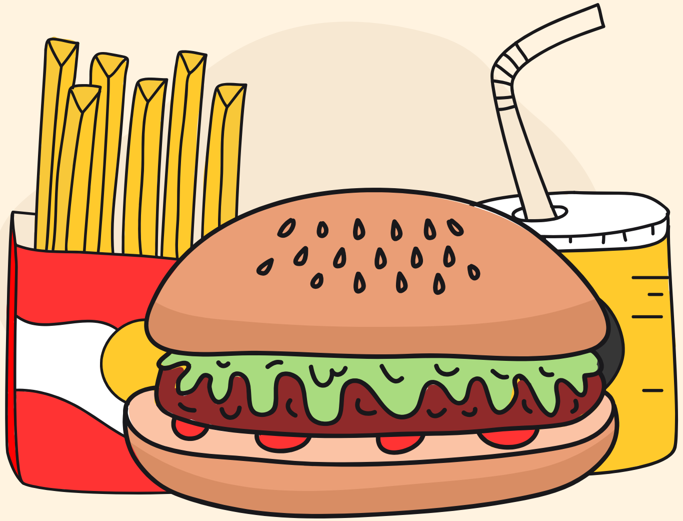


Quick Eats

Kalpit Mody (Scrum Master)
Jack St. Hilaire (Developer)
Leah Harper (Developer)

April 19, 2022

Project Recap

Quick Eats Recap

- Answering the demand for quick drive thru service
- Combating long wait time at fast food drive thrus
- Allows for shorter waits at fast food restaurant drive thrus
    - Inform users of live wait times
- Key Features:
    - Account creation
    - Searching and sorting
    - Viewing on map and lists
    - Favoriting

System Analysis

System Overview

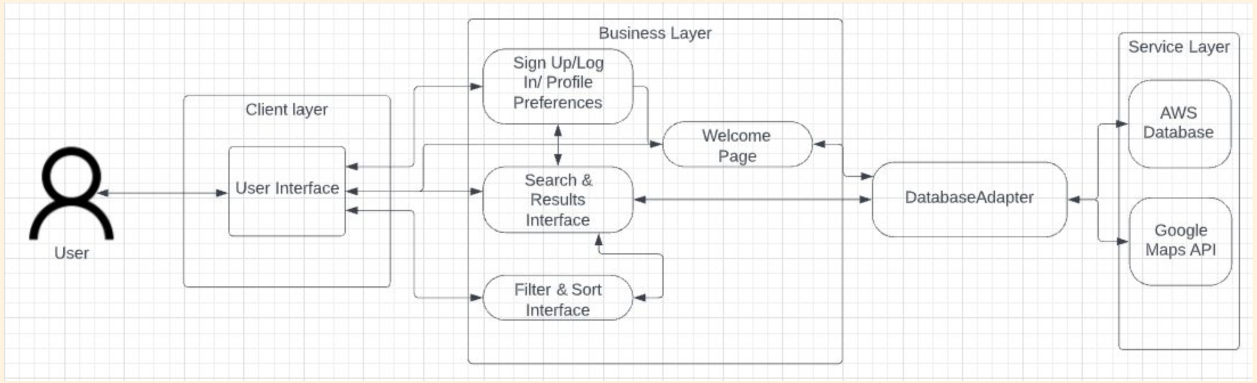
- **Client Layer**
  - iOS and Android clients to display relevant features in user interface

- **Business**
  - Where the magic happens
  - Take in user input such as location and query existing APIs
  - Main user preferences such as favorite restaurants

- **Service**
  - Host and maintain databases
  - API queries

System Diagram

- The user interface talks with
  - Profile preferences
  - The search & results interface
  - The filter & sorting interfaces
  - The welcome page
- The welcome page and search & results interacts with the DatabaseAdapater to access the AWS database and Google Maps API



Actor Identification

- **New Users**
  - Have never created an account before
- **Registered Users**
  - Have created an account before
- **Server**
  - AWS to host database
  - Holds user data

Design Rationale

Architectural Style

We will utilize a **three-tier architecture** in order to structure the project in an effective and efficient way.

