# PokéMate

romantic adventures for your Pokémons

Alex Wolf 12-526-216
Simon Bachmann 14-709-893

# Outline

Mockups

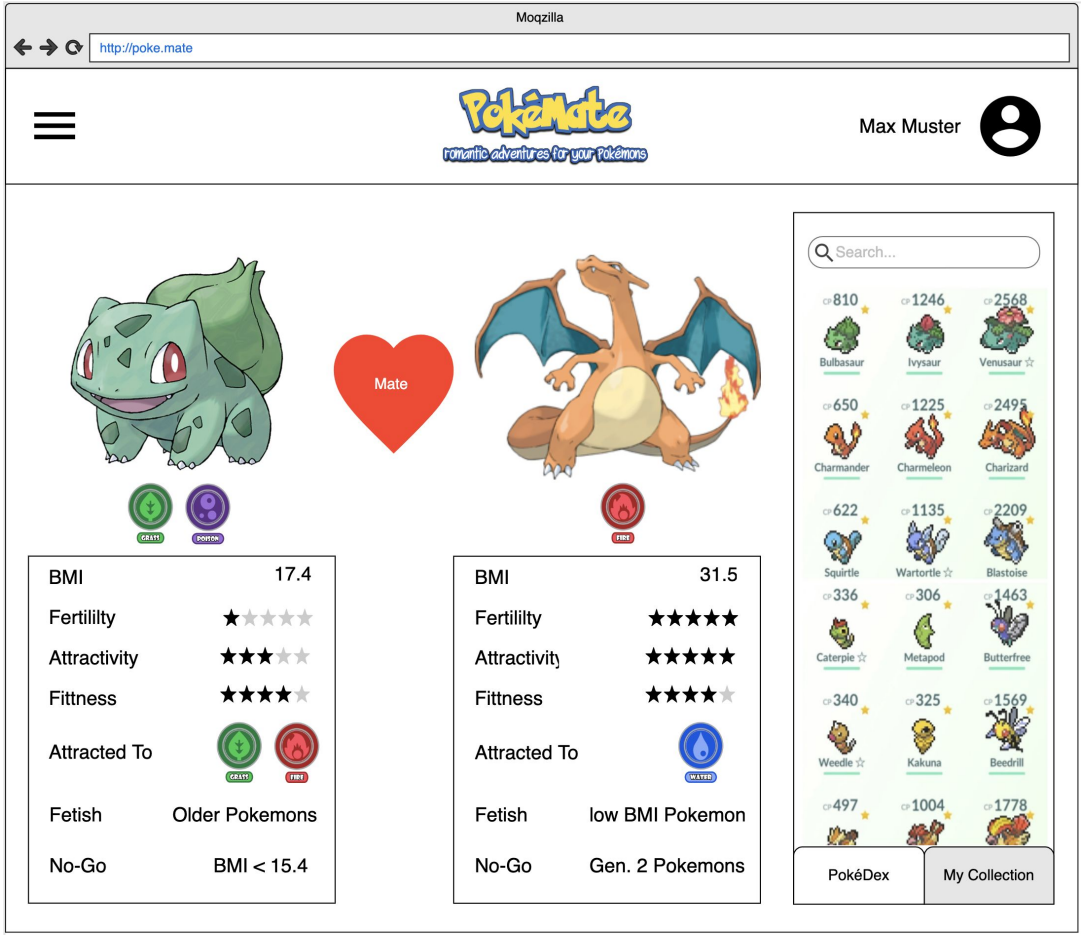
Technologies

