ADR-8

Adoption of Docker for Local Development and Deployment

Date: 12-04-2025

Status

Accepted

Author

System Architect

Context

The project comprises multiple independently deployable microservices (user, facility, booking, payment), along with a frontend and a shared database. Running and managing all these components individually during development or testing introduces complexity and slows down the feedback loop. Ensuring consistent environments across developer machines without a standardized approach would create unnecessary overhead.

Decision

Docker will be used to containerize all services, the frontend, and the database. Docker Compose will be employed to orchestrate them during development and testing phases. This approach enables seamless local deployment and environment consistency.

Alternatives

Alternative	Pros	Cons
Manual local setup per service	Familiar and flexible	Tedious, error-prone, hard to scale, inconsistent environments
Use Docker (Chosen)	Lightweight, consistent environments, fast setup,	Requires Docker knowledge, managing Compose files and

	scalable, easy orchestration	volumes

Rationale

Docker provides lightweight containerization, fast startup, and environment parity. It simplifies onboarding, ensures consistent environments across team members, and enables dependency isolation. With Docker Compose, services can be spun up with a single command, easing the development and integration workflow.

Consequences

Positive:

- Simplified setup for developers
- Consistent environments
- Easy service orchestration and teardown
- Isolation from host system dependencies

Negative:

- Slight overhead in managing volumes, networks, and Dockerfiles