Git: https://github.com/Ricearoni26/CECS-491-Project

Product Requirements Documents

Team name: FutureGoogleDevs

Team Leader: Hadi Al Lawati

Joey Rice

Malhar Pandya

Michael Ching

Instructor: Arash Saifhashemi

California State University, Long Beach

College of Engineering

CECS 491A, Sec 07 122553,

Fall 2022 September 16, 2022

App Name: Crave

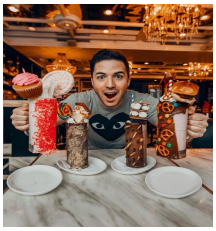
Elevator Pitch:

- Introduce your team:
    - Michael Ching
        - A software developer. He has been programming for about 5 years, and has taken courses in data structures, algorithms, and databases. His expertise is in the backend, as he is most familiar with Java and Python.
    - Hadi Al Lawati
        - An amateur computer science major with 22 years of experience in backend and frontend. especially (python and C++).
    - Joey Rice
        - A software engineer with 5 years experience. He specializes in C++, but knows 6 other programming languages. He has experience with app development using Google's Flutter open source framework.
    - Malhar Pandya
        - A software developer with 5 years of experience in Python, Java and C++. Malhar also has valuable experience in Databases and Machine Learning. He is most comfortable working on the backend but is eager to learn more about Software Engineering for the front end.
- Introduce a pain:
    - Are you indecisive? Have you ever been stuck trying to figure out what to eat?
- Promote a Product:
    - With our app, we can help indecisive people find appropriate restaurants for them.
- Describe your market:
    - Our app is built for anyone in the public that wants to find a place to eat.
- Who is your competition/How are you different:

- What makes our app special is that with our app, we can build profiles that are tailored to our users preferences. It can be used to suggest, find, and rate food in a user-specified zone or, if they're feeling adventurous, we can find a random restaurant fit for them.

How are we different to Yelp	Our app focuses heavily on the user experience and tailors the recommendations based on the user's profile. On top of this, we can choose random restaurants for you!
How are we different to Google maps	We make searching for restaurants even easier by tailoring the preferences according to the user's profile. We also allow the user to draw a custom area to search for restaurants!

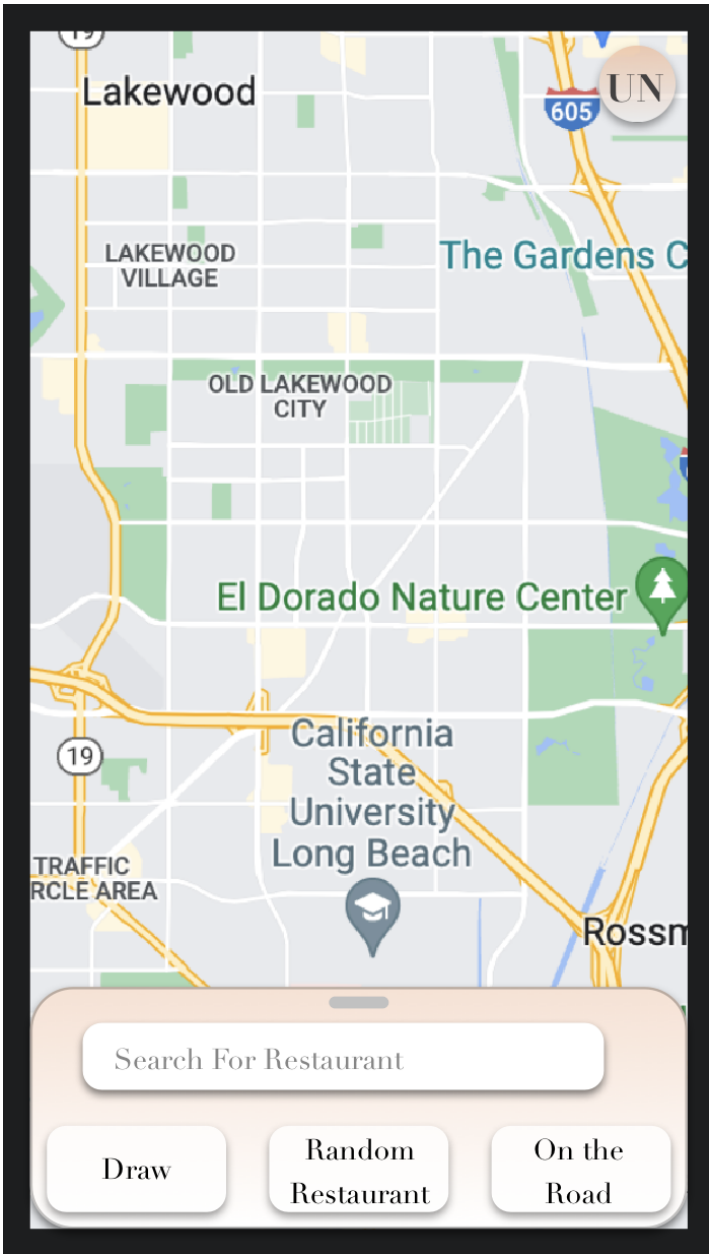
User Persona:

