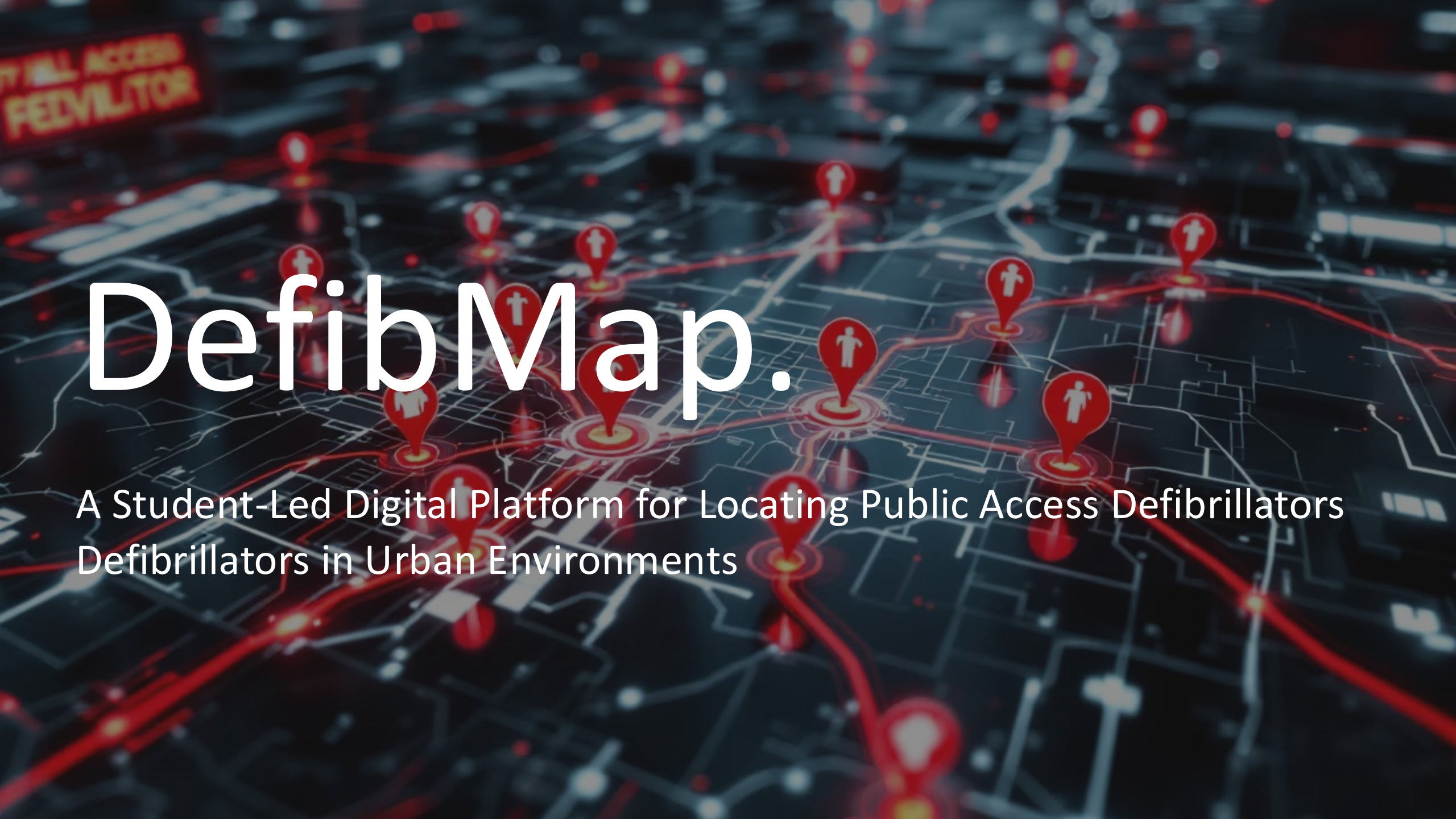




DefibMap.

A Student-Led Digital Platform for Locating Public Access Defibrillators
Defibrillators in Urban Environments

DefibMap London: Bridging the Defibrillator Gap

Out-of-hospital cardiac arrest (OHCA) carries a dire survival rate in the UK, where delayed defibrillation is a critical factor. Despite thousands of deployed AEDs, fragmented and outdated data severely hinder public access during emergencies, creating a critical gap in life-saving efforts.



The Challenge

Time-sensitive intervention for OHCA is crucial. However, the public struggles to locate nearby AEDs quickly due to scattered, scattered, unverified, and non-public data, impacting survival rates.



Our Solution

DefibMap London is a geospatial digital platform designed to integrate, verify, and visualise real-time AED locations, empowering the public with immediate access and guided routing.

AED Access: The Current Landscape

Despite their life-saving potential, public access defibrillators (AEDs) are frequently underutilised due to significant barriers in awareness and awareness and accessibility. This is the challenge DefibMap London aims to address.

The Problem

- Over 30,000 cardiac arrests annually in the UK; survival rates below 10%.
- Early defibrillation can dramatically increase survival to 70%.
- Public awareness of critical AED locations is extremely low.
- Absence of a dedicated, instant UK mobile app for AED identification.

What Exists Today

- defibfinder.uk (The Circuit): Web-only, lacks offline functionality and routing.
- Overseas applications (e.g., Poland) are not compatible with UK with UK data.
- French AED app: Poor user ratings, unstable, reported to crash during emergencies.

