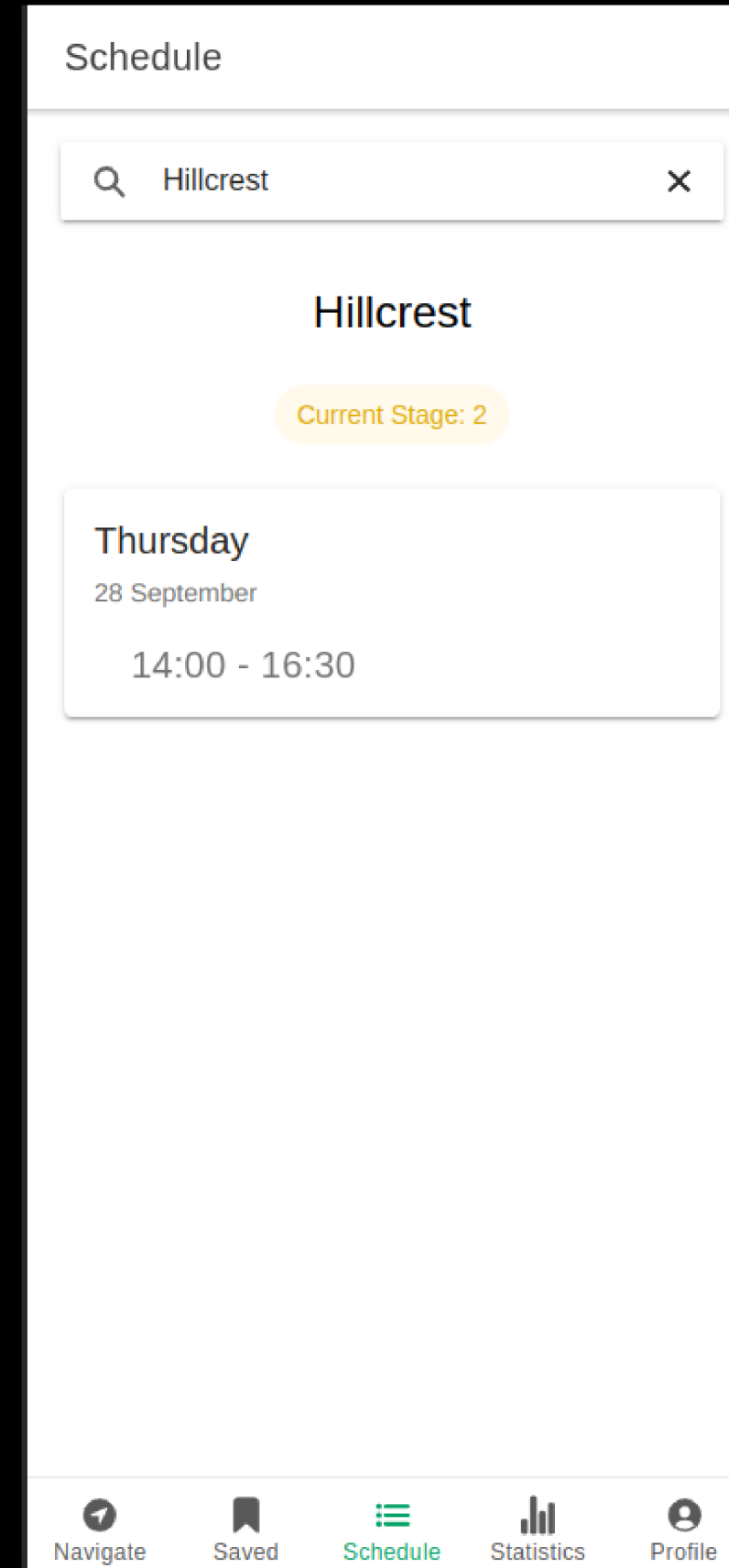
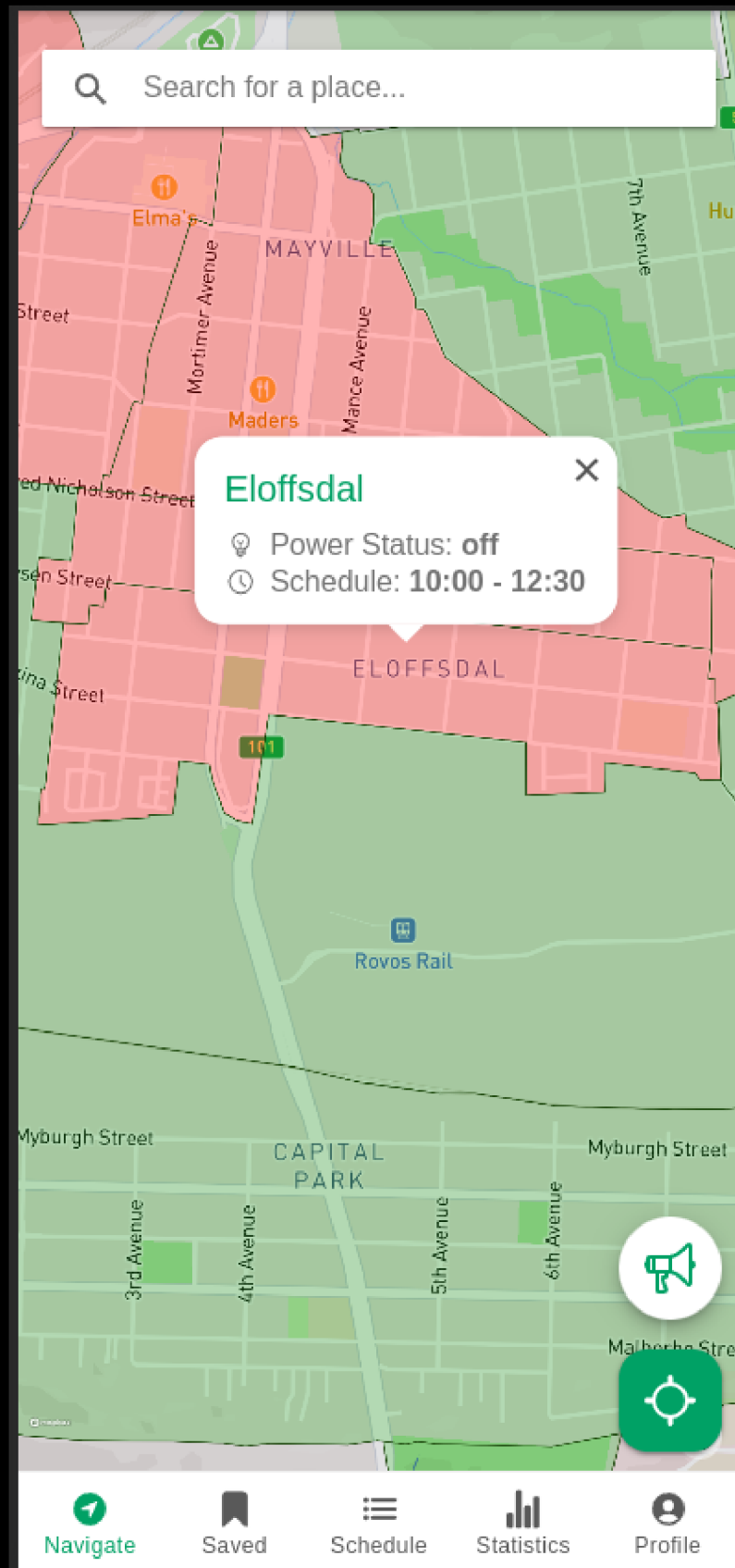
application/json

