

Food Oasis

Donald Weaver
Lord Wiafe
Stuart Idehen
Marcus Anderson

Goal

- Create a network of fresh food sources
- Connect users with local food sources
- Make the connection as simple and easy as possible
- Android App



Purpose

- Help encourage healthy eating
- Increase food availability
- Promote sustainable food sources
- Support local farms and businesses
- Decrease food waste



Mission: Promote Healthy Eating

- 42% of American adults are obese
- Increase visibility of healthy food options
- LOCAL FRESH FOOD TASTES SO MUCH BETTER



Mission: Increase food availability

- In 2021 8.4 million U.S. households experienced food insecurity
- Covid stinks
- There are tons of people giving away food
- 30-40% of the U.S. food supply is wasted each year



Can we achieve
both?

Functional Requirements

- The system needs to show users a list of nearby food sources
- A user should be able to select a food source to view a source profile
- The system should be able to give directions to a food source when a user requests it
- A food supplier should be able to create a food source profile
- A food supplier should be able to view previously created food source profiles in order to retroactively edit them



Non-functional requirements

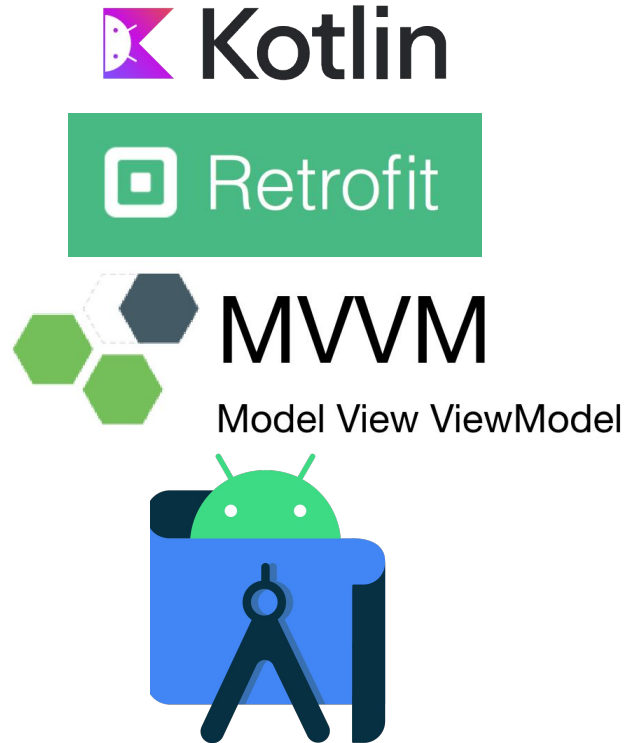
- Loosely coupled system.
- Strongly use facade pattern to enhance modularity.
- Logging system for keeping track of operational issues.
- Server-side must be accessible 24/7 by Android app.*
- System should utilize caching of requests to speed up response times.



Implementation

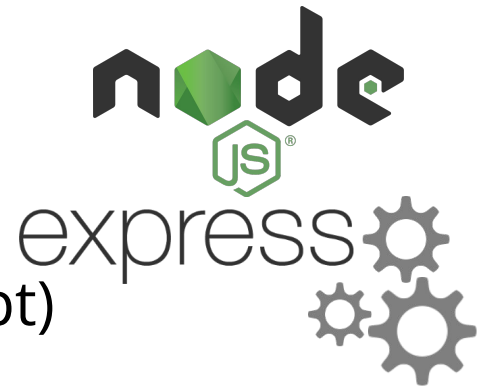
Front-end

- Programming Language: Kotlin
- API Handler: Retrofit
- Architecture: MVVM (Model View ViewModel)
- User Interface: XML
- Development Environment: Android Studio



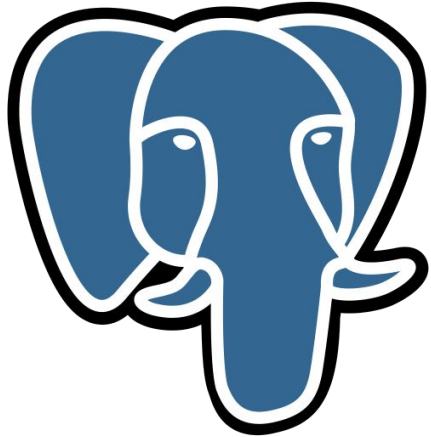
Back-end

- Programming Language: NodeJS (Typescript)
- Framework: Express.js
- API Handler: Routing Controllers
- Dependency Injection: TypeDI
- Authentication: JWT
- System Logging: Winston
- API Documentation: Swagger
- Development Environment: VS Code