(De Rolt et al., 2017; European Resuscitation Council, 2021; Sakib, Alam and Islam, 2024)

Gaps Highlighted by Research

→ Emergency Response Research (De Rolt et al., 2017) Shows:

- Mobile emergency systems rely on real-time location, nearby users, and fast interaction.
- Crowdsensing can enhance emergency support but requires stable mobile tools.

(Competence-Based CRNs)

→ Health Usability Research (De Rolt et al., 2017) Shows:

- Most emergency apps fail due to poor usability, complex UI, or instability.
- Users need fast, simple, reliable, mobile-first experiences.

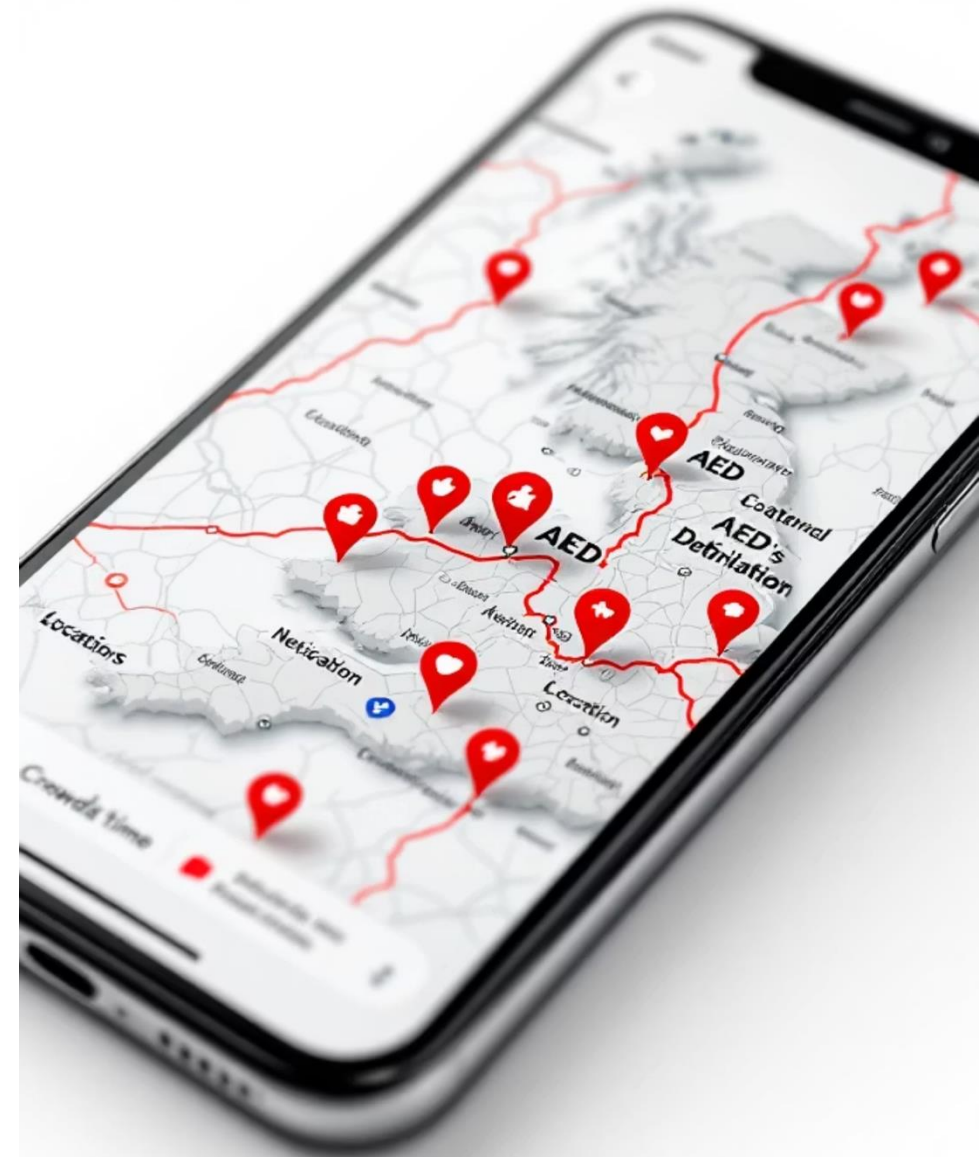
(User-Centered mHealth App Study)



The Opportunity for DefibMap

DefibMap can fill the national gap by providing:

- A dedicated UK AED locator app
- Real-time location & nearest-AED routing
- Offline access for low-signal environments
- Reliable, stable app experience
- Crowdsourced AED submissions + admin verification
- Backed by research on emergency response & mobile usability



Back-End Developer & Data Engineer

– Marius Bobitiu

As the Back-End Developer & Data Engineer, I designed and built the robust infrastructure that underpins DefibMap, ensuring seamless data flow, secure API interactions, and system reliability.

Responsibilities

- Build and maintain resilient server-side logic powering DefibMap.
- Manage AED data ingestion, validation, and secure storage.
- Provide secure and efficient APIs for front-end integration.
- Maintain high performance and reliability of all back-end systems.