Pokemate Environment Overview

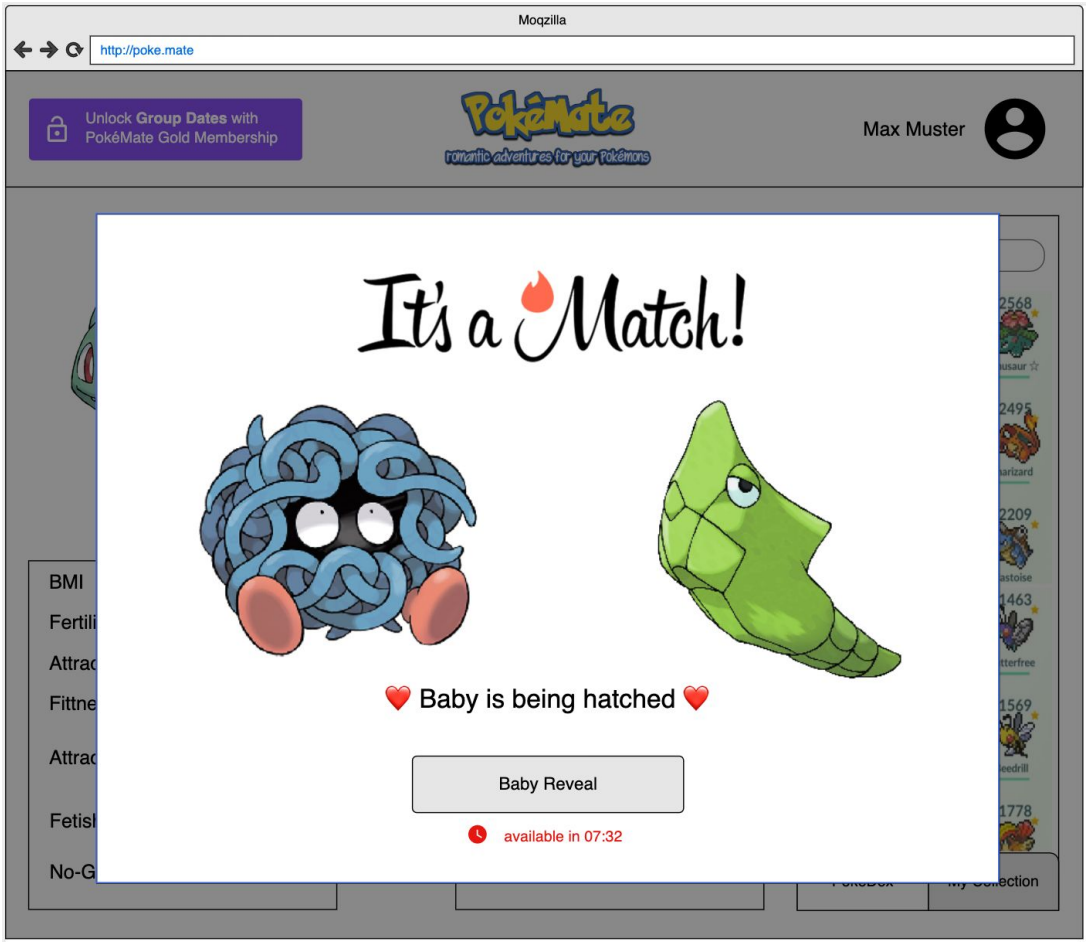
Github Progress/Walkthrough

Github Actions

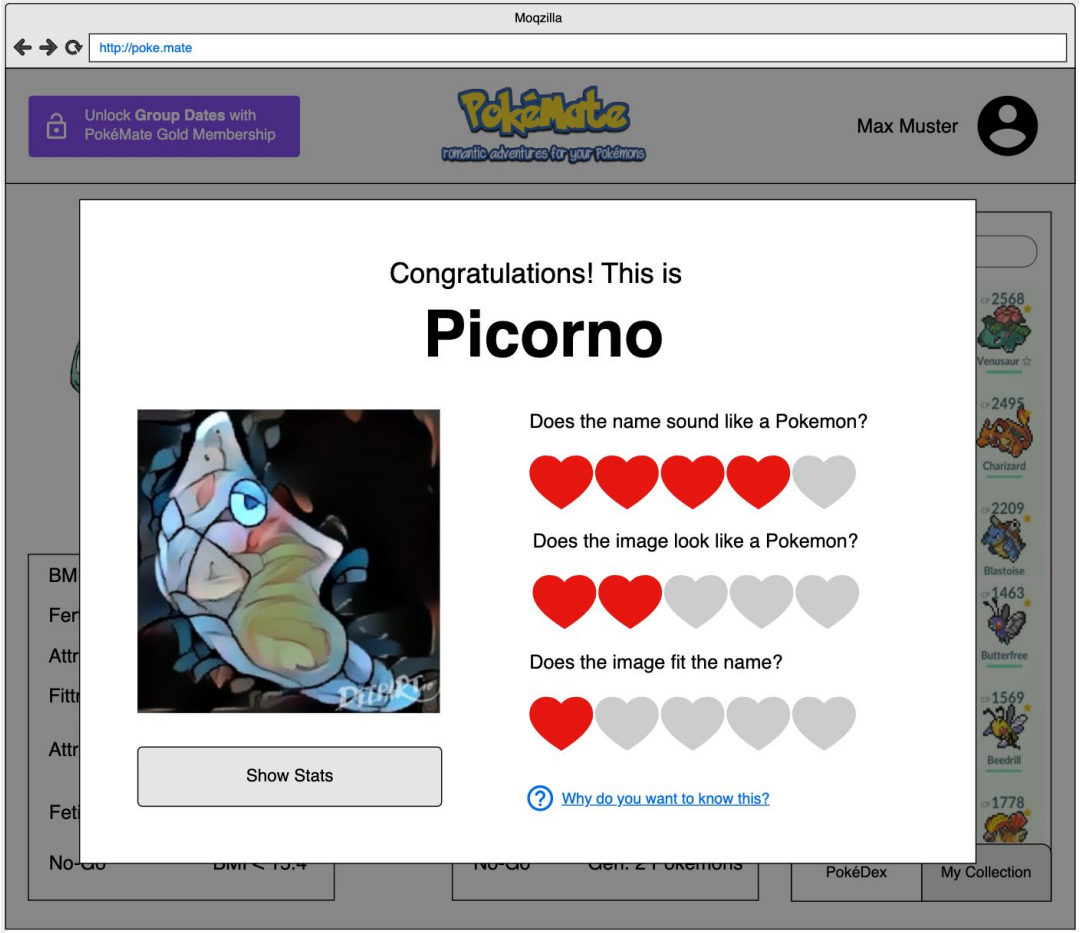
Next Steps




# Mockups - Home




Mockups - Match (creating an image takes some time - GPU intense)




Mockups - Baby (Rating is stored to improve the image and name creation algorithm)