Example Value | Schema

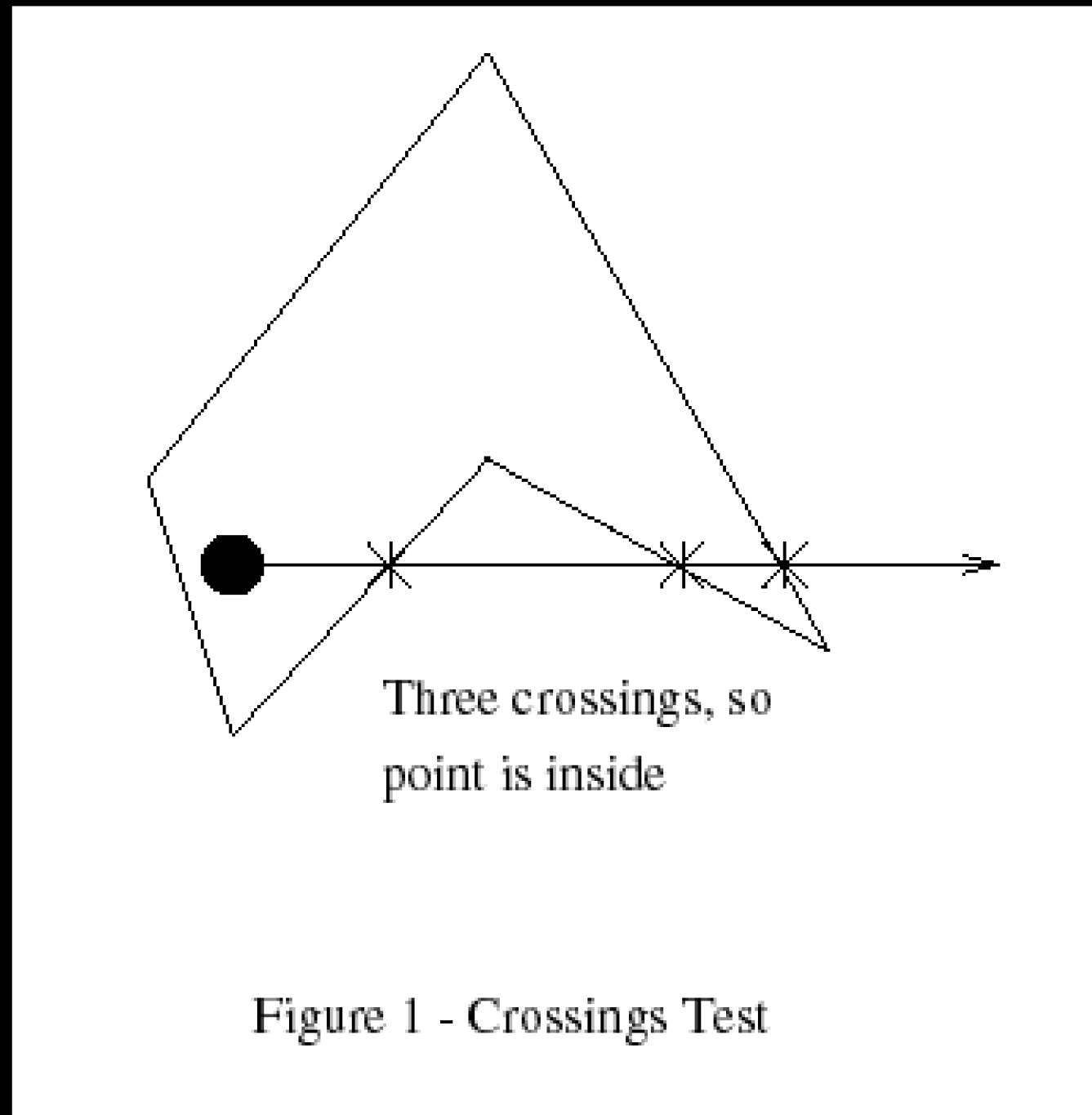
```
{
  "suburbId": 1245
}
```

Responses

# Loadshedding Schedule



# Navigation



Search: Suiderberg, 0082, Pretoria, Gauteng

Distance: 16 km

Duration: 29m

ETA: 11:23

Begin Trip

Search: Suiderberg, 0082, Pretoria, Gauteng

16 km

29 min • 11:23

- Drive east on Ring Road.
- Enter the roundabout and take the 2...
- Turn right onto Herold Street.
- Turn left onto Lynnwood Road/M6.
- Turn right onto Lynnwood Road/M6....
- Bear right toward M3.
- Turn right onto Nelson Mandela Driv...
- Turn left toward M2.
- Turn right onto M1.
- Turn left toward Soshanguve.
- Take exit 42 onto R55 toward Brem...



# Architecture

## DECOMPOSITION STRATEGY

This strategy breaks the system up into smaller factors which makes reasoning large complex problems easier to maintain. This strategy will also allow for ease of reusable code since the system will be decomposed into smaller parts.

## MULTITIER ARCHITECTURE

- Presentation Tier
- Logic Tier
- Data Tier

## MODEL-VIEW-CONTROLLER(MVC)

- MVC is mainly used for ease of control when working with GUIs (Frontend related).
- This will decompose the frontend by having **models, views** and **controllers**.
- This decomposition allows for views and controllers to be reusable

## MICRO SERVICES ARCHITECTURE

- Decomposing the system into different services
- Since each endpoint on the system runs on a thread it is possible to spin up many instances of a service to be served.
- It is also possible (given the budget) to distribute each API endpoint it's own service

# Design Patterns

## RAII Guards (Resource Acquisition is Initialisation)

- use `std::sync::{Mutex, Arc}`;
- use `std::rc::Rc`;

## Fold - Apply an algorithm across an entire collection.

- use `std::iter::Iterator::`  
`map, fold, for_each`  
`};`

## Temporary Mutability - Variables can be made mutable, and then re-declared immutable.

### Interior Mutability

- use `std::cell::{Cell, RefCell}`;

## Builder - Instantiate an object using a set algorithm, but let the user pick the parts one by one.

- use `rocket::Build`;