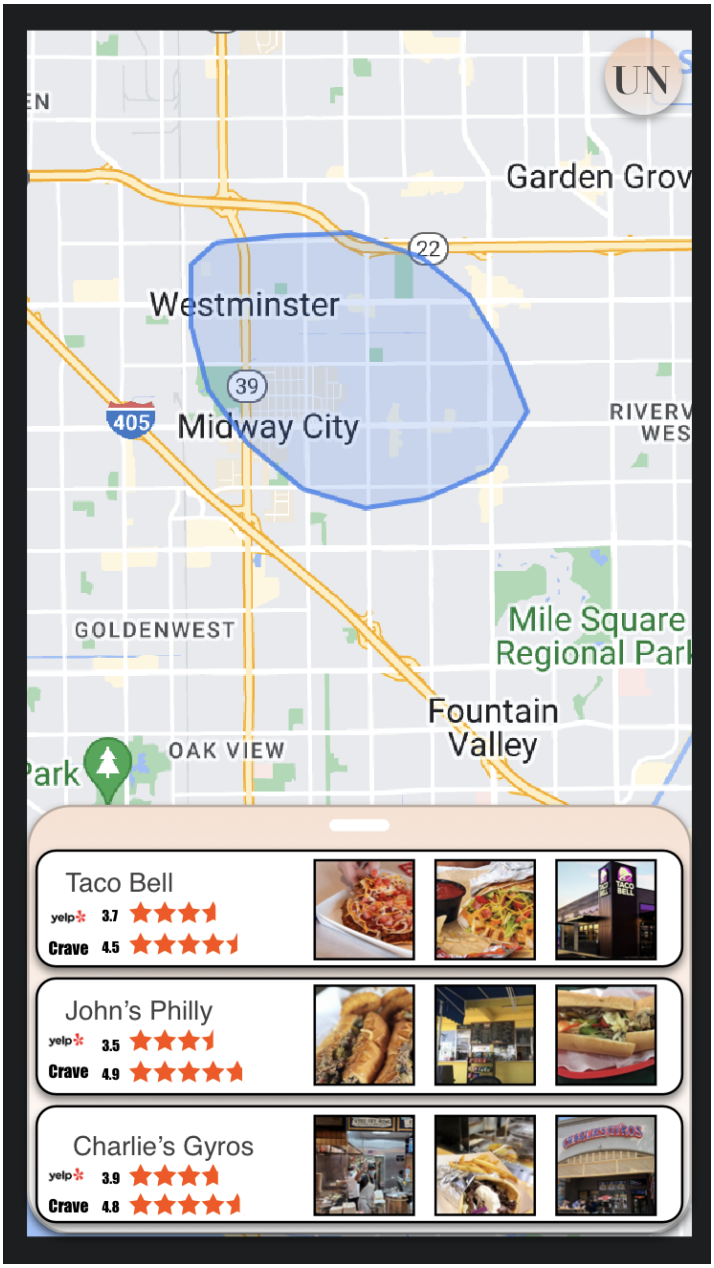
Food Enthusiast	Indecisive People	Low Spenders	Hungry People
			