- **Client Layer**
  - User interface for iOS and Android, allowing users to interact with our data easily without worrying about what is going on behind the scenes
- **Business Logic Layer**
  - Still using the React framework, this layer will be where we determine what information the user needs, calculate wait times based on user selections, and collect/analyze user submitted data
- **Service Layer**
  - This layer will deal with our databases and provide both security and authentication, utilizing AWS and relational databases

Design Patterns

- **Adapter**
  - Multiple different APIs, along with many of our own components and services
  - Adapter design pattern will make previously incompatible components compatible
  - Allows us to create a seamless three-tier system

- **Facade**
  - Lots of complexity and back-end processes that will be operating within our Business Logic layer
  - Users do not need to be aware of most of this
  - Application will be more usable and efficient if the UI is simple and straightforward

- **Strategy**
  - Lots of analysis, prediction, and data manipulation within the back-end of the application
  - Very important to be able to utilize various algorithms in an interchangeable way

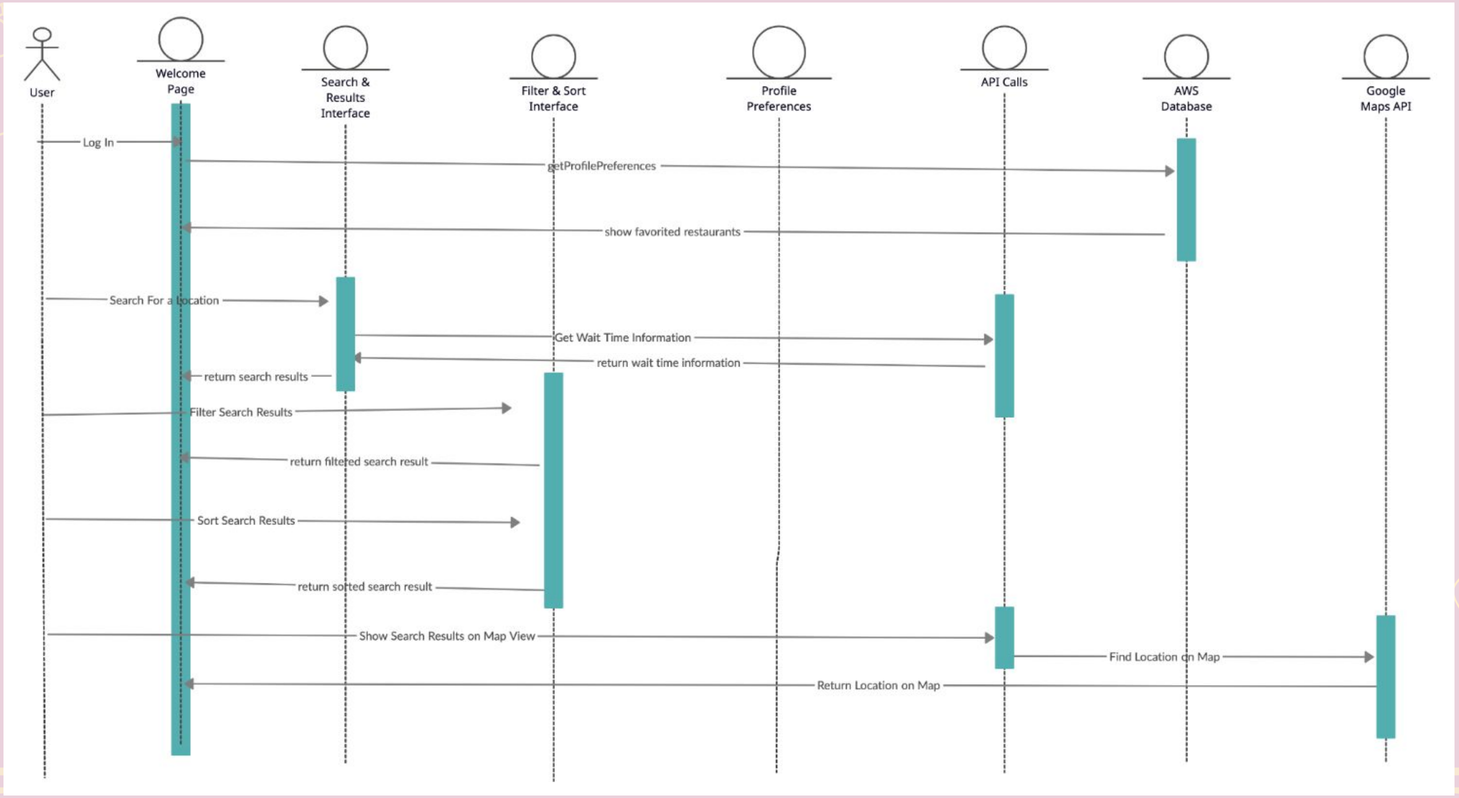
Framework

- **The framework selected was React**
  - Well known and well documented
  - Provides an immense level of flexibility

