

Agenda:

- 1) Progress
- 2) Plan out UI
- 3) Use UI plan to inform choice of features for MVP
- 4) Make sure we are all on the same page with concepts
- 5) Assign Parts to work on for 11/11 deliverable

Progress:

- 1) Determined app idea
- 2) Looked into APIs, Locu seems promising
- 3) Looked into existing apps, nothing seems to do what we will do

Minutes:

- 1) UI Plan:
 - a) Mostly contained in single-page app.
 - b) User profile page and initial app walkthrough sequence separate from main single-page structure
 - c) Main features for MVP:
 - i) Map with restaurants
 - ii) List of meal recommendations, each with:
 - 1) place
 - 2) price
 - 3) meal
 - 4) description
 - 5) justification
 - 6) favorite/forbid button
 - iii) Filter meals based on search box, price
 - iv) User profile/Taste profile editing
- 2) Concept clarifications:
 - a) taste profile. Made up of the following:
 - i) 'dietary_restrictions' - ingredients or foods that the user cannot eat, whether because of diet constraints (vegan, vegetarian, ...) or allergies
 - ii) 'likes' - ingredients or foods that the user likes
 - iii) 'dislikes' - ingredients or foods that the user tends not to like
 - iv) 'favorites' - particular meals that the the user thinks look good
 - v) 'forbidden' - particular meals that the user does not want to eat
 - b) making recommendations
 - i) Recommendation engine works as black box. Takes a user object and a meal object and returns an integer score

- ii) Immediately return a score of $-\infty$ (or some other lower bound score) if the meal has been specified as forbidden or if contains any ingredients in the 'dietary_restrictions' list
 - iii) for each ingredient that is in the user's 'likes' list, add some points
 - iv) if the meal is marked as a favorite, add a bunch of points (more so than for a liked ingredient)
 - v) for each ingredient in the dislikes list, subtract a number of points that is greater than the number of points added for a like, but less than the number of points added for a favorite
- 3) 11/11 Deliverable: division of labor. These are the items that need to be included in the deliverable, accompanied by which team member is responsible for each item
 - a) Design Doc
 - i) Motivation (Danielle)
 - ii) Context Diagram (Harihar)
 - iii) Data Model (Harihar)
 - iv) Concepts (Tawanda)
 - v) Security Concerns (Ryan)
 - vi) UI (Danielle)
 - vii) Design Challenges (Ryan)
 - b) Teamwork Plan (Tawanda)