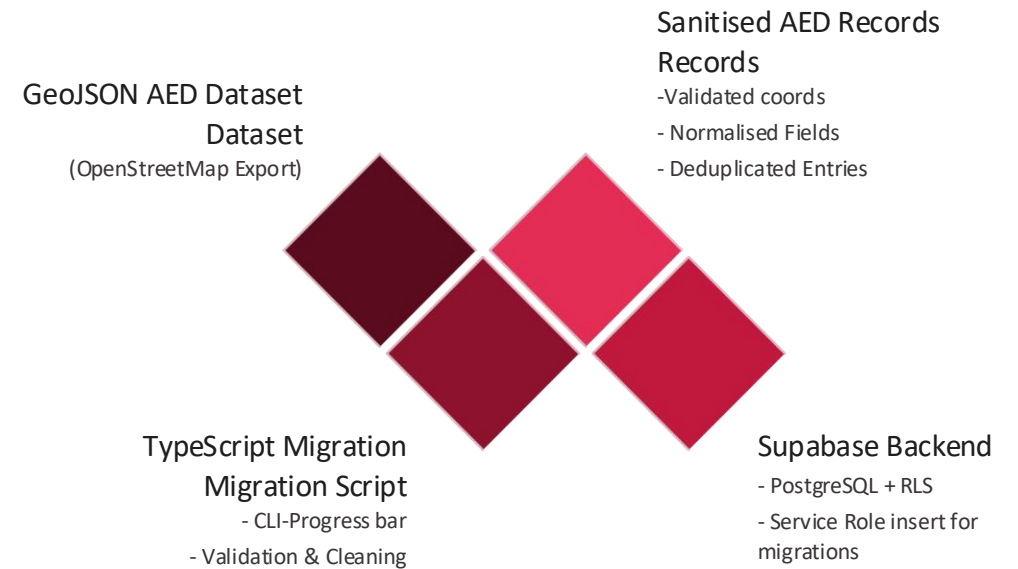
Key Contributions

- Established a scalable and secure data pipeline for critical AED information.
- Developed API endpoints enabling real-time map updates and user interactions.
- Implemented data validation protocols to ensure accuracy and consistency.
- Collaborated with DevSecOps on CI/CD integration and robust security measures.