# Technologies

## Kanban Board

Organisation with unified Github project across all repositories

## Continuous Integration

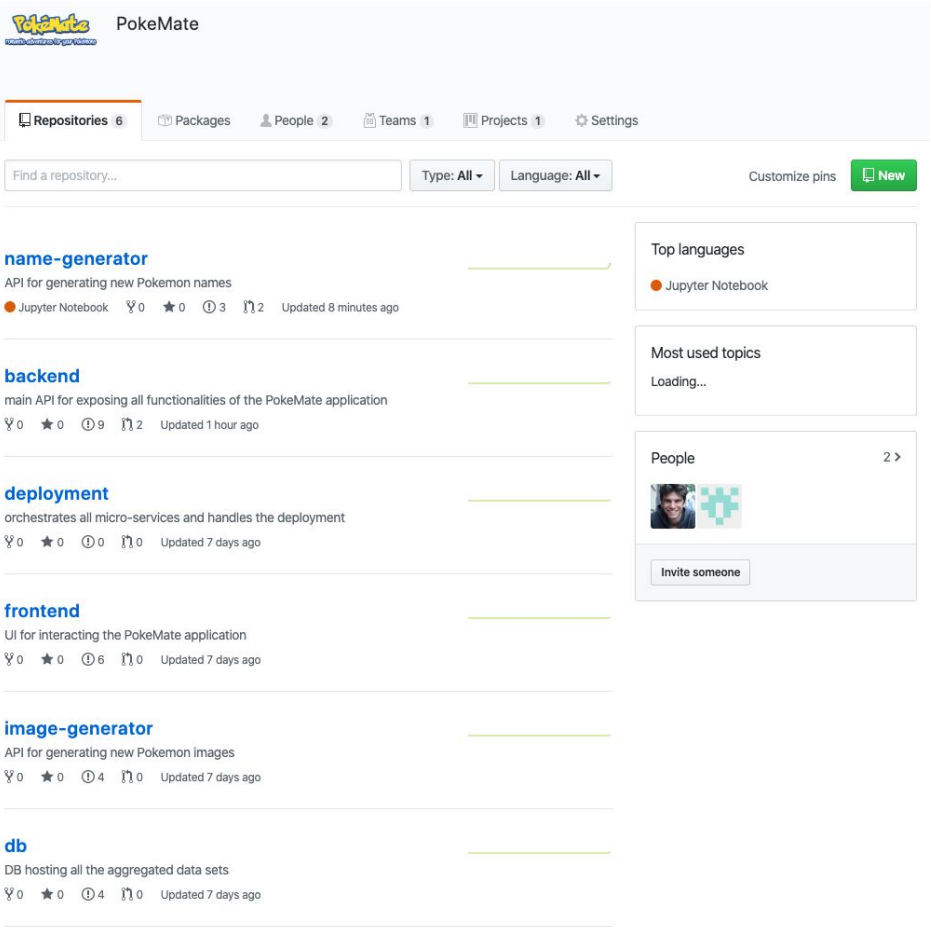
Github actions

## API Documentation

Swagger

## Quality Checks

Sonar Qube