Goal: Find a good restaurant that sparks their interest	Goal: Choose a restaurant for them	Goal: Finding a cheap restaurant	Goal: Find food quickly
Motivation: Being able to discover new restaurants	Motivation: Picking a restaurant that fits their demands	Motivation: Reasonable budget	Motivation: Satisfy their cravings based on a personalized profile
Frustrations: Not able to find restaurants that peak their interests	Frustrations: Not being able to choose a restaurant	Frustrations: Expensive restaurants	Frustrations: Not being able to find the right restaurant fast enough

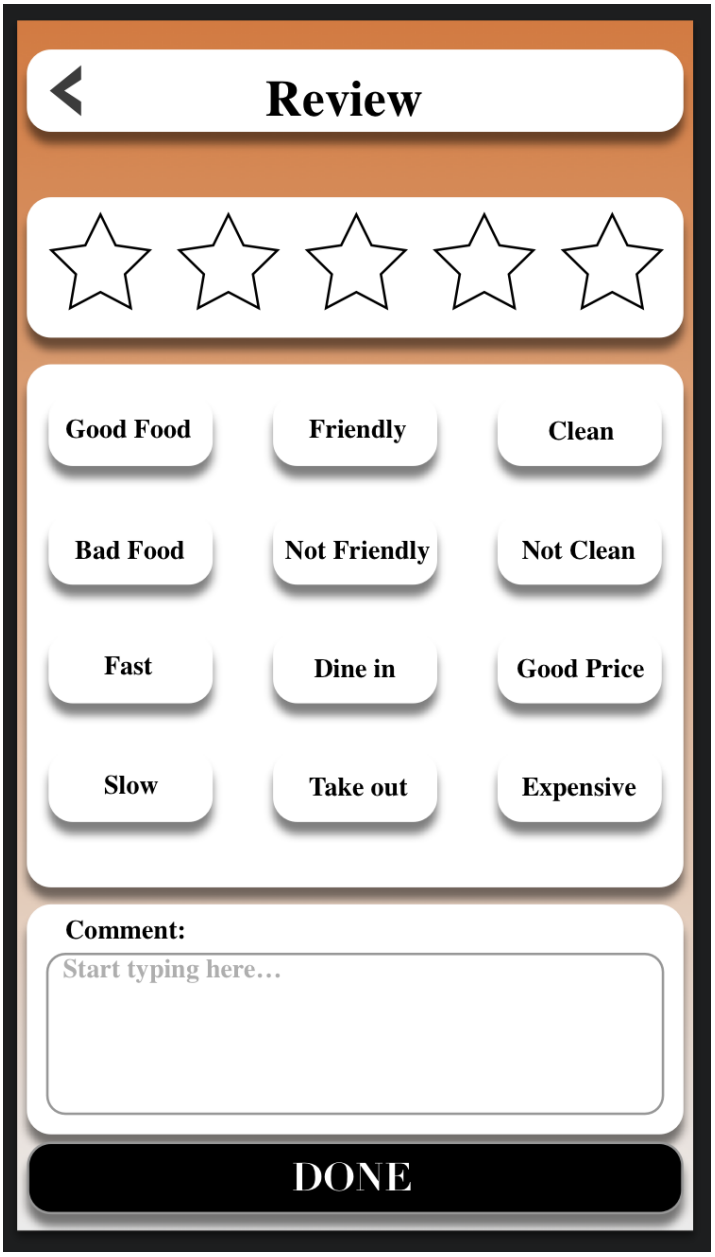
UI/UX:

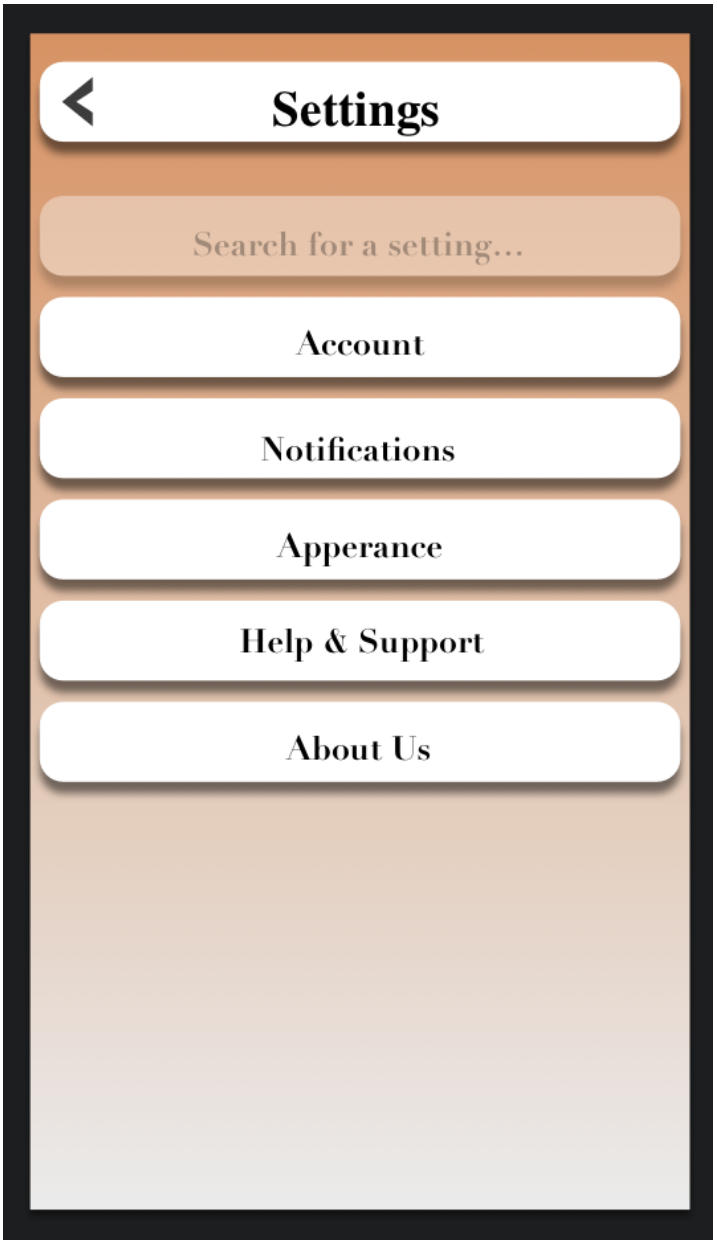
- External Interface Requirements:
    - User interface


Crave
email
password
Forgot password?
LOGIN
Sign up


Lakewood
605
LAKEWOOD VILLAGE
The Gardens C
OLD LAKEWOOD CITY
El Dorado Nature Center
19
California State University Long Beach
TRAFFIC RCLE AREA
Search For Restaurant
Draw
Random Restaurant
On the Road


UN
Garden Grov
22
Westminster
39
405
Midway City
RIVERV
WES
GOLDENWEST
Mile Square
Regional Park
Fountain
Valley
OAK VIEW
Park
Taco Bell
yelp 3.7
Crave 4.5
John's Philly
yelp 3.5
Crave 4.9
Charlie's Gyros
yelp 3.9
Crave 4.8


Review
Good Food
Friendly
Clean
Bad Food
Not Friendly
Not Clean
Fast
Dine in
Good Price
Slow
Take out
Expensive
Comment:
Start typing here…
DONE



- Hardware Interfaces:
    - Phone running android 6.0 and up or IOS 10.0 and up
- Software Interfaces
    - Phone running android 6.0 and up or IOS 10.0 and up
- Communication Interfaces
    - Dart will access the data and bring it to the users client

Use Cases:

P0	
Use Case 1	Register Account
Actors	User
Pre-Conditions	• User not logged in • Require network connection
Post-Conditions	• User profile is now displayed • Restaurant History available
Error-Condition	• Invalid user input • Invalid username or password • Confirmation Email not working
Non-Functional Requirements	• Logging in should take less than 5 seconds

P2	
Use Case 2	Delete Account
Actors	User
Pre-Conditions	• User must be logged in • Must be on account settings page • Require network connection
Post-Conditions	• User will no longer have access to the account. • Database is changed and updated.