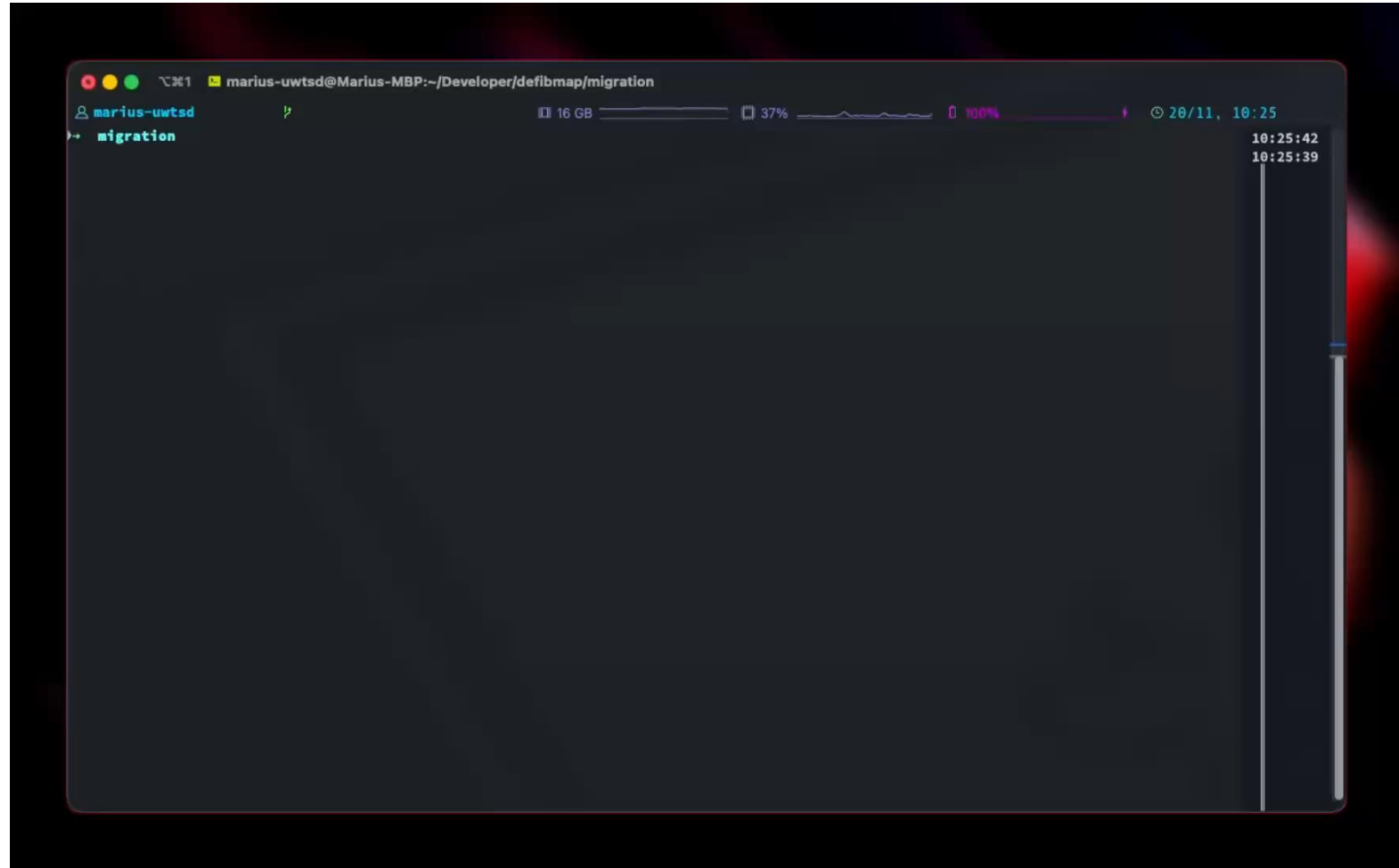
Data Integration Pipeline

- Built a TypeScript migration pipeline to ingest AED GeoJSON data. GeoJSON data.
- Uses `cli-progress` for real-time visual import tracking.
- Validates coordinates, removes duplicates, and normalises
- Securely connects to Supabase using service role keys.
- Ensures repeatable data loads (idempotent).
- Outputs clean, structured AED records for the backend API.
- Designed to support future dataset sources (OSM, BHF, council data).



Data Migration Script

- Custom TypeScript migration script sits at the core of our data integration workflow
- Ingests raw AED GeoJSON data into the pipeline
- Performs essential validation, normalisation, and deduplication
- Ensures only accurate and consistent AED entries progress to Supabase
- Uses cli-progress for clear, real-time visual feedback during ingestion
- Helps developers instantly spot issues or anomalies
- Guarantees that only high-quality, verified AED records populate the DefibMap database



Above is an example of the migration script in action, showcasing its real-time progress tracking and detailed logging.

Backend API (Go Microservice)

Our backend is a high-performance, containerised Go microservice designed for maximum flexibility, scalability, and security, ensuring a robust foundation for DefibMap London.

Independent & Migratable

A fully independent Go microservice, designed to allow seamless migration to any cloud provider in the future.

Structured REST API

Features a structured REST API with robust validation, validation, rate limiting, and comprehensive error handling for reliable operation.

Data Integration

Integrates securely with Supabase for data storage but is storage but is not directly dependent on its API, maintaining architectural independence.

Admin Workflow Support

Includes endpoints for admin dashboard integration, streamlining internal staff workflows for AED management.

Containerised Deployment

Deployed as a containerised service using Docker and managed via GitHub Actions for continuous integration and delivery.

Core Endpoints

- `GET /defibrillators` — Retrieves a list of verified AEDs. Accepts pagination, and boundaries to filter out number of results for performance.
- `GET /defibrillators/nearest` — Efficiently finds the nearest AEDs based on location (Go implementation).
- `DELETE /defibrillators/{id}` — Deletes an AED record (to be confirmed for production use).
- `POST /submissions` — Allows users to submit new AED locations (initially sent to a pending queue). Allow for paginated results.
- `GET/submissions` — Retrieves a list of all submissions made by users.
- `GET/submissions/{id}` — Retrieve a submission by ID.
- `PATCH /submissions/{id}` — Approve / Deny submission.
- `DELETE /submissions/{id}` — Deletes a requested submission.
- `GET/users` — Retrieves a list of all users created for the admin-panel.

Live API Requests Logs

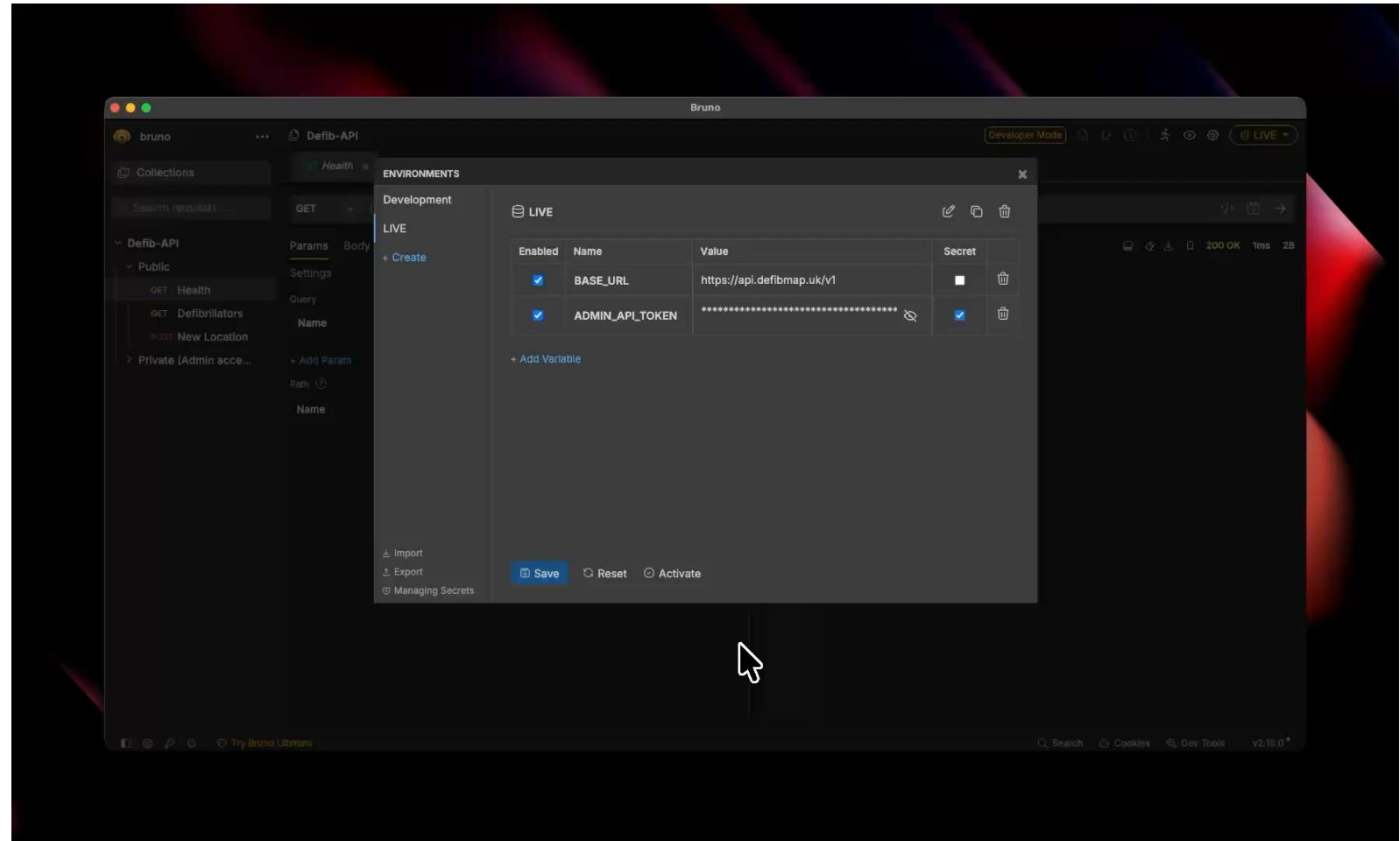
- **Real-time logs** showing continuous, live traffic interacting with the DefibMap API
- Requests include both **GET** (data retrieval) and **POST** (data submission) operations from different clients and devices
- Consistent **200/201** status codes across all endpoints, indicating stable, error-free behaviour
- Fast execution times — often under **0.1** seconds — demonstrating efficient query handling and strong backend performance
- Logs include **geospatial** filters, **pagination**, and **device registration**, showcasing the variety of real-world usage
- **IP** and **timestamp** tracking provide visibility into client activity and system responsiveness

