CS 242 Final Project Proposal

WTE
Xilun Jin (xjin12) , Marco Valentino
Zhiwei Zhang (zzhng107) , Marco Valentino
Chen Cheng(chen228) , Fang Guo

Abstract

Background/Motivation

It’s a hard problem every single person might face every single day more than 2 times in average. W hat do I have for meal today? People don’t really treat this problem seriously and never feel it annoying until they are actually facing it. Although they try to be thoughtful, none of them could really make a satisfying decision even after browsing all restaurants available on Yelp. Now, it’s time for our app.

Technical Specifications

Platform

iOS App

Programming Languages

Swift

Stylistic Conventions

camelCase

SDK

Xcode

Database

Realm

Functional Specifications

Basic Interactions

- Daily recommendation with tinder-style swiping and matching.
- Browse through the foods gallery and choose SAVE FOR LATER, DISLIKE or EAT THIS.
    - SAVE FOR LATER would mark food as potential choices for future and would slide in next choice.

    - DISIKE would make food and the similar appear less frequent.
    - EAT THIS will remember users' like and lower the recommendation frequency in certain amount of time but might be recommended again later.
- View the detail of the food including the location, the review and etc.
- View your SAVE FOR LATER, DISLIKE or EAT THIS food lists.

Timeline

Week 1

First Part of the Frontend

- Dish detail page
    - Basic info of the dish (2.0)
        - Basic objects: Images, labels ...
    - Rating component (2.0)
        - Able to set the size and number of items
        - Able to change the style
    - Map (2.0)
        - Google map library
    - Adaptive Layout (2.0)
    - Testing (2.0)

Week 2

- Database (2.0)
    - Collect the data from the yelp
    - Clean up the data
    - Import data to firebase
- APIs (4.0)
    - **Get /dish?key={$key}**
      Get a dish by Id. Return all the info about that dish
    - **Post /rate**
      Update the rating of a dish in a user's history.
    - **Post /savehist**
      Save a new dish in a user's history list
    - **Post /saveforlater**
      Save a new dish in a user's saveforlater list
    - **Post /getHistory**

    Get a list of dishes in a user's history
  - **Post /getSaveForLater**
    Get a list of dishes in a user's saveforlater list

- Fontend (2.0)

  - History/SaveForLater Page
    - Display a list of dishes
      - Basic info: Date, name, image, rating
  - Filter Page
    - Filter by rating
    - Filter by price
    - Filter by distance

- Testing (2.0)

  - UITest
    - Check if elements exist in pages
  - APITest
    - Postman
  - Unit Test
    - Test functions

Week 3

- Integration with backend (2.0)

  - Restaurant Detail Page
    - Call `/dish`
  - History/SaveForLater Page
    - Call `/gethist`
    - Call `/getsaveforlater`

- Delete dish (2.0)

  - Fontend:
    - Delete button
  - Backend:
    - **Post `/deletedish`**
      Delete the dish in a user's history or saveforlater list

- Local storage (2.0)

  - History list info
  - SaveForLater list info

- Navigation and data flow (2.0)

  - user logined in

    - dish selected

- Testing (2.0)
    - UITest
    - APITest

Week 4

- Integrations of all the parts(2.0)
    - Navigation
        - Mainpage -> History/Saved/Profile
        - Profile -> History/Saved
- Filter Page(2.0)
    - Enable the buttons
    - Pass data back to the main page
    - Data persistency
- Clean up the code (2.0)
    - Code Refactor
    - Comments
- Enhancement (2.0)
    - More buttons in the detail page
    - Spinner in the dish list pages
    - Adaptive
- Testing (2.0)
    - UnitTest