- **Smooth cross-platform functionality (iOS and Android) is essential**
  - React provides cross-platform design capabilities and documentation
  - Allows for interaction with APIs and services such as AWS, imperative to functionality

Functional Design

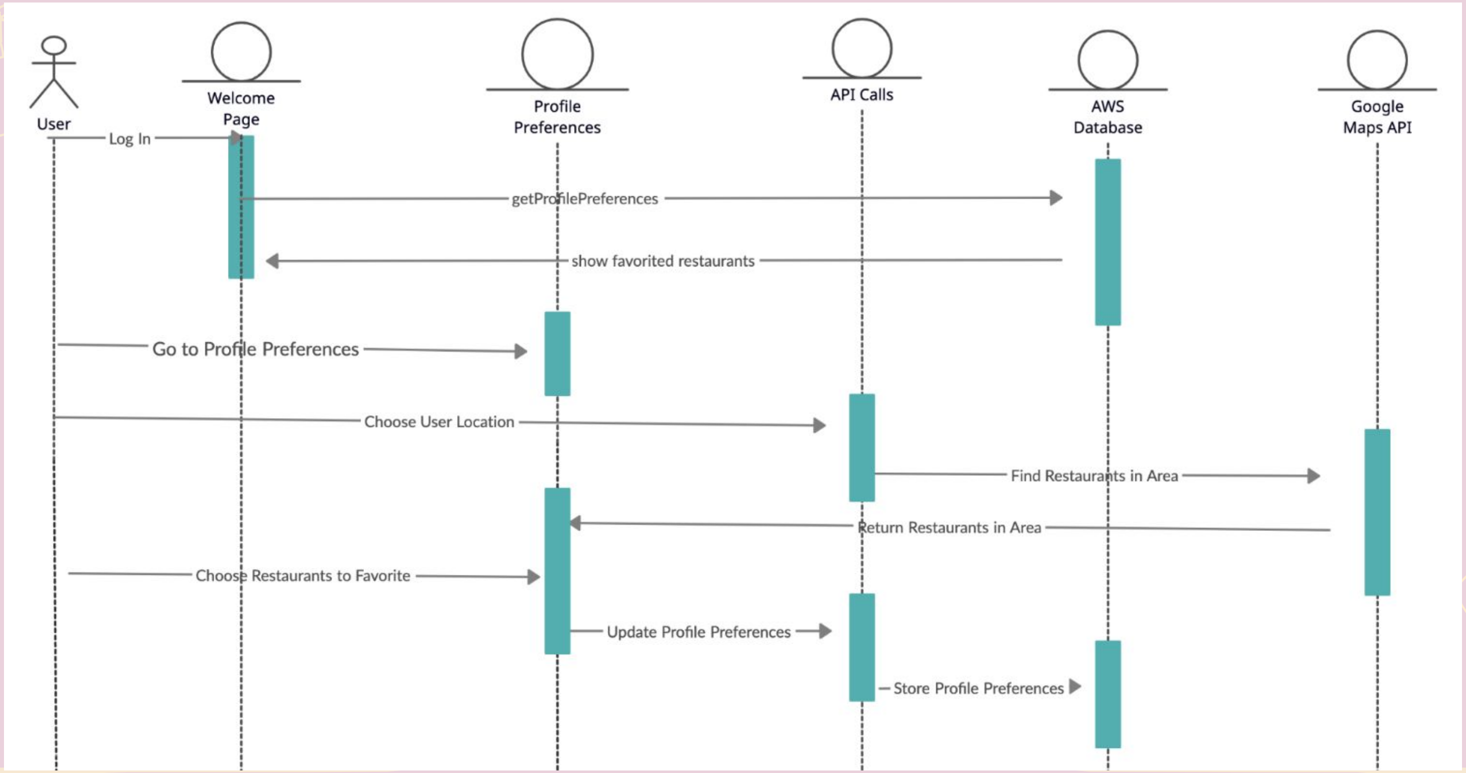
Finding Drive Through Wait Times



Finding Drive Through Wait Times

- Users start at the Welcome Page
- From there, users can:
    - view their favorite restaurants
    - search based on location
    - view/edit their profile
- When a user searches based on location they receive results in list or map form
- Wait times are then displayed for the various restaurants in their location