(Bobitiu, 2025)

info	GET	api.defibmap.uk	/v1/defibrillators?limit=1	Status: 200	Exec Time: 1.214149509s	IP: 172.70.160.215:11140	2025-12-10 10:38:52
info	GET	api.defibmap.uk	/v1/submissions?limit=50&offset=0	Status: 200	Exec Time: 0.501766557s	IP: 172.69.194.7:13035	2025-12-10 10:38:52
info	GET	api.defibmap.uk	/v1/defibrillators?min_lat=51.50203630358239&max_lat=51.55918237422514&min_lng=0.7...	Status: 200	Exec Time: 0.056016895s	IP: 162.158.216.145:12583	2025-12-08 20:42:45
info	GET	api.defibmap.uk	/v1/defibrillators?min_lat=51.51454205710543&max_lat=51.56454205710543&min_lng=0.7...	Status: 200	Exec Time: 0.410167753s	IP: 162.158.216.145:12583	2025-12-08 20:42:41
info	POST	api.defibmap.uk	/v1/devices/register	Status: 201	Exec Time: 0.041635873s	IP: 162.158.216.145:12583	2025-12-08 20:42:41
info	GET	api.defibmap.uk	/v1/defibrillators?min_lat=51.50368289749645&max_lat=51.560826902503536&min_lng=0....	Status: 200	Exec Time: 0.073737522s	IP: 141.101.99.27:12277	2025-12-08 20:15:01
info	GET	api.defibmap.uk	/v1/defibrillators?min_lat=51.464966119904304&max_lat=51.77047928486766&min_lng=0....	Status: 200	Exec Time: 0.090334251s	IP: 141.101.99.27:12277	2025-12-08 20:14:57
info	GET	api.defibmap.uk	/v1/defibrillators?min_lat=51.46030955909188&max_lat=51.76585396502858&min_lng=0.5...	Status: 200	Exec Time: 0.090070667s	IP: 141.101.99.27:12277	2025-12-08 20:14:56
info	GET	api.defibmap.uk	/v1/defibrillators?min_lat=51.44659581515542&max_lat=51.752232215307615&min_lng=0....	Status: 200	Exec Time: 0.148973867s	IP: 141.101.99.27:12277	2025-12-08 20:14:53

API in Action: Live Request Testing with Bruno

- Demonstration of real API calls using the Bruno REST client
- Shows how endpoints respond to different parameters and filters
- Validates correct JSON outputs, status codes, and error handling
- Highlights speed and consistency across repeated requests
- Demonstrates secure access using API keys and environment variables
- Provides a transparent view of how developers interact with the backend



Security by Design

Our commitment to data integrity and user safety is paramount. We implement a multi-layered approach to security, integrating robust protocols and best practices across all components of DefibMap London.

- API Key-protected endpoints for public API usage.
- Role-based access (User / Staff / Service) ensures controlled data interaction.
- Supabase Row-Level Security for granular database protection.
- Rigorous sanitization of coordinates & metadata prevents data corruption.
- Mandatory HTTPS and secure JWT/API key handling for all communications.
- Comprehensive logging & audit trails (Supabase + Go logs) for accountability.
- Intelligent rate limiting to prevent spam and abuse on our services.



Database Design (Overview)

- **Defibrillators Table**

Stores all verified AED records with full metadata and precise PostGIS geolocation.

