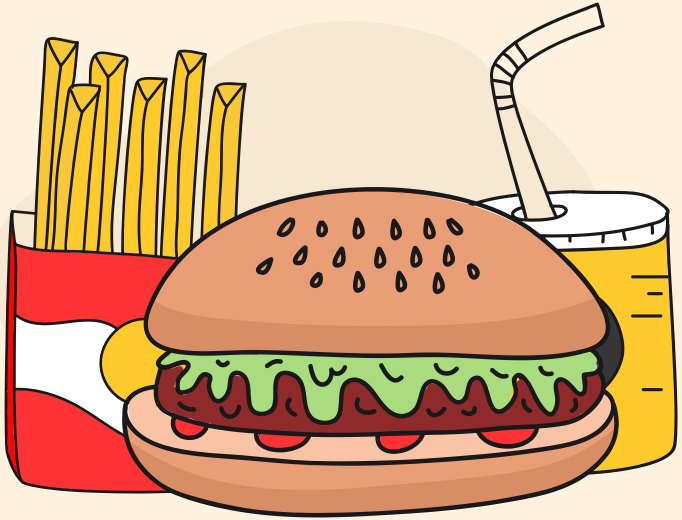


# Quick Eats

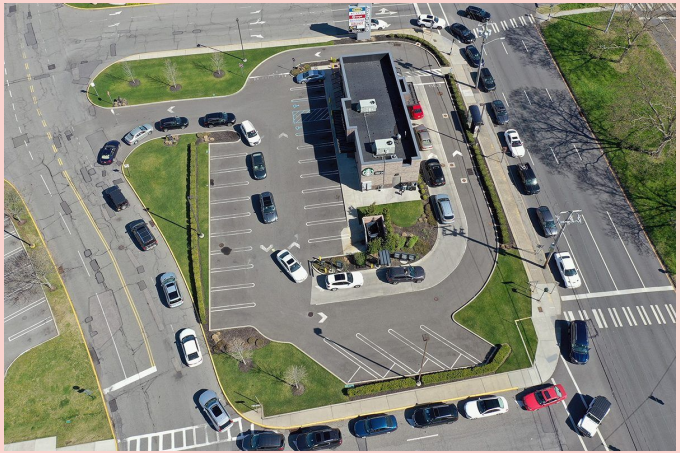
Kalpit Mody (Scrum Master)
Jack St. Hilaire (Developer)
Leah Harper (Developer)

March 01, 2022

# The Problem

# What is the issue at hand?

- Crowded cities lead to long wait drive thru times
- Increase in demand for drive thru service
- Demand for quick service



# Our Solution

# What is Quick Eats?

**A mobile application that analyzes and displays live drive-thru wait times at fast food restaurants**

Decreased time spent waiting in fast food drive-thrus
Reduced stress & pressure on fast food employees
Optimize traffic patterns in busy areas

# Functional Requirements

# FUNCTIONAL REQUIREMENTS (VIEWING RESULTS)

**View Welcome Page -** User views welcome page where they can leverage our search feature

**View Restaurants on Map -** User views fast food restaurants near a specified location on a map

**View Restaurants in List -** User views search results in a list

**View Favorite Restaurants -** User clicks a favorites tab to view their favorite restaurants

**View Estimated Live Wait Times -** User clicks on a fast food restaurant which tells users of the live estimated wait time

# FUNCTIONAL REQUIREMENTS (SEARCHING)

**Ability to Search -** User has the ability to use their current location or enter a location to search for fast food restaurants

**Ability to Filter and Sort -** User has the ability to select pre built filters and sort their search results in the list view

# FUNCTIONAL REQUIREMENTS (MANAGING ACCOUNT)

**Sign Up -** User provides an email, phone number, or Google account to register an account

**Log In -** User enters their credentials to gain access to our application

**Create User Profile -** User enters personal information along with preferences such as favorite restaurants and location

**Log Out -** User is able to sign out of their account

# Non Functional Requirements

# NON FUNCTIONAL REQUIREMENTS

**Useful Notifications -** Users will be notified about relevant (exceptionally short or long, etc.) wait time lengths according to their preferences

**Self-Reported Data Analytics -** Users will be able to self report when drive through lines are long or short at various restaurants (like Waze)