Database

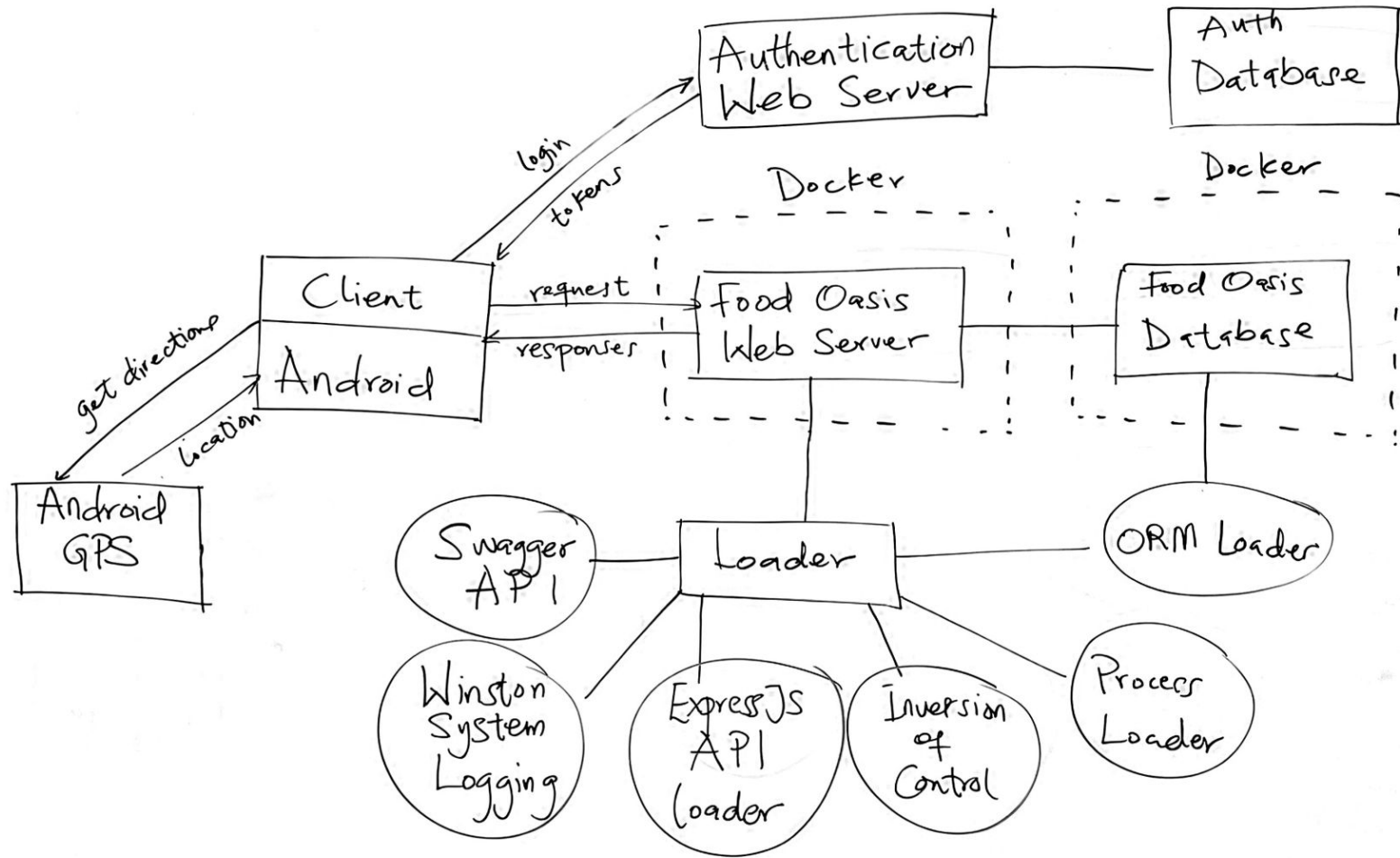
- Database: PostgreSQL
- Development Environment: PgAdmin



Project Management

- Designs: Miro





Haversine Formula

Algorithm for proximity detection between two GPS locations

$$\begin{aligned}d &= 2r \arcsin\left(\sqrt{\text{hav}(\varphi_2 - \varphi_1) + (1 - \text{hav}(\varphi_1 - \varphi_2) - \text{hav}(\varphi_1 + \varphi_2)) \cdot \text{hav}(\lambda_2 - \lambda_1)}\right) \\&= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \left(1 - \sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) - \sin^2\left(\frac{\varphi_2 + \varphi_1}{2}\right)\right) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \\&= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right).\end{aligned}$$

Demo

Conclusions

Experience

- Got a grasp on android app development
- Expanded our knowledge of front end development
- Learned basics of Kotlin
- Used our experience in databases
- Strengthened our ability to work in a team

Results

Success

- Achieved goal of connecting users with food sources
- Created an android app
- Users can find food sources
- Easy and simple to use
- Achieved mission: promote healthy eating

Failure

- Partially failed mission: increase food availability
- Users can distinguish donors from sellers
- Suppliers can't create shops
- App is not hosted

fin.