- **Submissions Table**

Holds community-submitted AED reports before approval, including coordinates, notes, and status.

- **Auth & Security (BetterAuth)**

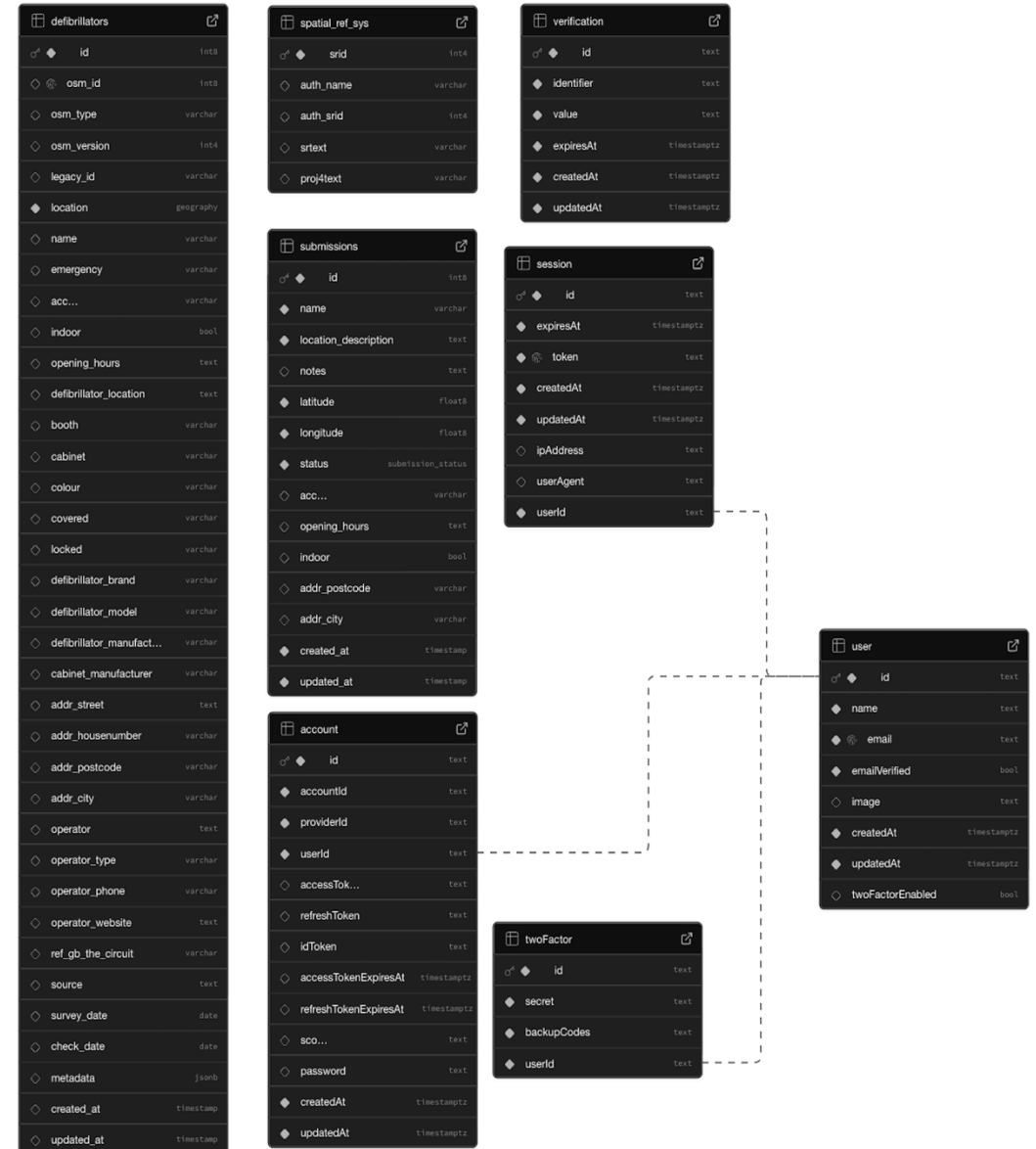
- user: core user profiles
- account: linked provider accounts
- session: active login sessions
- verification: email/token verification
- twoFactor: optional 2FA secrets + backup codes

- **Spatial Reference (spatial_ref_sys)**

Standard PostGIS reference table for accurate geospatial handling.

- **Design Logic**

Clear separation between verified AEDs and user submissions, supported by a secure and modern authentication layer.



Why Go is the Best Choice for This API



Lightning Fast

Go is compiled, not interpreted, delivering exceptional speed and efficiency for critical operations.



Geospatial Power

Optimised for handling complex geospatial and map-based data, crucial for DefibMap's core functionality.



Cost-Effective Hosting

Its minimal memory footprint translates to lower hosting costs, making our infrastructure more economical.



High Concurrency

Easily manages thousands of concurrent requests per second, ensuring ensuring responsiveness under heavy load.



Predictable Performance

Offers consistent, low-latency performance, vital for emergency services and real-time data access.



Robust Ecosystem

Benefits from a strong developer community and mature frameworks like Chi, Fiber, and Echo.



Easy Containerisation

Simplifies deployment with tiny Docker images, enhancing portability and CI/CD pipelines.



Industry Standard

Trusted by major infrastructure companies such as Cloudflare, Uber, Docker, and Supabase.