Error-Condition	• Lost network connection
Non-Functional Requirements	• Data will be saved in another location.

P2	
Use Case 3	Update User Account
Actors	User
Pre-Conditions	• User must be logged in • Must be on account settings page • Require network connection
Post-Conditions	• User will have made changes to the account. Database is changed and updated.
Error-Condition	• Lost network connection
Non-Functional Requirements	• Updating should take less than 5 seconds.

P0	
Use Case 4	Search for Restaurant

Actors	User, Yelp API
Pre-Conditions	• User must be logged in • Must be on main page • Require network connection
Post-Conditions	• Users will have found the restaurant searched and will be guided.
Error-Condition	• Lost network connection • Restaurant not found
Non-Functional Requirements	• Searching should take less than 7 seconds.

P2	
Use Case 5	Randomize Restaurant
Actors	User, Yelp API
Pre-Conditions	• User must be logged in • Must be on main page • Require network connection
Post-Conditions	• User will receive a random restaurant and will be guided.
Error-Condition	• Lost network connection • No restaurant in area
Non-Functional Requirements	• Receiving a random restaurant should take less than 7 seconds.

P1	
Use Case 6	Review of Restaurant
Actors	User, Yelp API
Pre-Conditions	• User must be logged in • Must be on main page • Require network connection
Post-Conditions	• Review sent to yelp • Preferences for current user updated
Error-Condition	• User exits the app
Non-Functional Requirements	• Posting review within 7 seconds

P1	
Use Case 7	Validate Login
Actors	Google Maps API
Pre-Conditions	• Network Connection • Must be on main page • Require network connection
Post-Conditions	• Google maps are displayed on screen • User is able to interact with them
Error-Condition	• Lost Network Connection • Credentials no match
Non-Functional Requirements	• Display map within 7 seconds

P1	
Use Case 8	Drawn searched area
Actors	Yelp API, User, Google Maps API
Pre-Conditions	● User must be logged in ● Network connection required ● Must allow user to fully draw a closed shape before starting the search
Post-Conditions	● Search area will be specified by the drawn area. ● Restaurants within the drawn area will be shown
Error-Condition	● Lost Network connection ● Search is made outside the drawn area ● No restaurants exist within specified area
Non-Functional Requirements	● Search must be completed within 10 seconds

P1	
Use Case 9	En-Route Restaurant Search
Actors	Yelp API, User, Google Maps API
Pre-Conditions	● User must be logged in ● Network connection required ● User must be enroute to a specific destination

Post-Conditions	• Restaurant en route will be displayed
Error-Condition	• Lost Network connection • No restaurants nearby
Non-Functional Requirements	• Search must be completed within 10 seconds

P1	
Use Case 10	Profile Preferences
Actors	User
Pre-Conditions	• User must be logged in • Network connection required
Post-Conditions	• User Preferences updated • Search algorithm updated
Error Condition	• Lost Network connection • Does not fill out form
Non-Functional Requirements	• Results must be saved within 5 seconds

Non-goals:

- This is not a food delivery or ordering service. Just finding a restaurant.

Non-functional system and environment requirements:

- The application should be able to support at least a hundred users.
- Updates to the database should take ten seconds maximum, from the time it was sent by the user, till the time the database was updated
- The password should be encrypted

- The app should be available on both IOS and Android.
- The app should be up and working 24 hours 7 days a week with some minor hiccups depending on servers.