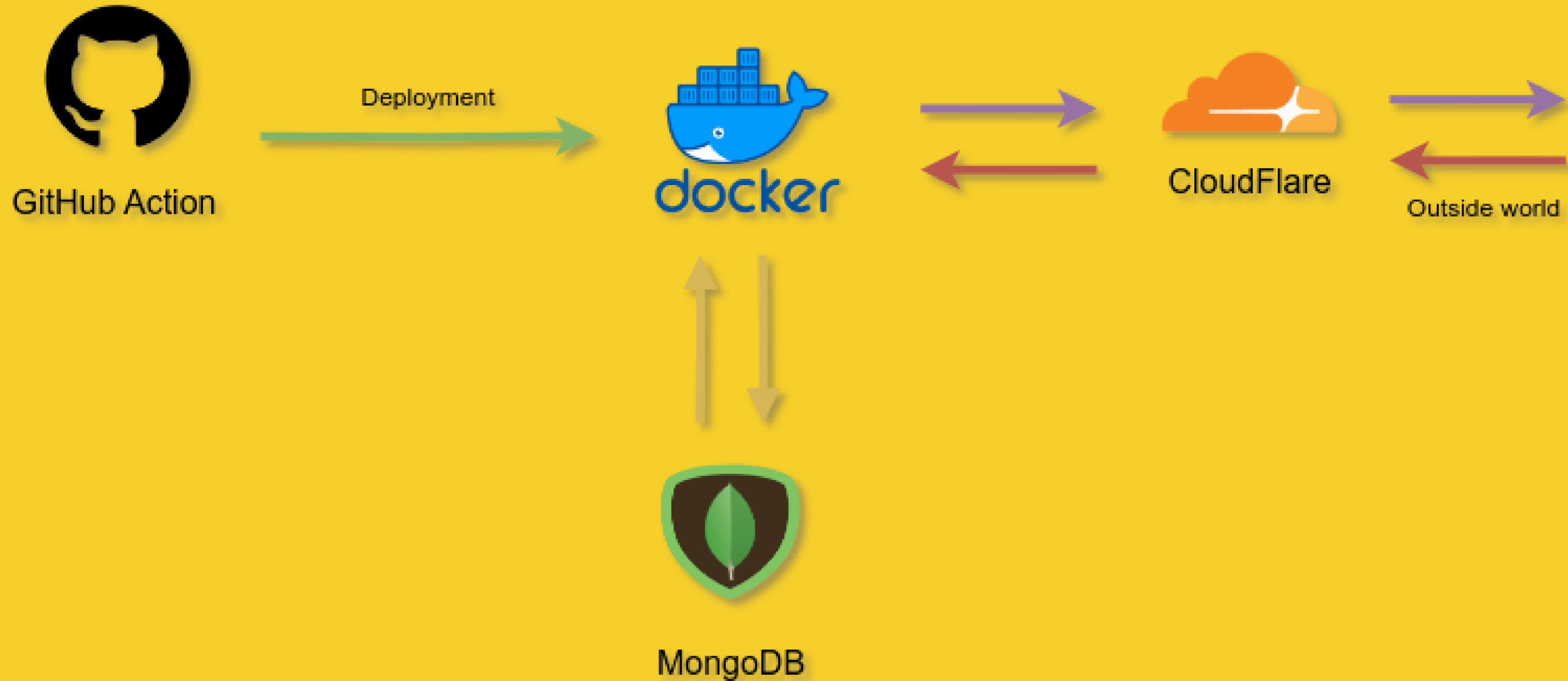
# Design Patterns

**Observer** - Allow subscriber(s) to receive updates from a subject. Used to provide state management in Angular (RXJS)

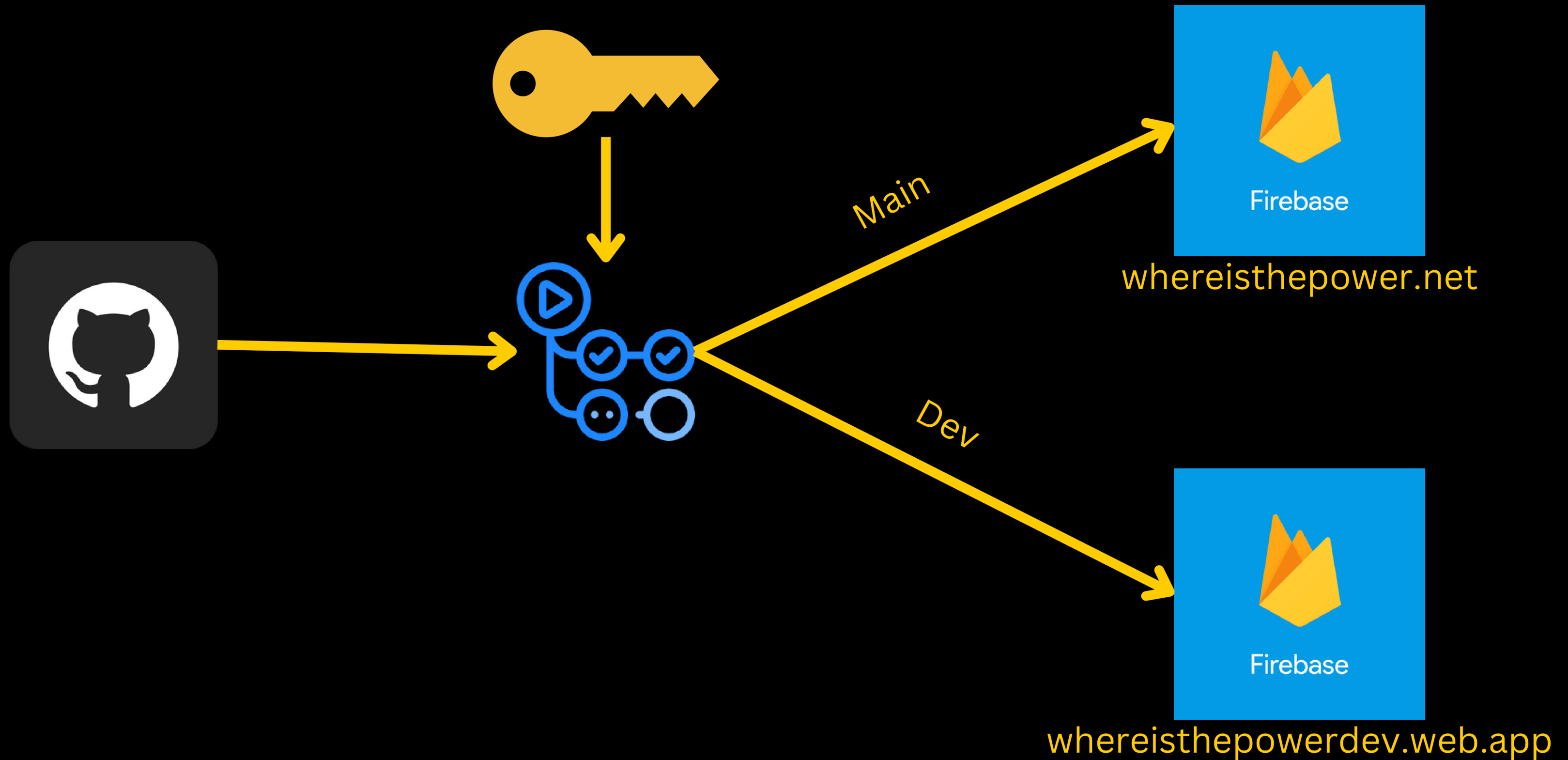
**Dependency Injection** - Create dependent objects outside the class, allowing loosely coupled code, which is easily maintainable and extendable. Angular Services.



# API Deployment



# APP Deployment



# Quality Assurance

Usability testing

User Feedback

Regular feedback reviews with client

Security

Efficient and Performant system

# API Testing

No mongo Driver for rust during testing, out of this project's scope

Rust is relatively new. Codecov can't give a good estimate for the code that we wrote.

- Business logic is tested in rust, endpoint and mongodb queries are covered by end-to-end tests.  
Business logic is kept outside of the context where the rust driver is not available.
- Calls to the database use an external function (Which can be mocked, but not tested).
- Rust compiler implicitly cover edge-cases.

