POGOH Station Operations Dashboard

Project Specification

1. Overview

The dashboard helps POGOH operations managers and dispatchers monitor bike-share stations in real time. It consumes the POGOH API, stores snapshots for historical charts, and presents an interactive map with KPIs, charts, and filters. Tech stack: Django + PostgreSQL (backend), React + WebSocket/Ajax (frontend), Google Maps JS API (visualisation).

2. Product Backlog (grouped by module)

Module	Action (user-observable behaviour)
Authentication	• Register a new account • Log in / Log out
Station Management	• Add a new station record (admin) • Edit station metadata (admin) • View station list with filter pane
Real-time Data	• Fetch current bike counts for all stations every 60 s via POGOH API • Persist snapshots for analytics
Dashboard Map	• Display all stations on map • Colour icon green / yellow / red for full / filling / empty • Click icon to open station detail
Station Detail View	• Show KPI “Bikes available now” • Show line chart of bikes today (00:00 → now) • Highlight KPI with same colour rule
Comments / Feedback	• Post comment about a station • List comments chronologically
Performance & DevOps	• WebSocket push to React when new snapshot arrives • Docker compose for local dev • Unit tests ≥ 70 % coverage

3. First-Sprint Backlog (Sprint 1 ends Wed Oct 29, 2025)

Backlog Item	Assignee
Create Django models and migrations	Alondra Robles
Connect to POGOH API, store snapshots	Alondra Robles
Build *comments* form component (React)	Sanjana Panwar
Build *login / registration* React page	Riley Weng
Implement Django views & serializers for auth and comments	Sanjana Panwar
Build *new-station* admin form	Sanjana Panwar

Product Owner (Sprint 1): Alondra Robles (arobles)

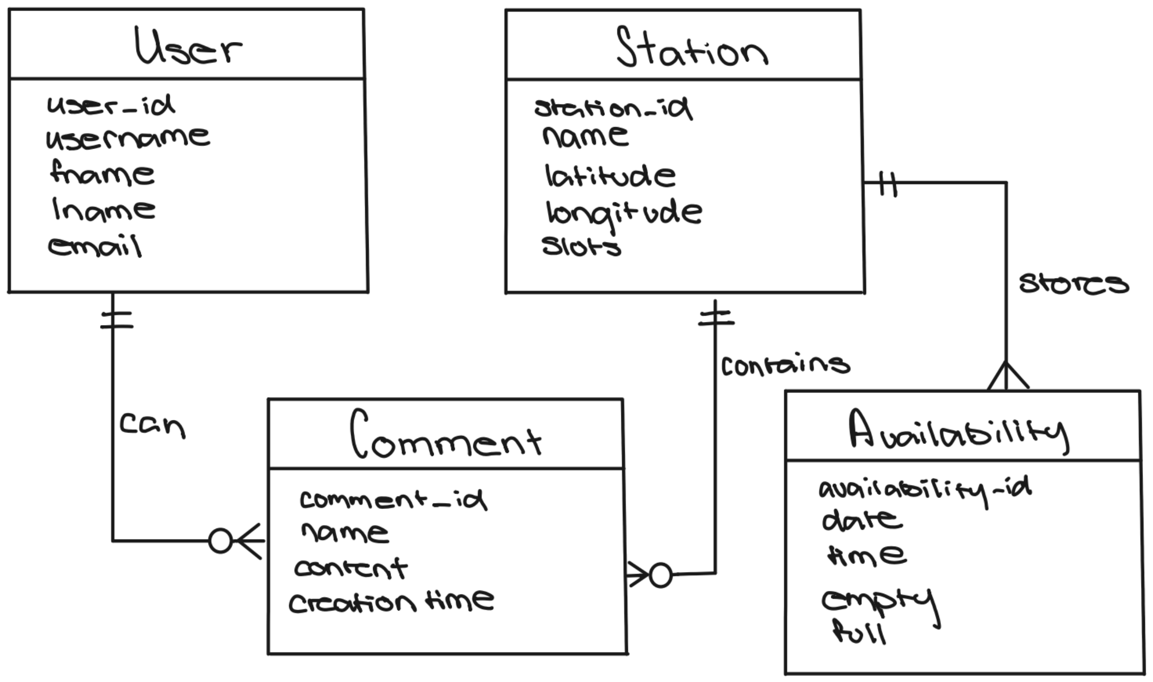
4. Data Model (models.py)

Python
```python
from django.db import models
from django.contrib.auth.models import User

class Station(models.Model):
      name = models.CharField(max_length=200)
      latitude = models.DecimalField(max_digits=9, decimal_places=6)
      longitude = models.DecimalField(max_digits=9, decimal_places=6)
      slots = models.IntegerField

class Availability:
      date = models.DateField()
      time = models.TimeField()
      empty = models.BooleanField()
      full = models.BooleanField()
      station = models.ForeignKey(Station, on_delete=models.PROTECT)


class Comment(models.Model):
    commented_to = models.ForeignKey(Station, on_delete=models.PROTECT)
    commented_by = models.ForeignKey(User, on_delete=models.PROTECT)
    content = models.CharField(max_length=200)
    name = models.CharField(max_length=20)
    creation_time = models.DateTimeField()
    def __str__(self):
        return f'Comment(id={self.id}): commented_by={self.posted_by}'
```



5. HTML Mock-Ups (essential views)

5.1 Login / Register (`login.html`)

HTML
```html
<!doctype html>
<html>
<head><title>POGOH Dashboard – Login</title></head>
<body>
  <main id="auth-box">
    <h1>Sign in</h1>
    <form>
```

```
      <label>Email <input type="email" /></label>
      <label>Password <input type="password" /></label>
      <button type="submit">Login</button>
    </form>
    <p><a href="/register">Create account</a></p>
  </main>
</body>
</html>
```

5.2 Main Dashboard (`dashboard.html`)

HTML
```
<!doctype html>
<html>
<head><title>POGOH Dashboard</title></head>
<body>
  <header><h1>Station Status</h1></header>

  <aside id="filter-pane">
    <h2>Stations</h2>
    <input placeholder="Search…" />
    <ul id="station-list"><!-- populated by React --></ul>
  </aside>

  <section id="map-container"><!-- Google Maps canvas
--></section>
</body>
</html>
```

5.3 Station Detail (`station.html`)

HTML
```
<!doctype html>
```

```
<html>
<head><title>Station Detail</title></head>
<body>
  <header><a href="/dashboard">← Back</a></header>

  <section id="kpi-card">
    <h2 id="station-name">12th & Carson</h2>
    <p id="bike-count" class="kpi-number">11 / 16 bikes</p>
  </section>

  <section id="chart-area"><canvas
id="availability-chart"></canvas></section>

  <section id="comments">
    <h3>Comments</h3>
    <form id="comment-form"><textarea
maxlength="500"></textarea><button>Post</button></form>
    <ul id="comment-list"><!-- React renders comments --></ul>
  </section>
</body>
</html>
```