CS 242 Final Project Proposal Template
<Restaurant picking elf>
<Jessie Le> (<kle11>), <Moderator>

1. Abstract

1.1. Project Purpose

What is the purpose of this project? What problems do you want to solve?

- Help people make decision on picking restaurants by picking restaurants randomly.

1.2. Background/Motivation

Why are you interested in doing this project? Have you worked on something similar before?

- I always find myself do not know where to eat after whole day of coding in Siebel although I have some restaurants in mind.

2. Technical Specifications

2.1. Platform: (i.e. Website, iOS App, Arduino hack)

- React Native App

2.2. Programming Languages: (i.e. PHP, Python, Lisp)

- React native

2.3. Stylistic Conventions: (i.e. commenting, naming conventions, camelCase)

- Comment each function
- Naming variable using camelCase

2.4. SDK: (i.e. Pebble SDK, LAMP Stack, Android SDK)

- WebStorm

2.5. IDE: (i.e. Eclipse, Visual Studio)

- WebStorm

2.6. Tools/Interfaces: (i.e. Android phone, Bluetooth tag, Google Chrome)

- Xcode simulator

2.7. Target Audience: (i.e. Botanists, iPhone users)

- Hungry people
- People who find hard to make a decision
- iPhone users

3. Functional Specifications

3.1. Features

Several bullet points of what kind of functionality your project will feature. What should the user be able to do?

- The user will be asked to pick several restaurant they want to go by searching and selecting options from api. They can save restaurant they love to his or her list. They can get a random result from what they have selected. They can randomize the result as many time as they want to.

3.2. Scope of project

What are some of the limitations of this product?

- Cannot give recommendation on restaurant picking but provide a randomized choice based on what people picked.

4. Timeline:

4.1. Week 1

- Do research, find a restaurant api and learn how it works.(Maybe yelp or google api)
  - Find a great restaurant api and learn about it.
  - Find some related api.(Show what found)
  - Try to find api but failed.(Show what found)
  - No work.
- Create a page show all restaurant and load data from api to it.
  - API data are used not fake data.
  - UI looks good.
  - UI not working but some amount of code.
  - No work.
- Create a list page that can add restaurant from previous restaurant page
  - Restaurant can be added to the page and UI looks good.
  - UI looks good.
  - UI not working but some amount of code.
  - No work.
- Build layout to be adapted.
  - Layout adapted to different size of screen as well as horizontal and vertical display.
  - Layout adapted to horizontal and vertical display
  - Some functions not displayed.
  - No work
- Test.

4.2. Week 2

- Implement searching bar at top of the screen to select restaurant.
  - Implement searching bar with sorting feature.
  - Implement searching bar.
  - Implemented searching bar not working but some amount of code written.
  - No work.
- Store restaurants data locally.

    - Use a clever schema to store all data locally
    - Store all data locally in backend.
    - Try to store locally but did not work.
    - No work.
- Direct user to a webpage which shows decided restaurants information.
    - Display webpage info in a app view(next week)
    - Successfully direct to the external webpage
    - Try to direct to the external webpage but fail(show code)
    - No work.
- Authentication access.
    - Login page
    - Authentication implemented
    - Authentication not working but show some code
    - No work.
- Test.

4.3. Week 3

- Create favorite page and read from user favourite from api.
    - Great clear UI read data from api.
    - Functional UI.
    - Cannot read data from api but some codes are shown.
    - No work.
- Allow user to favorite restaurants.
    - Set favorite restaurant through api call.
    - Cannot send data to api but some codes are shown.
    - No work.
- Create a history page add the restaurants to history page and after people select the restaurant.
    - Both send and get data from api works.
    - Can only add after selection.
    - Cannot set and get data from api but some codes are shown.
    - No work.
- Allow user to delete histories.
    - Delete history restaurants through api call.
    - Cannot send request to api but some codes are shown.
    - No work.
- Test.

4.4. Week 4

- Allow user to add restaurant from favorite view to list view.
    - Allow user to add restaurant in favorite to restaurant picking. Design a great UI for it.
    - Allow user to add restaurant in favorite to restaurant picking.
    - Did ui but not fully functional.
    - No work.
- Create restaurant info page for selected restaurant.
    - Create a restaurant info page fit all restaurants with great ui.

    - Create a restaurant info page fit most restaurants with okay ui.
    - Create a restaurant info page fit only one restaurants with okay ui.
    - No work.
- Polish UI by making searching page to a gallery view with restaurants image which easy for user to pick.
    - Create gallery view with real images from the api. Selecting image is fully functioned in a fancy way.
    - Create gallery view with real images from the api. User can add the restaurant by clicking the image
    - Create gallery view with real images from the api. Clicking the image to select the restaurant is not functioning.
    - No work
- Debug and implement one feature that was not working if applicable.
    - Eliminate the bugs and add reasonable amount of comment in a more uniform way
    - Eliminate the some bugs and add comment in a more uniform way
    - No work.
- Test.

5. **Future Enhancements**

What are some cool tweaks you'd want to make to your product after the core functionality is done? Are you planning to work on it in the future?

- Connect it to a mysql database to manage personal profile and try to develop the app with more social activities.