**Network Security -** The application will be protected from unauthorized access and misuse

**User Data Privacy / Security -** All user data will be protected against unauthorized access through encryption

# NON FUNCTIONAL REQUIREMENTS - TIME RELATED

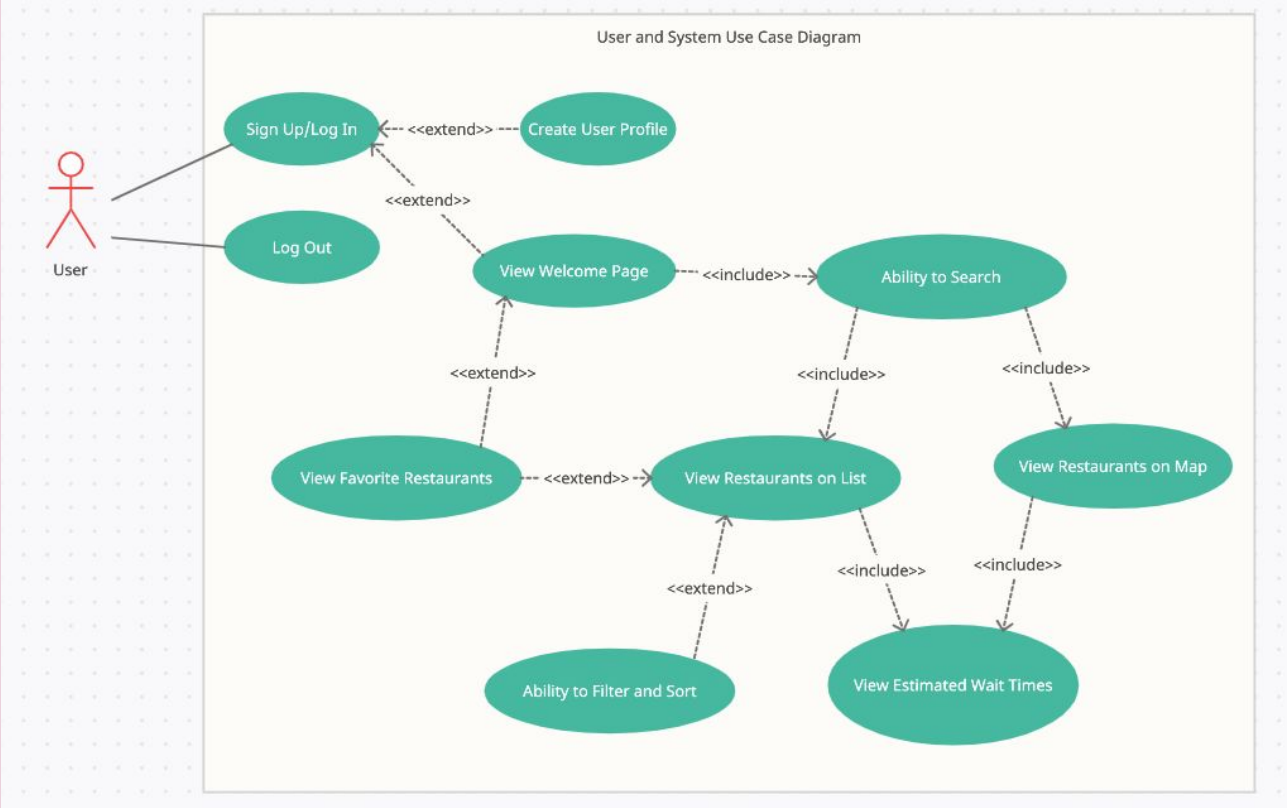
**Efficient Filtering and Sorting** - Filtering and sorting should return in less than 500ms

**Efficient API Interactions -** API Interactions should return in less than 0.5 seconds

**Search Result Time -** Search results should return in less than 5 seconds

# Use Case

# USE CASE DIAGRAM



# System Constraints

# Tool and Language Constraints

**React (Mobile Application Framework)** - We will utilize the react framework to deploy both the back and front end of our application

**Amazon Web Services** - AWS cloud storage will allow us to maintain scalability and secure data storage capabilities

**JavaScript** - Using JS with the React framework will allow us to develop for multiple platforms concurrently

**SQL** - Database interaction language that we will use to communicate with relational databases that we construct for this application

# Platform and Hardware

**Mobile App Platforms** - Through utilizing React, we will aim to develop for all popular platforms such as IOS, android, etc.

**Mobile Device** - Underlying hardware will guide design choices throughout the process, focusing on special considerations for popular phone models is key

# Network, Deployment, and Transition

**Internet Connection** - The application would not function properly if users do not have a connection to the internet

**Deploy to App Stores** - Using Firebase to deploy, we must consider all rules and regulations of various stores and marketplaces

**Timeline** - This project must be completed by the end of this semester

**End of Life** - When this class ends, someone else will have to take over the implementation of this project

# Evolutionary Requirements

**View Analytics (Functional)** - We will collect data as the app is being used to increase efficiency and usability over time



**Precondition** - Users must go through process of using the app for an extended time period

**Postcondition** - The data we gather will be able to help guide decision making based on trends

**Maintain Accurate Data (Non-functional)** - To maintain privacy and security for our users, data must be both accurate and secure

# Questions?