Updating User Preferences



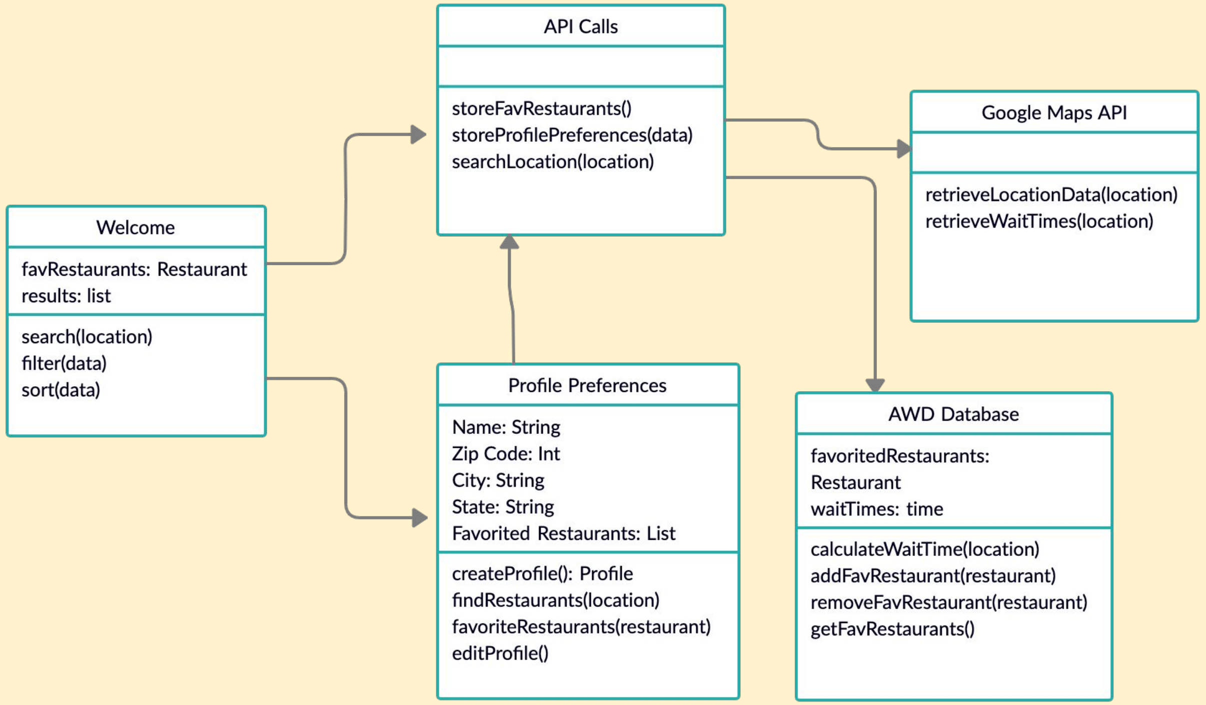
User
Welcome Page
Profile Preferences
API Calls
AWS Database
Google Maps API
Log In
getProfilePreferences
show favorited restaurants
Go to Profile Preferences
Choose User Location
Find Restaurants in Area
Return Restaurants in Area
Choose Restaurants to Favorite
Update Profile Preferences
Store Profile Preferences

Updating User Preferences

- Users start at the Welcome Page
- Users can navigate to the Profile Page
- From the Profile Page users can update/modify:
    - Profile information (name, location, etc)
    - Favorite restaurants

