

Computing and Information Sciences

Fall 2019 Senior Design Project

Work Away Ticket Management Web App

Students: Osniel Valdivia
Mentor: **Michael Reyeros**, Senior Developer at Assurant
Instructor/Faculty: **Dr. Masoud Sadjadi**, Florida International University



Problem

Companies often have a tremendous amount of support tickets. Users are often frustrated with call hold times. Support tickets are often inefficiently queued and the workload is unevenly distributed. Work Away is here as the cloud-based solution to the modern ticketing workflow problem. Using REST API and Docker, Work Away can scale its backend to support most companies' use-case. Work away is open source and can be deployed privately on premise. Or if you prefer, you can use our cloud platform which is accessible through the internet.

Current System

Work Away consists of a MongoDB Atlas data store, a NodeJS Express REST backend, Docker containerization, Chai, Mocha, and Supertest for automated testing of REST endpoints, and Open API doc spec with a hosted Swagger UI.

My Contribution

- Node Express HTTP Server
- Mongodb Atlas deployment
- User and Company api routes
- User and Company mongo schema
- User and Company test suite
- Auth middleware with jwt and bcryptjs
- OpenAPI yaml definition
- SwaggerUI api route
- Dockerization
- Deployment

Requirements

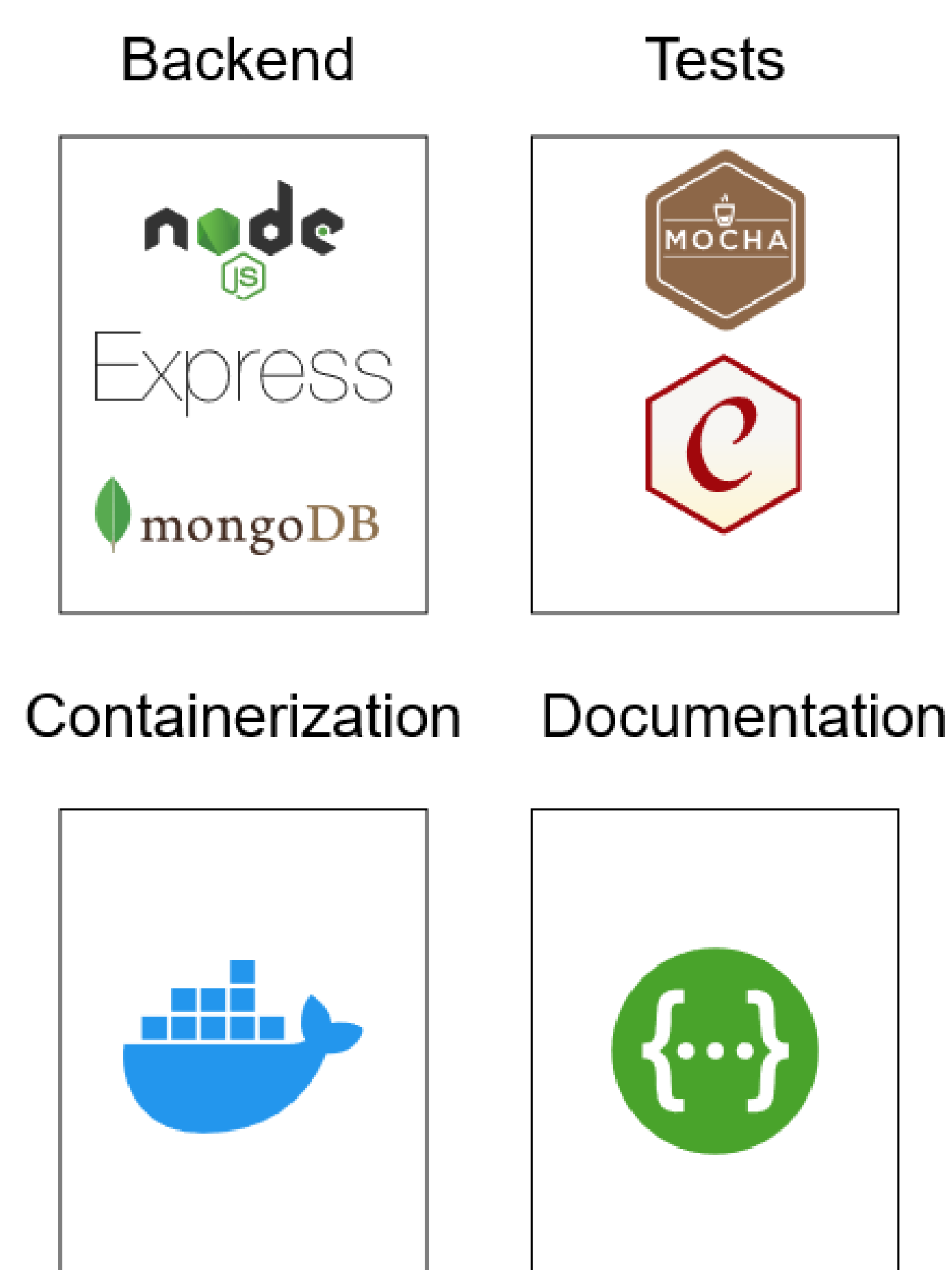
Hardware:

- Any computer hardware that runs Windows or MacOS or Linux
- Cloud-based provider to deploy on the internet

Software:

- Git
- NodeJS
- Docker
- CMD or Terminal or bash
- Code editor (VS Code)
- Mongodb Atlas instance

System Design



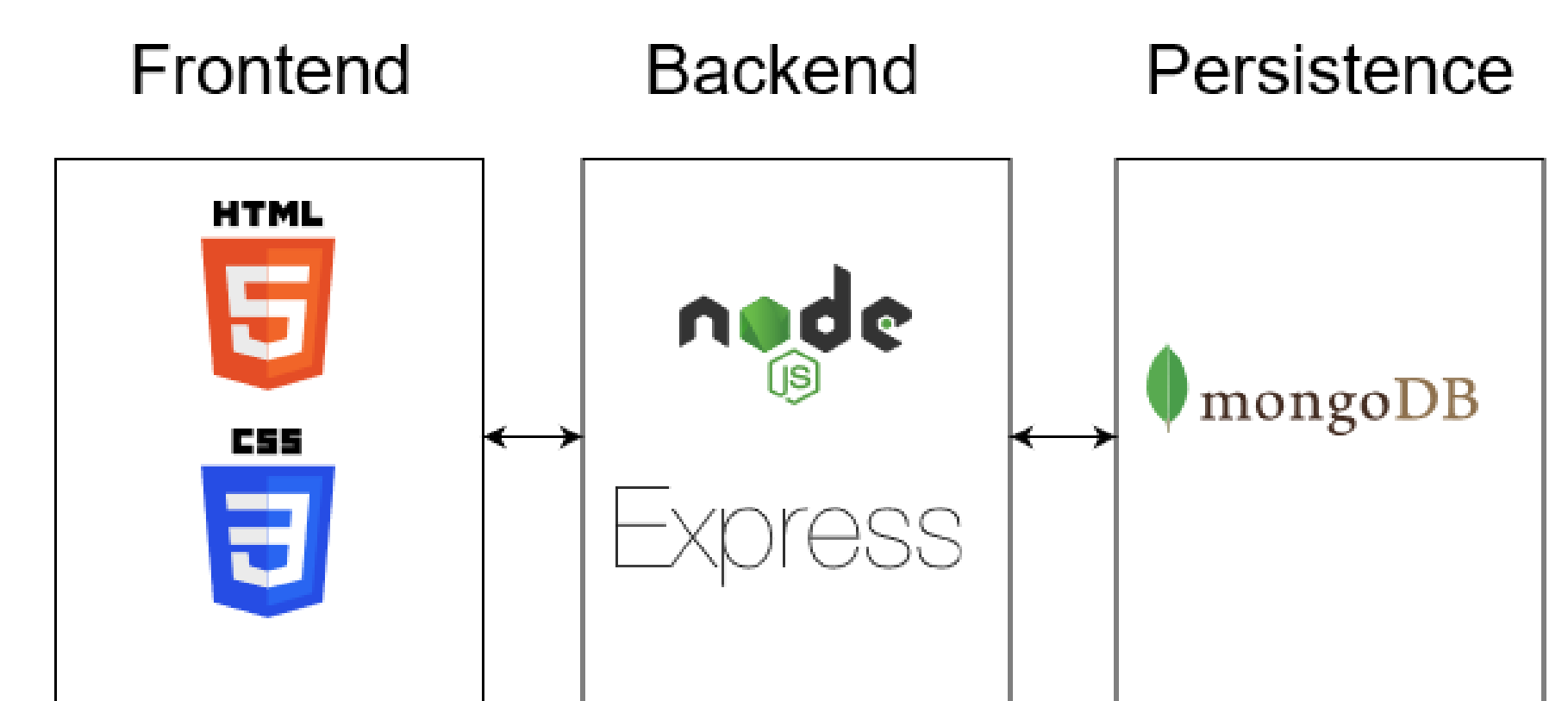
Implementation

The Work Away app was built using VS Code, Javascript, Node, and cmd. File formats such as js, yaml, json, and Dockerfile are present in the project. Dependencies can be installed using npm. Project instructions can be found in the Readme.md file