# Github Organisation to Unify All Repositories

# PokeMate Environment Overview

## All

All containers should be accessible by name

## Frontend

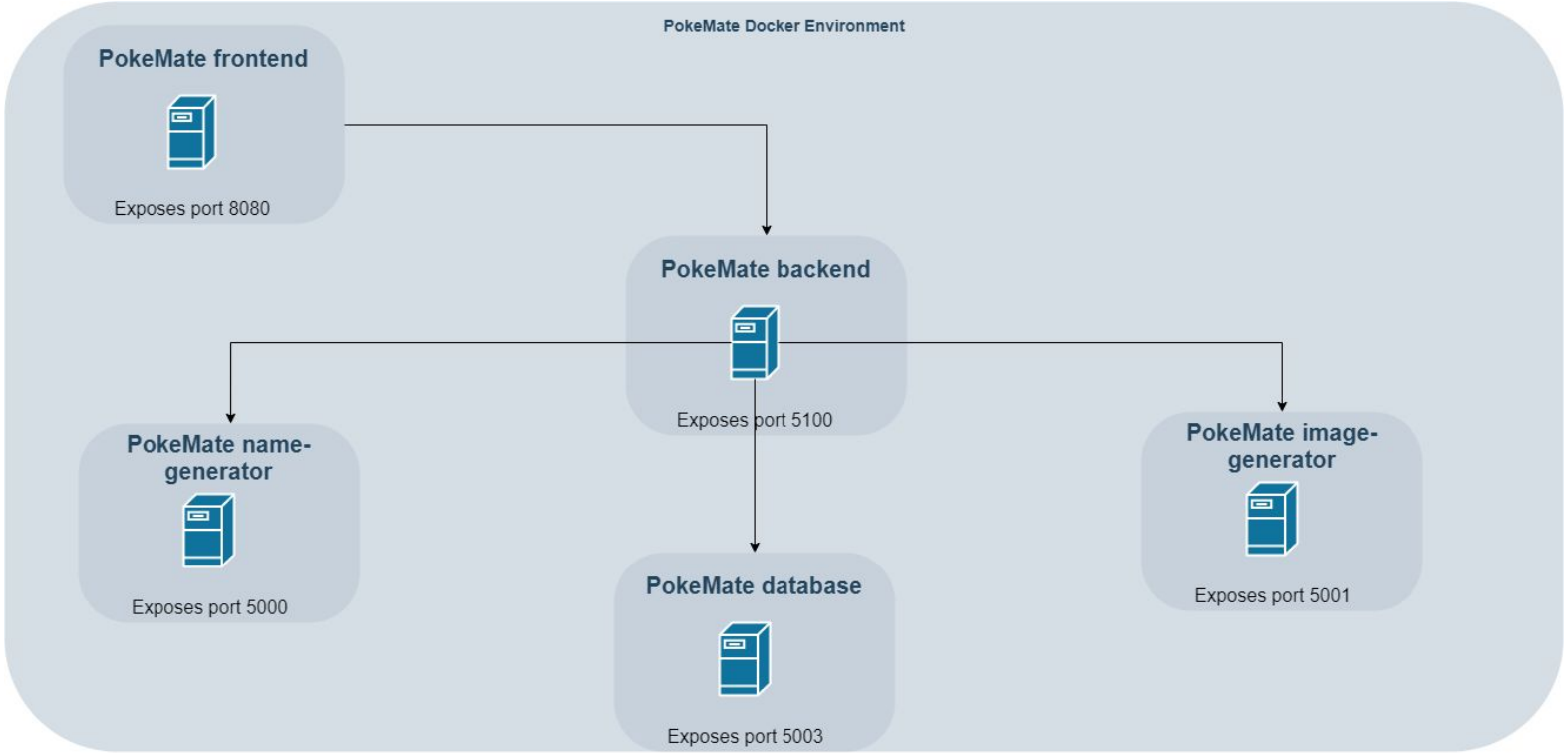
Frontend containers use ports from 8000 upwards

