Software Requirements Specification

Ethan Kolkmeier, Filimon Gebrekidan, James Durflinger, Carl Trautwein

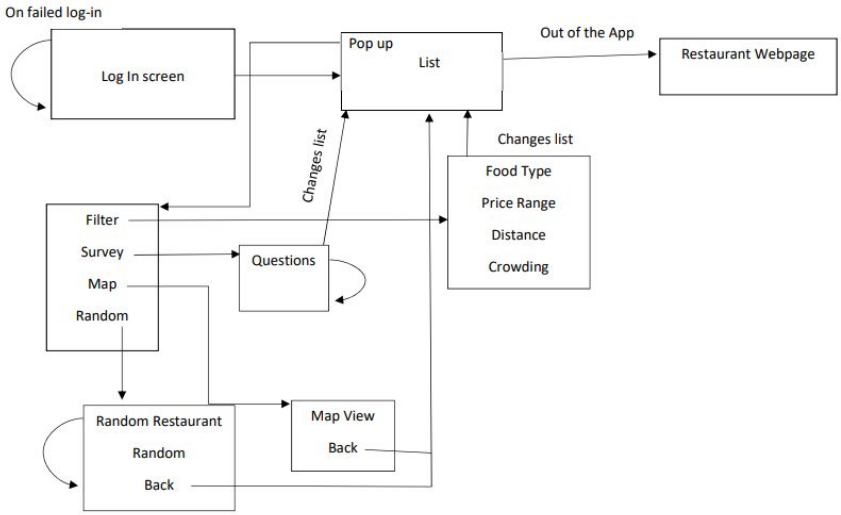
Introduction

Restaurant Finder is an application that helps users make dining decisions for themselves, or a group. The application provides the user with relevant information about each restaurant, such as distance, crowding, and relative price. This will also allow users to experience new restaurants and food they have never tried before.

- Mbps: Megabits per second
- RF: Restaurant Finder
- ms: millisecond
- UI: user Interface

This document will explore the scope and features of RF using the IEEE Requirements Standard. Following will be a high-level view of RF, and then specification details. Finally, any supporting information will be at the end of the document.

Description



Restaurant finder will be a new app that is started from scratch. The origin of the app was the problem that we all have, not knowing where to eat.

Restaurant Finder will compile a list of all restaurants within 50 miles of the user. Using this data RF will be able to show the user some suggestions. The app will also offer users a short quiz and make restaurant suggestions based on the results. If all else fails RF will offer the user a “Random” option that picks a random restaurant from the list. Each user will have a unique username and standard password which they will use to sign in. Each user will be able to assign their own restaurant preferences, which the app will remember and suggest more often.

Primary users of RF will be people who eat out often. A typical user may be a young student who eats at various fast-food restaurants. Other expected users will be people who are tired of eating the same food and need help discovering new places. However, anyone who wants to eat out for any occasion could be assisted by RF.

The application must be functional in English speaking and developed regions of the world. The app must also be able to perform with 5 Mbps and above (which should be sufficient for performing a search query).

Requirements

External Interface

The application's UI will include a login screen, which will have text boxes for users to input data, such as user name, email address and age. Once a user is logged in, the app will have a search screen for sorting through restaurants based on user specifications. The app will also have a screen for displaying found restaurants in list form, each object in the list should have the functionality to redirect to the restaurant's website if one is available. In addition, there will be a map of the surrounding area which displays both the user and the closest 10 restaurants. The app will also have a survey mode where it asks the user multiple choice questions (ie “How does Mexican sound?”) and then display a list of possible choices. A random button will be available that displays one random restaurant, this button can be pressed as many times as the user wants.

Functional Requirements

The application will require a user to login to an account, if the user input incorrect credentials they should not be allowed to continue until they make an account. Each username per device must be unique and will be stored locally, while each password may be unique but it is not required.. Each device should be able to hold several distinct users. The application will allow the user to filter through restaurants with terms such as maximum distance for searching (default 50), restaurant type, minimum restaurant rating, and relative price. The app should be able to remember a user’s favorites and list them. The app should also be able to pick a restaurant based on a user survey.

Performance Requirements

The sign in process should take no more than 50 ms because of the data being stored locally. Upon sign in, the phone should get all restaurants within 50 miles which should take less than 1 second with an internet connection. If the user changes the search distance the app should query again up to a maximum of 200 miles and should take less than 10 seconds to return results.

Design Constraints

Because developing ios application requires an annual developers fee, the app will only be available on android devices [2].

Appendices

[1] Google Developers. (2019). *Overview | Places API | Google Developers*. [online] Available at: https://developers.google.com/places/web-service/intro [Accessed 23 Sep. 2019].

[2] T. Mackenzie, "App store fees, percentages, and payouts: What developers need to know," TechRepublic, 2019. [Online]. Available: https://www.techrepublic.com/blog/software-engineer/app-store-fees-percentages-and-payouts-what-developers-need-to-know/. [Accessed: 23- Sep- 2019].