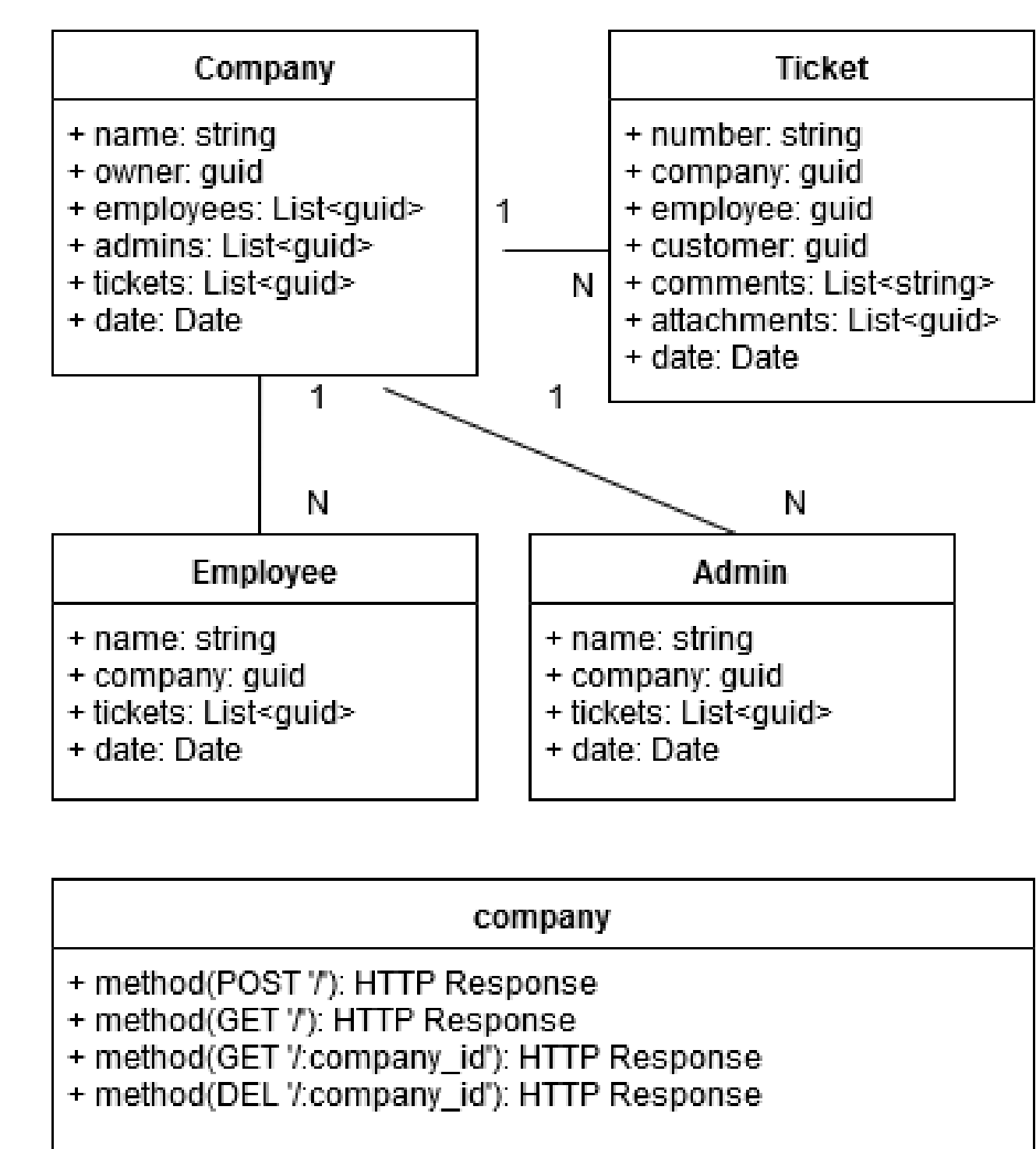
Components

- The Work Away API was built using NodeJS, Express, and MongoDB Atlas.
- The Work Away test suite was built using chai, mocha, and supertest.
- The Work Away API docs was built using OpenAPI spec and SwaggerUI.
- The Work Away API was dockerized with Docker and deployed with Azure

