

# KETCHUP ON THYME

*No more door hopping looking for a table*

*Prepared by Kenneth Koh*

## 1.0 PROJECT OVERVIEW

### 1.1 DESCRIPTION

Ketchup on Thyme is a data-centric, web application that provide users with live data on the occupancy of a restaurant at any given time. To make this work, customers and restaurants would have to be using the same application and the app's data would be created and destroyed by restaurant hosts as customers enter or leave the restaurant.

### 1.2 PROBLEM

Imagine being out downtown on a Friday night in a group of 7 looking for a place to hang out, any place. I've been there - risky. Now imagine knowing beforehand that there are 3 places within a 400m walking radius that are able to host your group if you turn up in the next 15 minutes. Ketchup on Thyme recognizes that your thyme is important, the more you walk around, frustrations increases, hope shimmers away, the less you ketchup.

### 1.3 USER STORIES

#### **1. Users - Hungry people looking for places to take them in.**

Users page would show restaurant ratings from web APIs, as well as individual restaurant's capacity updated by operations.

#### **2. Owners - Restaurant owners looking to invite people through their doors.**

Restaurant owners would have complete control over their own i) data structure and ii) users viewable content.

### 3. Operations - Waiting staff looking to better manage walk-ins.

Restaurant hosts would be able to update data to free up or take up spaces based on availability

## 1.4 REQUIREMENTS

- The web application's database must be structured to show tables as table numbers, capacity per table, and overall restaurant capacity. Simulating restaurant owners customizing application for their unique case.
- Inputs from the operations screen would be sent as data to database.
- User screen must have a dynamic Google Map section to search for restaurants.
- User screen must show restaurant ratings from 3 APIs: Google, Zomato, Yelp.
- User screen must be able to show in percentage, how occupied the space is.

## 1.5 TECH STACK AND APIs

- PostgreSQL database
- Express
- React
- Node JS
- Google Maps API
- Zomato API
- Yelp API

## 2.0 CLIENT-SIDE IMPLEMENTATION

- User's Homepage
  - Google Map with search bar
  - Ratings from 3 sources
  - Restaurant's status of availability in percentage
- Operation's Homepage
  - Number of tables and capacity in divs
  - Ability to update information of tables to add/subtract people
  - Restaurant's status of availability in raw numbers

## 3.0 SERVER-SIDE IMPLEMENTATION

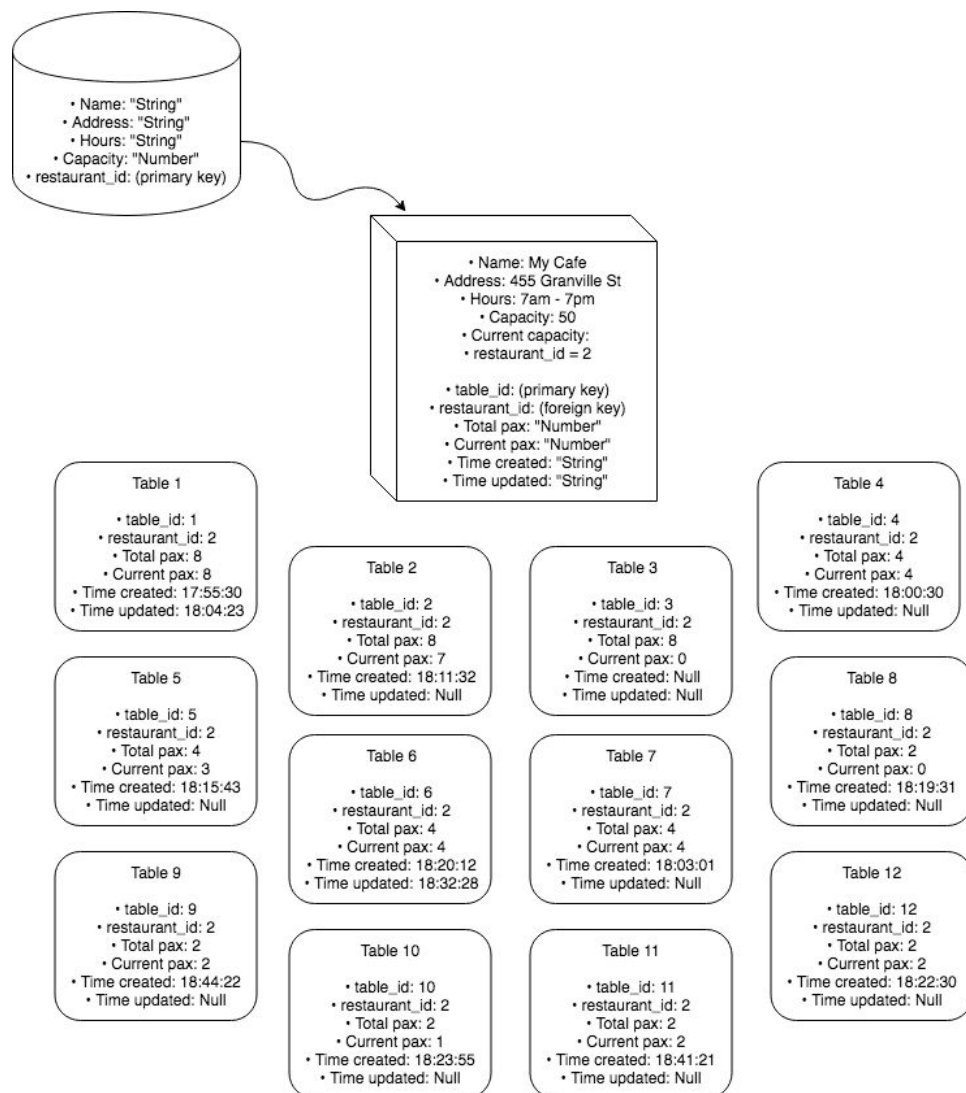
### 3.1 END-POINT DESCRIPTIONS

GET /user - to receive information from database

POST /operations - to change information on database

GET /owner - to view information from database

### 3.3 DATABASE STRUCTURE



## 3.2 PROJECT ROADMAP

<u>Day</u>	<u>Task</u>
1 (Tue)	Design database models.
2 (Wed)	<ul style="list-style-type: none"><li>• Add dummy data for 3 restaurants.</li><li>• Start work on express functionalities.</li></ul>
3 (Thu)	Create express endpoints, hook up with database.
<i><u>MILESTONE 1: BACKEND DONE AND TESTED =&gt; OUTPUT IN RESTFUL API FORMAT</u></i>	
4 (Fri)	Start on React structure, create components and routing.
5 (Sat)	Build react functionalities within front-end server.
6 (Sun)	Study API: Google Maps.
7 (Mon)	<ul style="list-style-type: none"><li>• Study API: Google Maps.</li><li>• Add functional Google Maps search bar into app.</li><li>• Implement map search &amp; pinpoint functionality into app's front end.</li></ul>
8 (Tue)	<ul style="list-style-type: none"><li>• Study API: Zomato, Yelp.</li><li>• Implement the 3 API's ratings of restaurants into app.</li></ul>
9 (Wed)	Link up full stack web application. Basic styling with Materialize UI and Bootstrap.
<i><u>MILESTONE 2: PROJECT FUNCTIONAL, READY FOR DEPLOYMENT</u></i>	