# APP Testing

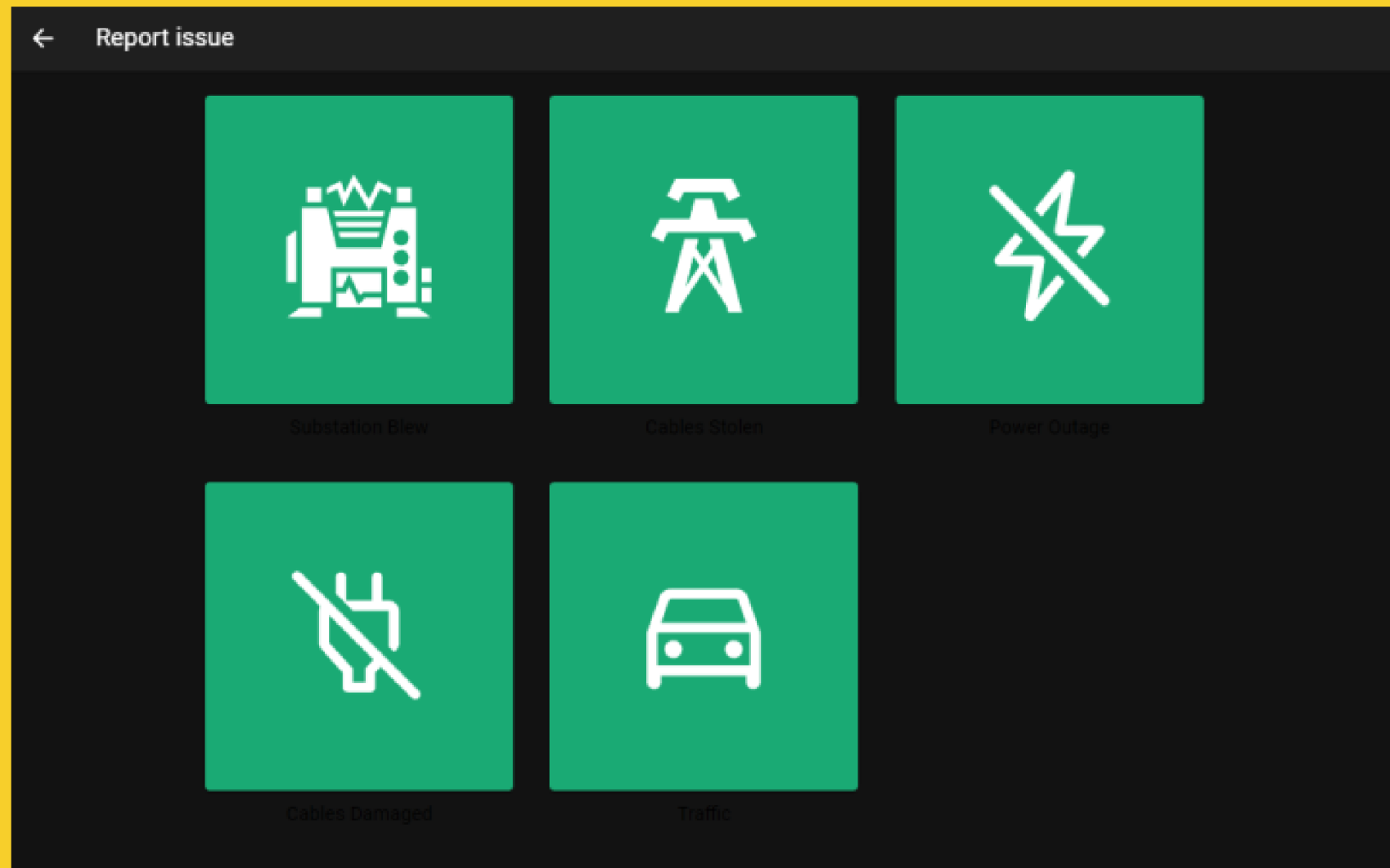


Constant usability feedback  
from the team

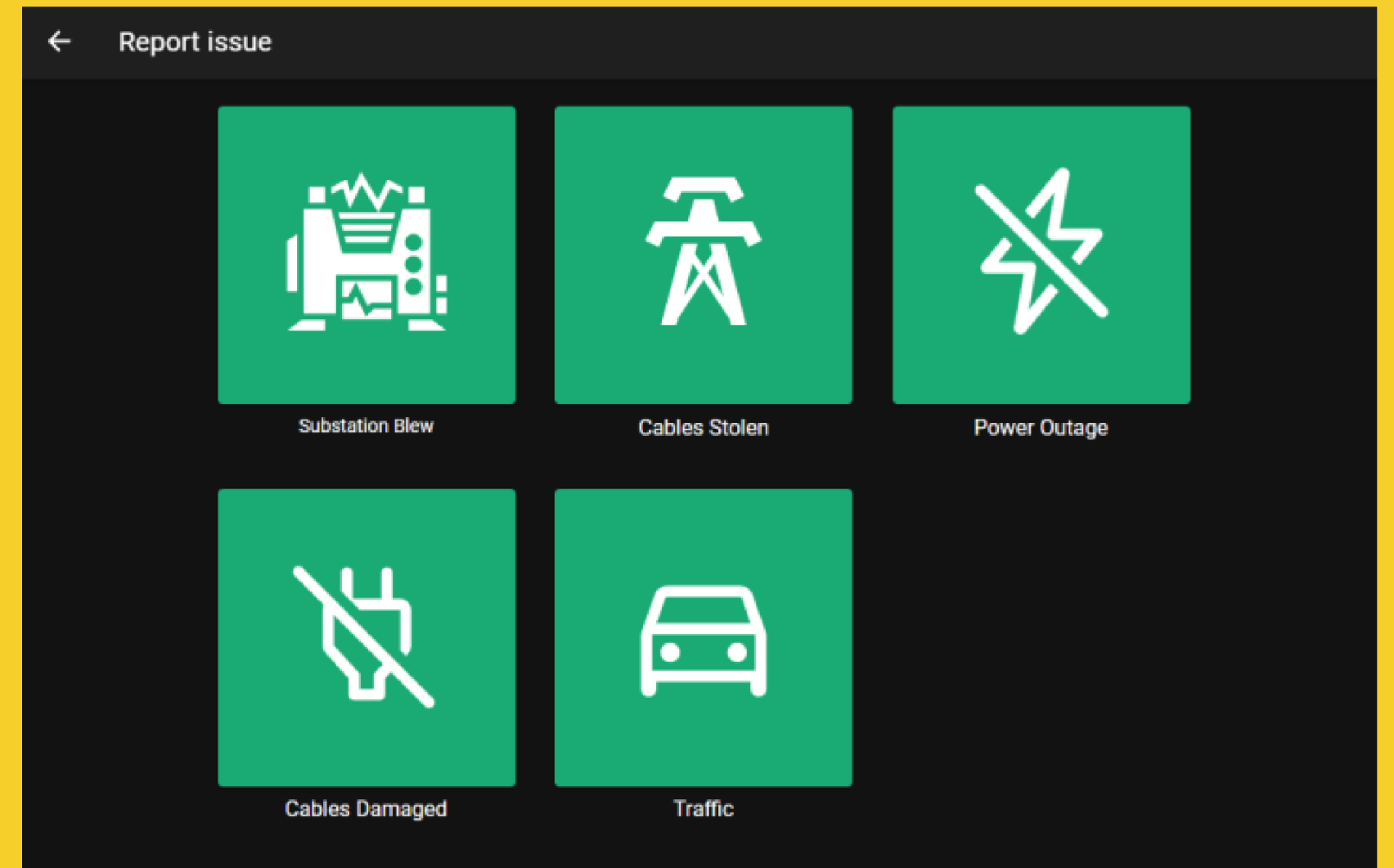
```
- name: Run Tests
  working-directory: ./app/WhereIsThePower
  run: npm run test:coverage

- name: Upload coverage report
  uses: codecov/codecov-action@v3
  with:
    name: frontend coverage-report
    fail_ci_if_error: true
```