Backend Roadmap

Foundation & Core Architecture

Finalize the AED data model, establish the project structure, and implement core endpoints to support initial data retrieval submissions.

Geospatial & Real-Time Features

Introduce Go-powered geospatial search, real-time geolocation support, and automated sync with OpenStreetMap for accurate and up-to-date AED coverage.

Observability & Offline Access

Deploy monitoring dashboards for system performance and enable offline AED data export for mobile users during emergencies.

Integration & Deployment

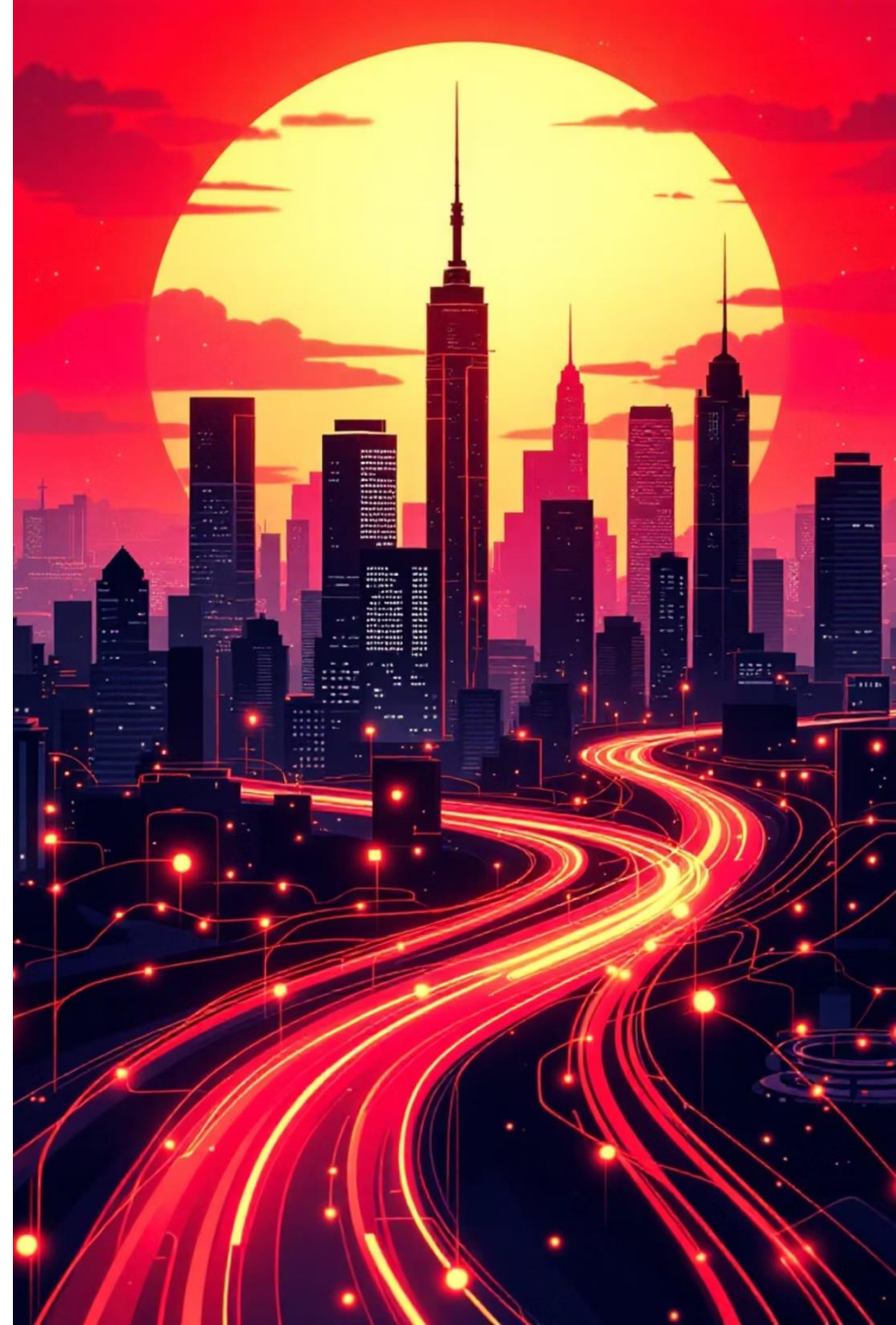
Connect the Go microservice API with the mobile app, integrate API keys and role-based access, and set up CI/CD with Docker and GitHub Actions.

Data Quality & Partner Expansion

Build ingestion pipelines for partner datasets, add automated data validation and deduplication jobs, and improve overall data reliability.

Admin & Verification Tools

Develop streamlined admin workflows for reviewing, verifying, and managing user-submitted AED locations.



Project Reflection

A summary of key insights and lessons learned from the development of DefibMap London.



The development process underscored the critical need for establishing a clear backend architecture early, ensuring both scalability and consistency and consistent data handling throughout the project lifecycle.



Work on geospatial search and submission workflows highlighted the extensive requirements for robust data validation, effective deduplication strategies, and consistent structured metadata across diverse data sources.



Real-time testing demonstrated solid performance metrics, while simultaneously revealing significant opportunities to enhance system observability and implement more sophisticated automated monitoring solutions.



Integrating the backend with the mobile application profoundly reinforced the value of well-defined interfaces and comprehensive documentation for facilitating smooth, efficient cross-platform development.