## Externally exposed

Externally exposed containers use ports from 5100 upwards

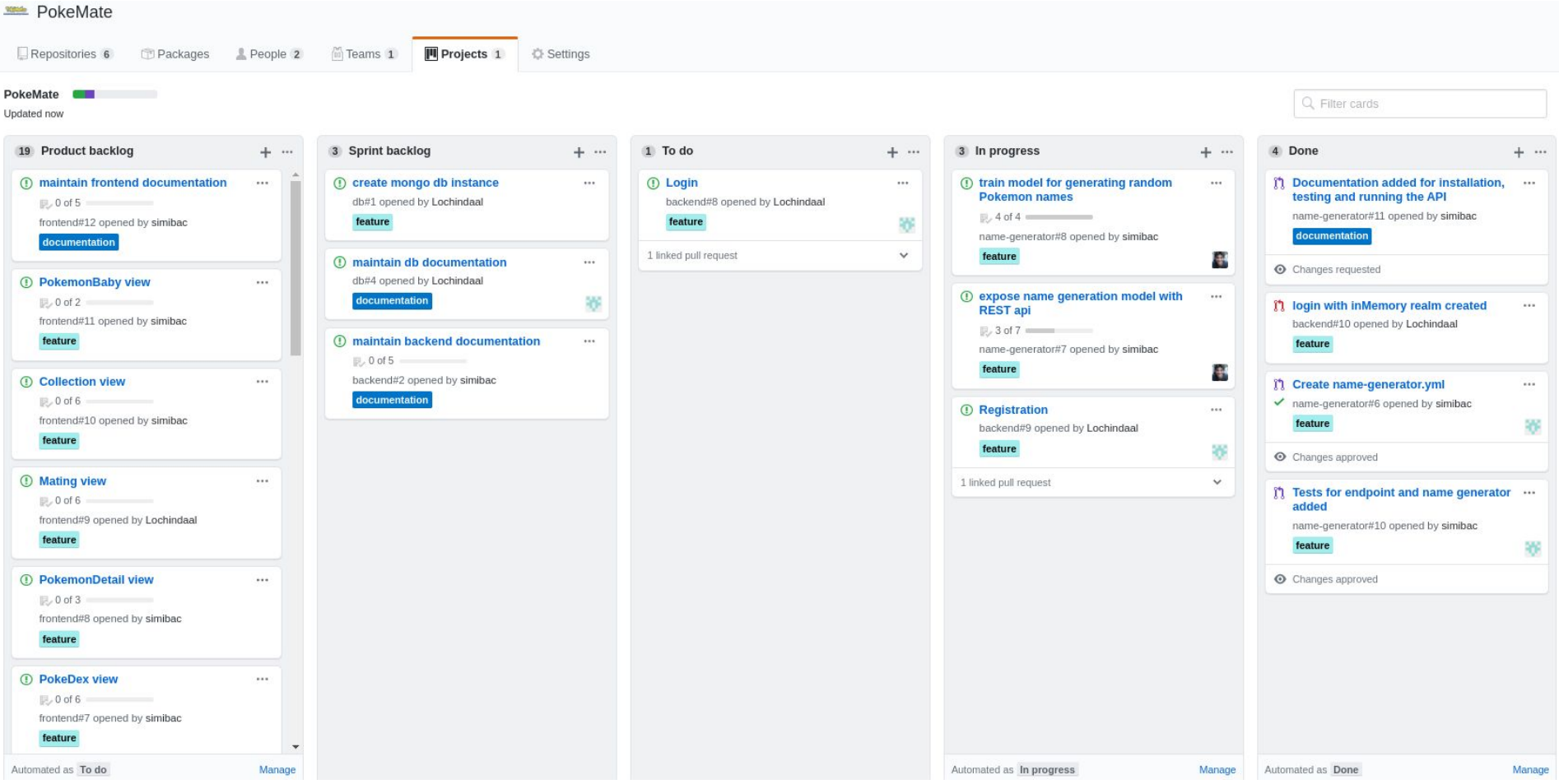
## Internally

Internal containers use ports from 5000 to 5099




## PokeMate Docker Environment

**Note** | Externally exposed containers have ports from 5100 upwards




# PokeMate Sprint Overview

# Project GitHub Progress

## PokeMate Progress

⚠️See link in email.