Structural Design

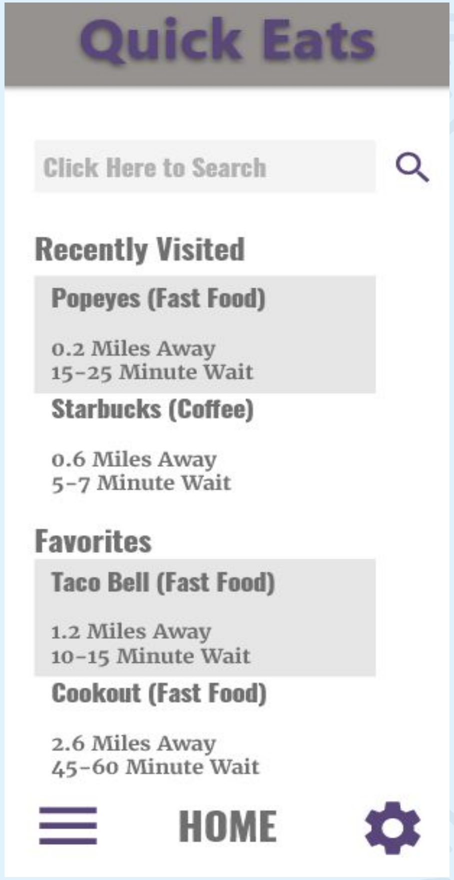
Class Diagram



Project Mockup

Home Page

- Search bar allowing the user to find restaurants based on location
- Recently visited section
- Favorites section
- Navigation bar, including app settings


Quick Eats

Click Here to Search

Recently Visited

Popeyes (Fast Food)
0.2 Miles Away
15-25 Minute Wait

Starbucks (Coffee)
0.6 Miles Away
5-7 Minute Wait

Favorites

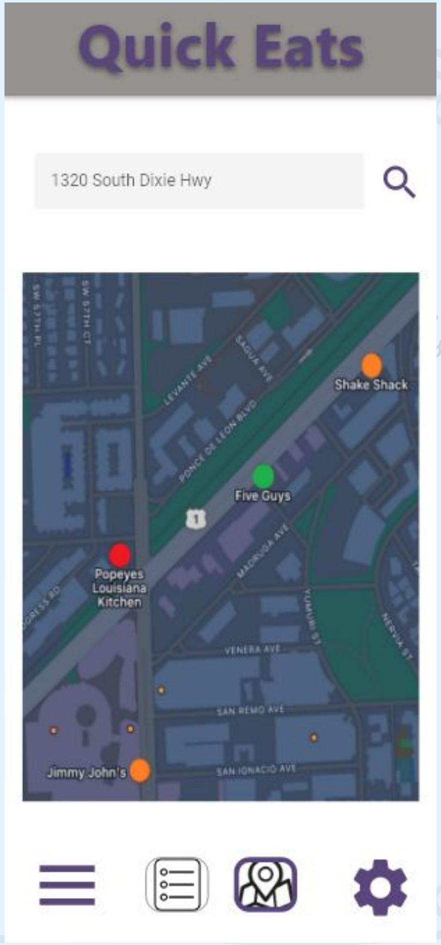
Taco Bell (Fast Food)
1.2 Miles Away
10-15 Minute Wait

Cookout (Fast Food)
2.6 Miles Away
45-60 Minute Wait

HOME

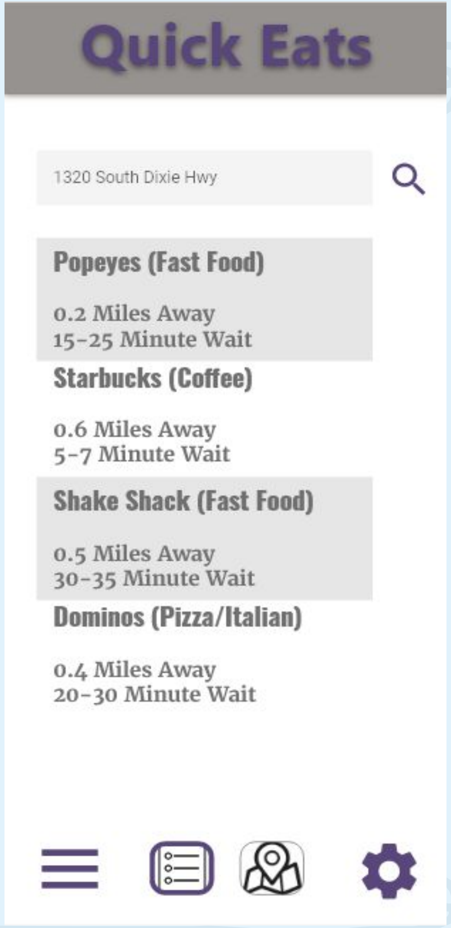
Map View

- Interactive map, color coded to give a general sense of wait times
- Search bar that allows you to change location
- Switch between list and map view on navigation bar