Architectural Diagram



Class Diagram



Verification

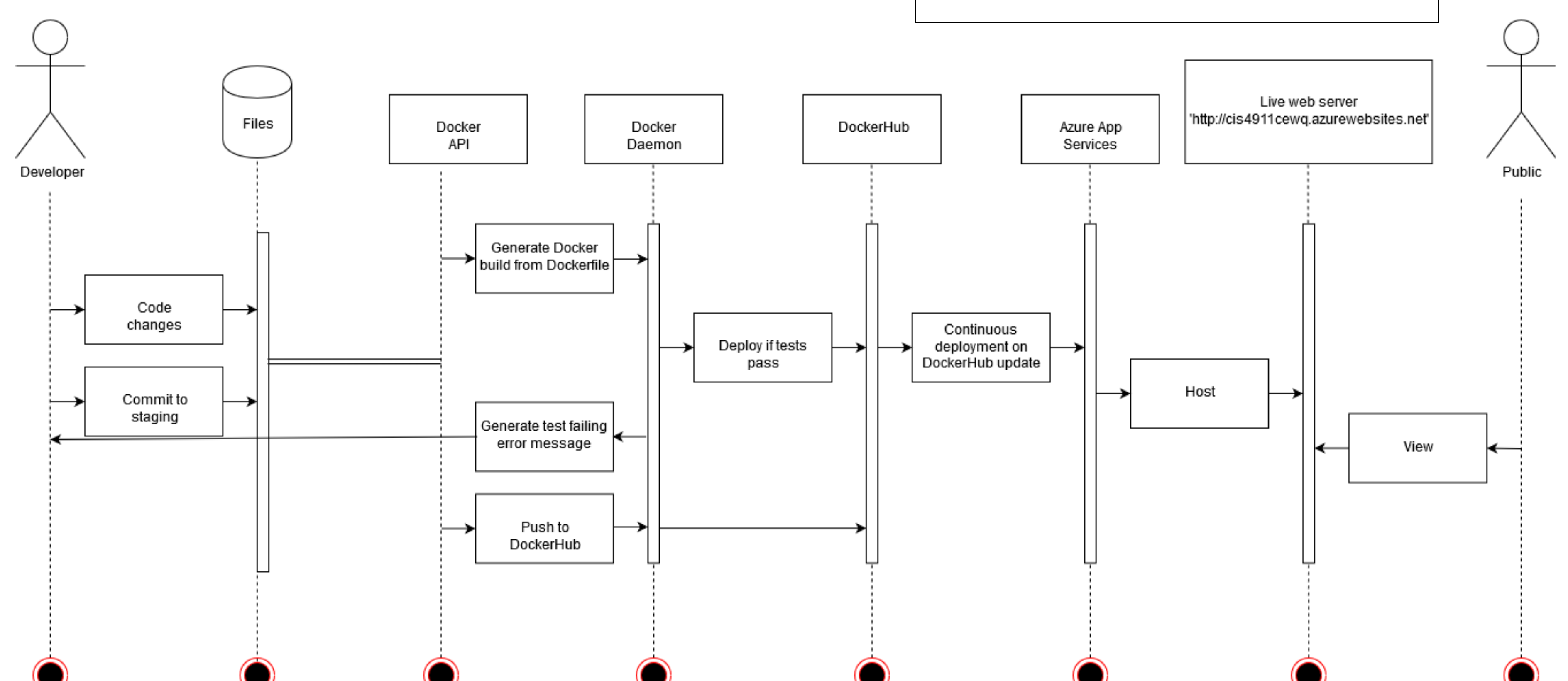
Before Work Away can be built and deployed it must pass all of the automatic tests. Docker has continuous integration built in when specified in the Dockerfile, building only if the "npm run test" command passes.

Test Suite

- Clear test database
- Generate fake data via faker npm library
- Setup database
- Setup Users
- Register Owner
- Change user type to owner
- Create Company
- Test company restrictions
- Get Company
- Teardown suite

Summary

Deployment Sequence Diagram



Acknowledgement

Special thanks to **Dr. Masoud Sadjadi** and **Michael Reyeros** for the opportunity to work on this project.