Collectively, the project significantly deepened our understanding of core backend engineering principles, with a particular focus on reliability, data governance, and long-term system maintainability.

Additional Project Contributions

Expanding the DefibMap ecosystem with essential tools and user-facing applications beyond the core backend services.

Admin Panel (Next.js + Supabase)

- Full-featured dashboard for AED management.
- Supports review, approval, and verification of community submissions.
- Includes user management, moderation, and secure workflows.
- Integrated API for real-time data synchronisation.
- Uses BetterAuth for secure role-based access.

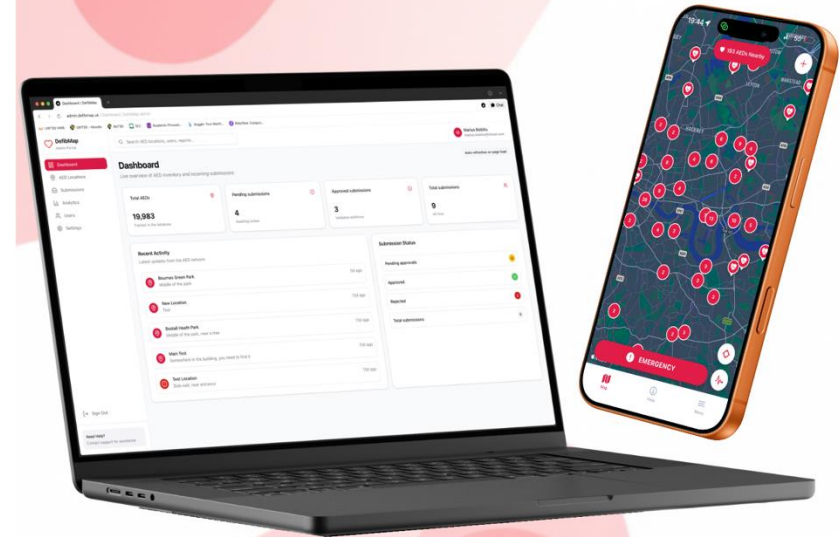
iOS Mobile Application (React Native + Expo)

- Cross-platform app for real-time AED discovery.
- Features live location tracking and nearest-AED routing.
- Includes essential offline support.
- Integrates with the Go API for fast, low-latency data.
- Clean UI following mHealth usability principles.
- Tested on-device with native iOS build pipeline.

APP Available on **TestFlight** (currently internal testing).

Admin available at: <https://admin.defibmap.uk>

API (v1) online: <https://api.defibmap.uk/v1>



References

Academic Papers:

- **Aditya, W. (2018)** 'Evaluating Concurrency Patterns in Go for High-Performance Network Services', Journal of Computer Science, 14(5), pp. 623–634.
- **De Rolt, C.R., Foschini, L., Gauthier, F., Hasse, D. and Montanari, R. (2017)** 'Competence-Based Mobile Community Response Networks', IEEE Symposium on Computers and Communications, pp. 1–7.
- **Ganti, R.K., Ye, F. and Lei, H. (2011)** 'Mobile Crowdsensing: Current State and Future Challenges', IEEE Communications Magazine, 49(11), pp. 32–39.
- **Obe, R.O. and Hsu, L.S. (2021)** PostGIS in Action. 3rd edn. Manning Publications.
- **Sakib, A.M., Alam, S.A. and Islam, A. (2024)** 'User-Centered Design and Validation of mHealth App for Providing Vital Assistance and Emergency Healthcare Support in Bangladesh', International Congress on HORA, pp. 1–6.
- **Schrepp, M., Kollmorgen, J. and Thomaschewski, J. (2023)** 'A comparison of SUS, UMUX-LITE, and UEQ-S', Journal of User Experience, 18(2), pp. 1–12.
- **Symeonakis, O. and Kourtesis, D. (2020)** 'Evaluating Go Microservices for Scalable Cloud-Native Applications', International Journal of Cloud Computing, 9(4), pp. 312–330.
- **European Resuscitation Council (2021)** Guidelines for Resuscitation 2021. Available at: <https://erc.edu/> (Accessed: 05 Nov 2025)

Technical Documentation

- **PostgreSQL Global Development Group (2024)** PostgreSQL 16 Documentation. Available at: <https://www.postgresql.org/docs/> (Accessed: 12 Nov 2025).
- **PostGIS Project (2024)** PostGIS 3.4 Manual. Available at: <https://postgis.net/documentation/> (Accessed: 12 Nov 2025).

- **Supabase (2024)** Supabase Documentation. Available at: <https://supabase.com/docs> (Accessed: 19 Nov 2025)
- **The Go Authors (2024)** The Go Programming Language Specification. Available at: <https://go.dev/ref/spec> (Accessed: 26 Nov 2025).
- **Docker Inc. (2024)** Docker Documentation. Available at: <https://docs.docker.com> (Accessed: 26 Nov 2025).

Project Artefacts:

- **Bobitiu, M. (2025)** *DefibMap London* – Backend API (Go Microservice). Internal project artefact.
- **Bobitiu, M. (2025)** *DefibMap London* – TypeScript Data Migration Pipeline. Internal project artefact.
- **Bobitiu, M. (2025)** *DefibMap London* – iOS Mobile Application Build (Expo/React Native). Internal project artefact.
- **Bobitiu, M. (2025)** *DefibMap London* – Admin Panel Interface (Next.js). Internal project artefact.