List View

- Listed restaurants, including distance and estimated wait time details
- Search and navigation bars identical to the map view page, allowing you to switch between views


Quick Eats

1320 South Dixie Hwy

Popeyes (Fast Food)

0.2 Miles Away
15-25 Minute Wait

Starbucks (Coffee)

0.6 Miles Away
5-7 Minute Wait

Shake Shack (Fast Food)

0.5 Miles Away
30-35 Minute Wait

Dominos (Pizza/Italian)

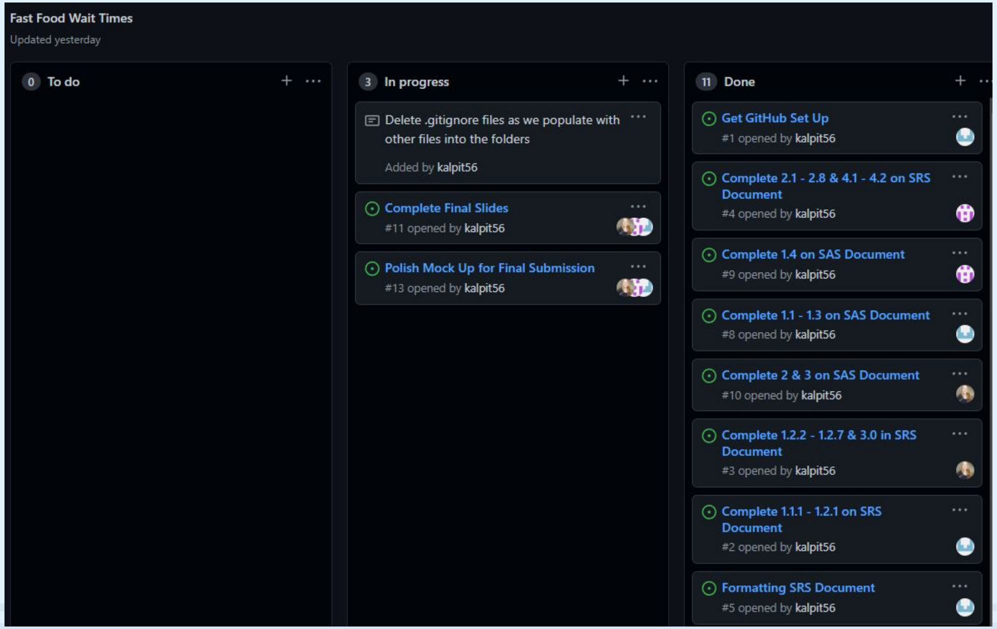
0.4 Miles Away
20-30 Minute Wait

Quick Demo

https://xd.adobe.com/view/0fa65ea4-8ded-423b-ad34-a025d7e30547-5af5/screen/32b5a276-9676-45da-bf08-d07a57cc1dc8/

Project Tracking & Link

GitHub Link: https://github.com/kalpit56/Quick-Eats



Fast Food Wait Times
Updated yesterday

0 To do

3 In progress

Delete .gitignore files as we populate with other files into the folders
Added by kalpit56

Complete Final Slides
#11 opened by kalpit56

Polish Mock Up for Final Submission
#13 opened by kalpit56

11 Done

Get GitHub Set Up
#1 opened by kalpit56

Complete 2.1 - 2.8 & 4.1 - 4.2 on SRS Document
#4 opened by kalpit56

Complete 1.4 on SAS Document
#9 opened by kalpit56

Complete 1.1 - 1.3 on SAS Document
#8 opened by kalpit56

Complete 2 & 3 on SAS Document
#10 opened by kalpit56

Complete 1.2.2 - 1.2.7 & 3.0 in SRS Document
#3 opened by kalpit56

Complete 1.1.1 - 1.2.1 on SRS Document
#2 opened by kalpit56

Formatting SRS Document
#5 opened by kalpit56

Questions?