# User Feedback



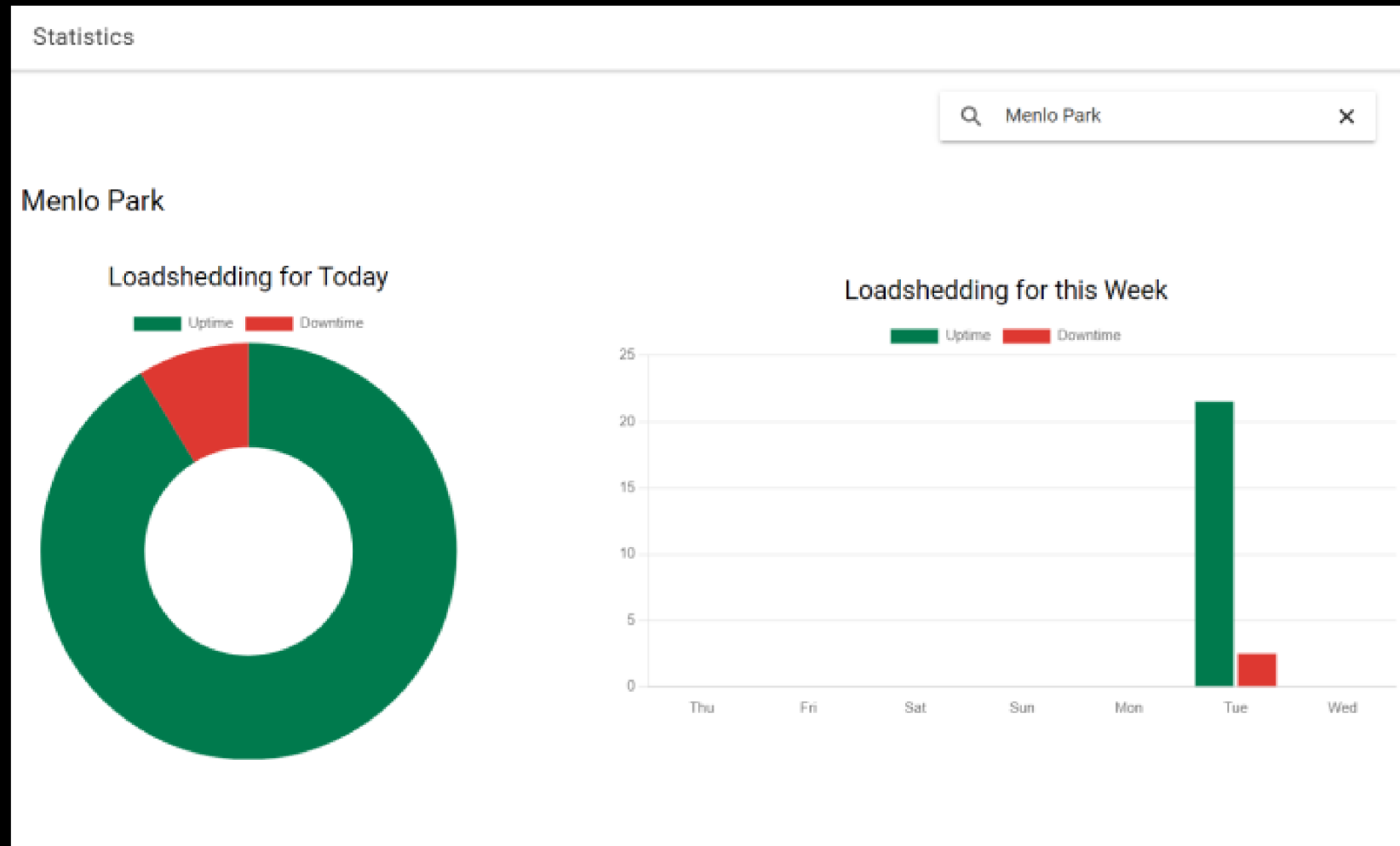
**Before**



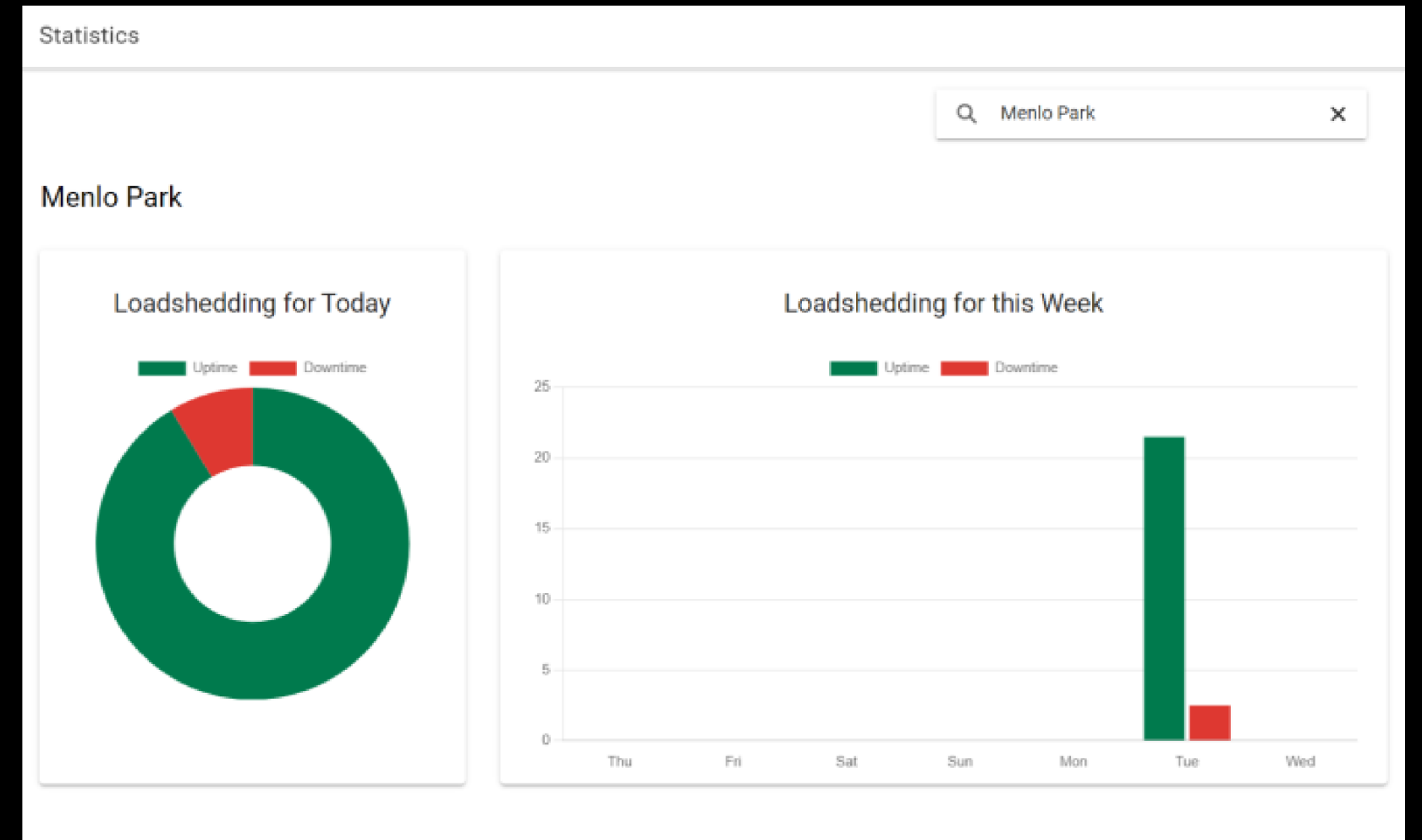
**After**



# User Feedback



Before



After

# Real World Use

- Users are able to find faster routes and plan their trips accordingly.
- Since Where is the power is available on any device, anyone leaving work can quickly check their device to avoid major traffic.
- Communities can report problems in their area, and since the app requires one's location, you can trust the reports that do come in.
- Emergency services can make use of this software to enable more efficient transport through the loadshedding chaos and save lives.

# Real World Use (Dev)

- The API is publicly available (As long as the server is up and being paid for) for other developers to develop on wards.
- All code is open source
- Swagger allows other developers to test our endpoints and potentially migrate some of these endpoints in their customs software

# Demo

 whereisthepower.net

