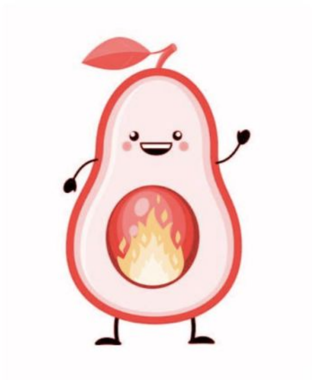
# Spicy Avocado

| | | |
|---|---|---|
| |  | |

**Project Name**: Blüber TBD

**Problem**:  People that use blue bikes cannot easily get an estimated time of how long travel takes because they use a combination of walking and biking to reach their destination, which is not currently supported by Google Maps.

**Purpose:** Solve the problem by providing a travel time planning app for traveling with Blue Bikes.

**Why the project matters:** Busy people often use blue bikes to go to work or school. As a result, reaching their destinations on time is important. By providing an accurate estimate of their travel, people will be able to travel without worrying about missing their time-sensitive events.

**How the app works:** Users enter their desired start and end location in the Boston Area, and the app outputs the most optimal (in terms of time or costs) **trip** between these two **locations** represented as a sequence of (walking and/or biking) **routes** between intermediate **stops**. The app will also output the total expected **duration** of the trip.

**Key Concepts:**

- Location
    - Purpose: Identifies points of interests on Earth in the Greater Boston Area
    - Structure: Contains longitude, latitude
    - Actions:

- create(lat: number, lon: number, out loc: Location)
- delete(loc: Location)
- getLat(loc: Location, out lat: number)
- getLon(loc: Location, out lon: number)
- Tactic:
  - If a location is created and not deleted, then the latitude and longitude of it can be obtained to represent a point of interest in the Greater Boston Area

- Duration
  - Purpose: Quantifies how long something will take
  - Structure: Time in seconds
  - Actions:
    - create(t: number, out d: Duration)
    - show(t: Duration, out t: number)
    - delete(d: Duration)
  - Tactic:
    - If a duration is created and not deleted, then the number of seconds of it can be obtained to show how long it will take
- Route:
  - Purpose:  To better categorize the time a portion of the total journey will take depending on its mode of transportation
  - Structure: has start Location, end Location, expected Duration to travel from start to end, and transportation type, which is a string of values either "walking" or "biking"
  - Actions:
    - create(s: Location, e: Location,  d: Duration, t: String, out r: Route)
    - delete(r: Route)
    - getStart(r: Route, out s: Location)
    - getEnd(r: Route, out e: Location)
    - getDuration(r: Route, out d: Duration)
    - getTransportation(r: Route, out t: String)
  - Tactic:
    - If a route is created and not deleted, we can obtain the time needed for this route given the transportation chosen to traverse this route
- Hub:
  - Purpose: Specifies the BlueBikes dock locations
  - Structure: Is a subset of Location; has a unique identifier
  - Actions:
    - create(l: Location, id: Number)

    - delete(l: Location)
    - show(id: Number, out h: Hub)
  - Tactic:
    - If a hub is created and not deleted, then we can show its location
- Stop
  - Purpose: Distinguishes two different routes of a total trip, and allows users to know how much time they need to wait between routes
  - Structure: Contains a Hub and an associated wait time, belongs to a trip
  - Actions:
    - create(h: Hub, out s: Stop)
    - show(s: Stop, out h: Hub)
    - getWaitTime(s: Stop, out d: Duration)
  - Tactics:
    - If a stop is created and not deleted, it can be used to detect the time a user will have to stay waiting after finishing one route and before starting the next
- Trip:
  - Purpose: Provides a way for users to plan in distance and in time their entire journey, using different modes of transportation to get from one point to another
  - Structure: Contains 1 or more Routes and Stops in sequence; start Location is the start Location of first route, end Location is the end Location of last route; expected Duration will be the sum of the expected durations of all its routes plus the wait time at the Stops
  - Actions:
    - create(r: Route[], s: Stops[], out t: Trip)
    - delete(t: Trip)
    - getDuration(t: Trip, out d: Duration)
    - getStart(t: Trip, out s: Location)
    - getEnd(t: Trip, out e: Location)
    - getRoutes(t: Trip, out r: Route[])
    - getStops(t: Trip, out s: Stop[])
  - Tactic:
    - If a trip is created and not deleted, then we can obtain the necessary routes and stops required and the expected time the total trip takes to go from start to end.
- Account
  - Purpose: Allows users to save favorite trips for easy access and selection in the future

    - Structure: Contains a unique username, and has associated with it any trips of the user
    - Actions:
        - create(u: String, out a: Account)
        - saveTrip(t: Trip, a: Account)
        - removeTrip(t: Trip, a: Account)
        - getTrips(a: Account, out t: Trip[])
        - delete(a: Account)
    - Tactic
        - If an account is created, not deleted, users of the app can add a trip if it doesn't already exist and delete any pre-existing trips from the account to save their favorite trips

**Why will the project involve substantive conceptual design?**

- Our travel planning app combines more than one type of transportation
    - Google Maps supports public transit with buses and trains, but not transportation with Blue Bikes
    - This is represented with our Trip concept, which is not widely used
- Blue bikes cost varies based on rent time, so we will have to design our app such that it either minimizes travel time, travel cost, or both (where the choice is up to the user)
- We will have to make API calls to both Blue Bikes and Google Maps so we will have to design our app to smoothly integrate both types of API calls