# PokeMate GitHub Actions

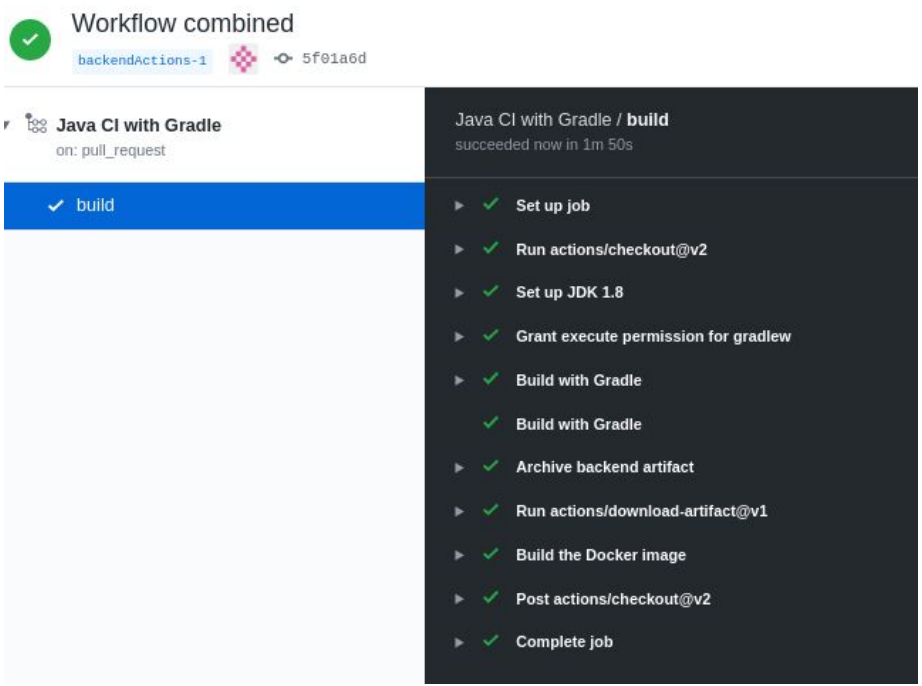
## CI / CD tasks

All CI / CD tasks are implemented using GitHub Actions

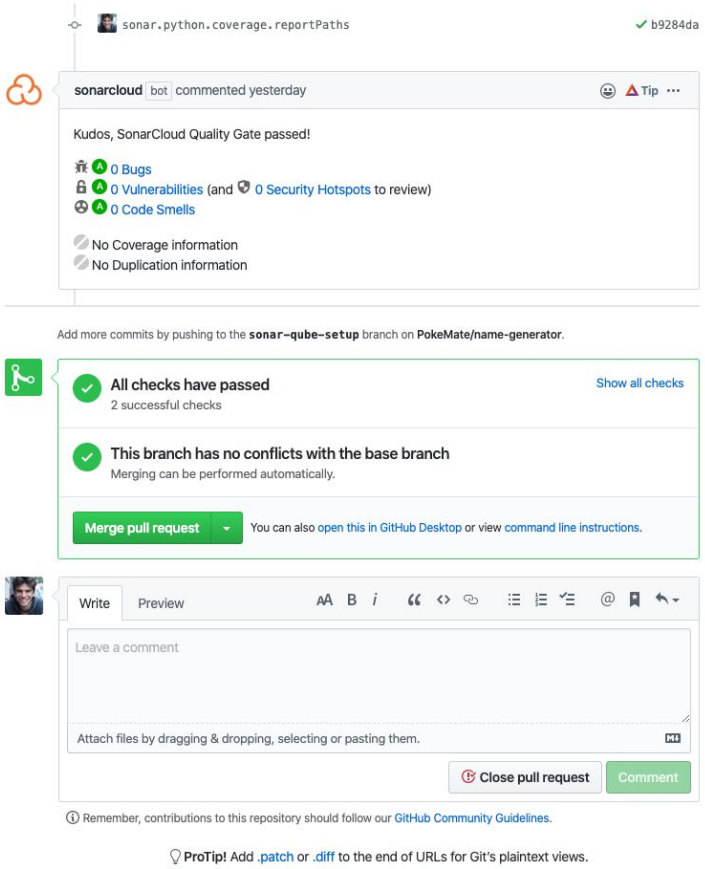
## Reason

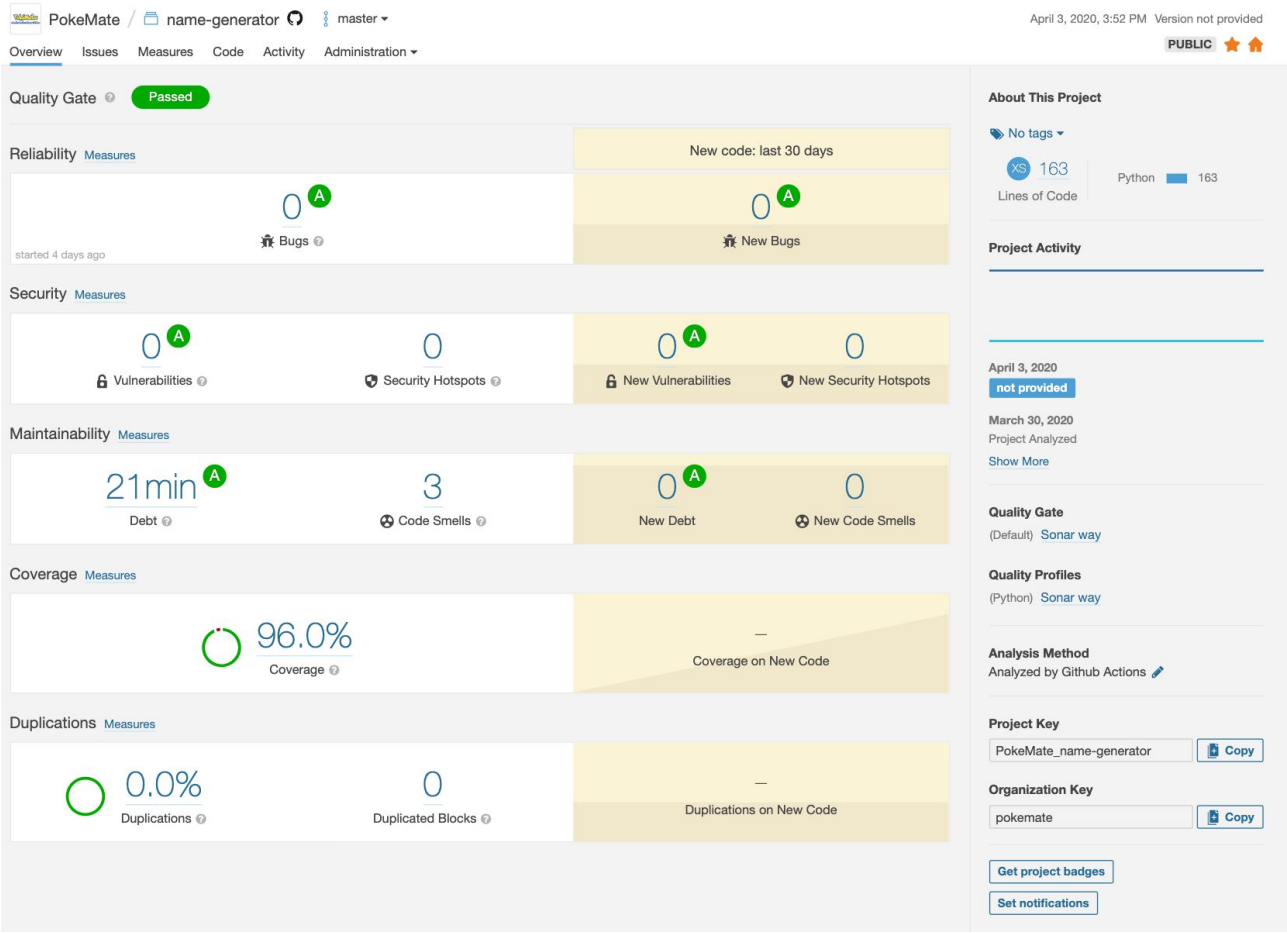
Github as a single point of collaboration for the whole team

# PokeMate GitHub Actions



# PokeMate SonarQube Evaluation






## Sonar Qube Analytics (more here)

## Dates

### Create Date

`POST /dates`

Send two pokemons on a date.

This is a protected enpoint. Make sure you use the JWT in the authorization header :

```
Authorization: Bearer <token>
```

Required JSON data in body:

| Parameter | Type | Required | Description |
|---|---|---|---|
| `id_0` | `int` | `yes` | pokemon id |
| `id_1` | `int` | `yes` | pokemon id |

Example request:

```
curl --location --request POST 'localhost:8080/pokemon' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lI
--header 'Content-Type: application/json' \
--data-raw '{
        "id_0": 1,
        "id_1": 2
}'
```

Example response:

```
{
  "date_id": 343
}
```

**Backend API Specifications (full docs in backend wiki)**

# Next Steps

Define endpoints and implement communication between services

Create docker-compose

Create frontend

Make first version release (master)

# Questions?