Wifil

A crowd-sourced Wi-Fi hotspot location application for Android phones.

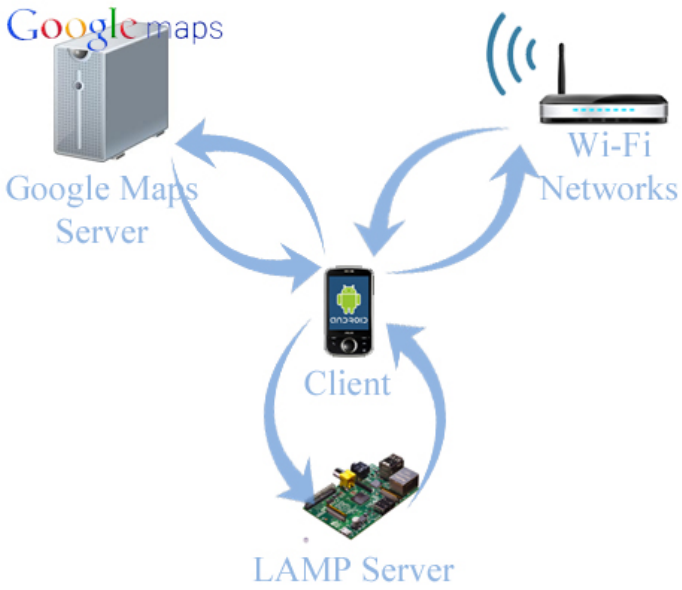
Team 6
Michael Clayton, Brandan Miller, Billy King, Bakytzhan Kudebayev

Requirements Document

A. Problem Statement:
Users have difficulty locating Wi-Fi hotspots offering public internet access.

B. Background Information:
There are millions of free public Wi-Fi hotspots around the world enabling people to gain access to internet if they know the locations of hotspots. We would like to help people locate these Wi-Fi hotspots more easily and faster.
The Wifil application makes it easy to locate free Wi-Fi hotspots. It filters the paid and locked Hotspots and shows you only free hotspots. Over time, some hotspots may change their settings and become locked or inaccessible. Therefore, we keep track of when hotspots' settings change and do not send users to locked or paid hotspots. Wifil shows you fast and reliable Wi-Fi connections on the go, based on our registered user ratings. Your phone will notify you if you are in range of free Wi-Fi hotspots. The application controls GPS inaccuracy through Wi-Fi triangulation and has fast and efficient lookup of hotspots based on location of user.

C. Environment & System Models:



- **Client:**
  Users will interact with an Android application. The Android-running device must support location services and Wi-Fi connectivity and must be running android 2.8 or newer. It will make requests to the server to acquire and refresh data.
- **Server:**
  The server will run Apache. PHP will be used to generate data for the client and mySQL will be used to store and manage data. Initially, the server software will run

on a Gen. 1 Model B Raspberry Pi with a transition to a more powerful server should the need arise.

- **Other:**
  The application will access the Google Maps API in order to display hotspot maps. Obviously, local Wi-Fi hotspots are required in order to collect and display data.

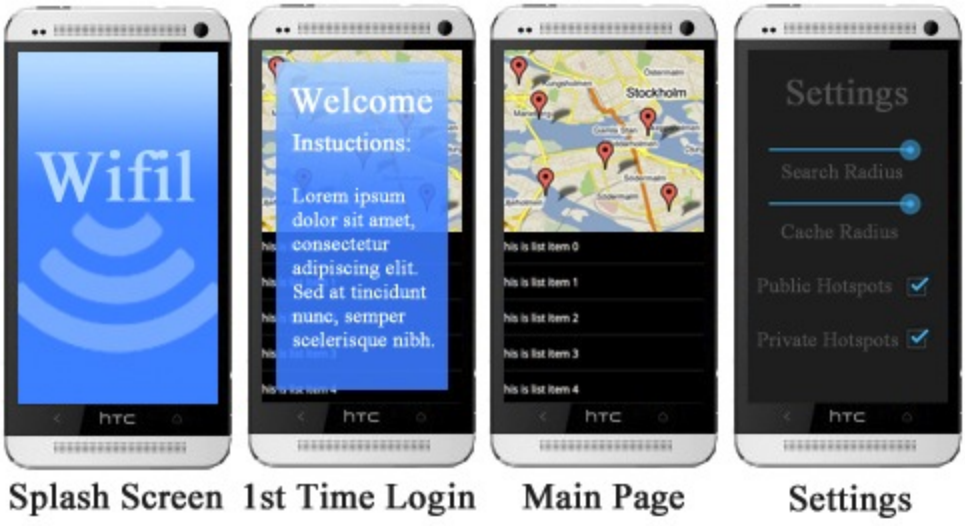
D. Functional Requirements:

- **Inputs:**
  The user of the application can add, rate, name, and describe publicly accessible Wi-Fi networks. While the user has the Wifil application active, Wifil will cache hotspot lists by radius and location. If any problem results regarding inaccurate information, there will be the capability of administrative management of the networks and users from a server-side front end.
- **Outputs:**
  The Wifil application will output to the user, Wi-Fi hotspots within a specified radius and will return details to the user about the Wi-Fi spot along with a display on Google Maps with Wi-Fi location pins. However, if no information is cached or if no network connection is available, then the user will receive an error message.
- **Computations and Timing:**
  Search results are returned within ten seconds of a request from a device with a stable internet connection.

E. Quality, platform and process requirements:

One of the biggest constraint for our application is to return search results within 10 seconds of data being input, in order to keep the user satisfied. With this constraint being imposed, we will also have the constraint to keep the speed for search completion to sublinear time. Also, the largest platform requirement is to make the application compatible with the android operating system version 2.3 (Gingerbread).

F. Use Cases:



(Diagram of Main Application Pages)

- **Case: User wants to find a local hotspot.**
  1. User specifies the use of current location.
  2. User specifies a search radius.
  3. Wifil displays hotspots within radius of current location.
- **Case: User wants to find a non-local hotspot.**
  1. User specifies an address for a location.
  2. User specifies a search radius.
  3. Wifil displays hotspots within radius of provided location.
- **Case: User wants to be notified of hotspots as he/she travels.**
  1. User turns on hotspot notifications.
  2. User specifies a search radius.
  3. Wifil displays notifications when the user enters within the given radius of a Wi-Fi hotspot.
- **Case: User wants to add a hotspot to the map.**
  1. User connects to a hotspot.
  2. User drops a pin via Google maps API at the current location.
  3. User inputs details about the network.
  4. Wifil saves this information to the Wifil database.