Name – Koushik Annapureddy Student Number – 84973F

OtaSizzle Location Application for Mac OS X

I have implemented a platform for collecting measurement of signals from RF Beacons like Wi-Fi Access points and Bluetooth devices on MAC OSX. These radio signals are used to determine the location context of the OtaSizzle users.

Main features present in the application –

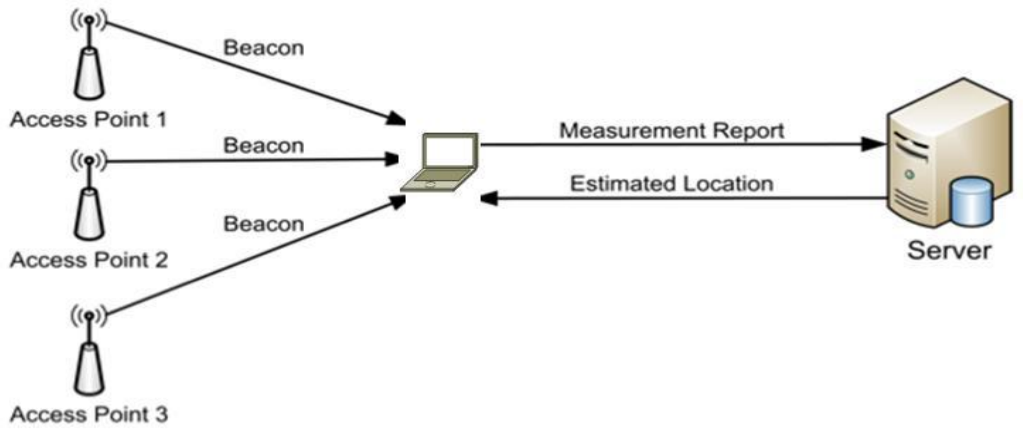
Wi-Fi Scanning

Bluetooth scanning

Offline Capability (MAC OSX and Windows)

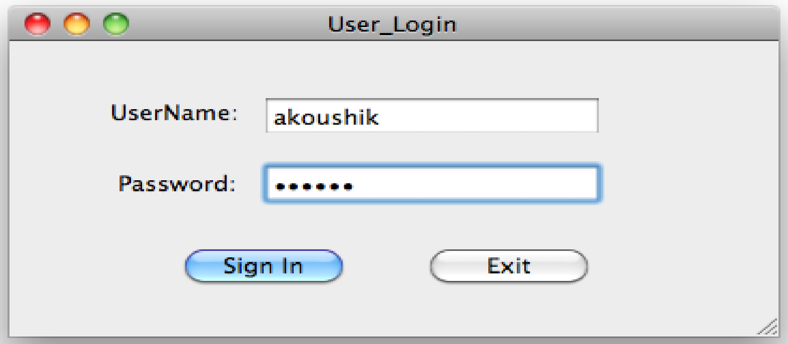
Design –

Choosing an appropriate design is one of the important considerations while developing location based services. In our implementation we are collecting the measurement of signals from surrounding Access points and Bluetooth devices and sending the report (which is prepared by application) and send it to server at regular time intervals. Server computes the estimated location based on the received measurement report and sends the results back to location. This mode of design helps in reducing the computation load from location devices, server takes care of all computational process, moreover it helps in upgrading the algorithm/system with ease since it doesn't require modification in all individual location devices, update in server is sufficient.



User Authentication –

During implementation of OtaSizzle location application user authentication is the important problem to be solved. The aim is to allow OtaSizzle user to reuse his/her OtaSizzle account for our application and there is no necessity of saving the user credentials in the database. Our application uses cookie to authenticate a user (Cookie is provided by OtaSizzle platform).



Here are the steps involved in User Authentication process –

1) User enters the username and password to the application and clicks on Sign In button
2) The location application creates a user session with the OtaSizzle platform using POST request which contains username and password in the body of the request.
3) OtaSizzle platform responds with the status code 201 created if user has entered correct username and password. The response message contains session cookie and OtaSizzle user id . Once this session cookie is received by location application it saves it in memory and uses the same session cookie for the rest of the communication.
4) From now onwards when ever location application needs to make a HTTP request to any of the REST service it includes the session cookie it received earlier.
5) Once the server receives the HTTP request message from the location application it will extract the session cookie from it and then it sends a GET message to the OtaSizzle platform including the session cookie to validate it.
6) If session cookie is valid then the OtaSizzle platform sends a HTTP response message with 200 OK otherwise it sends 404 Not Found.
7) If server receives 200 OK then the HTTP request from the location application will be processed and response is sent in JSON format, else 401 Unauthorized is sent back to the location application.
8) Depending on the response from the server, location application displays the result of user authentication to user.

Wi-Fi Scan -

CoreWLAN framework which is the Objective-C public API for the MAC OSX provides functionalities such as scanning for networks, querying the wireless interface for static and dynamic parameters etc. These are the three attributes the location application scans from the surrounding access points.

1) **BSSID – Basic Service Set Identifier**

2) **SSID – Service Set Identifier**

3) **RSSI – Receive Signal Strength Indicator**

After the successful authentication the Wi-Fi scan thread performs scan for all accessible Wi-Fi access points for every 10 seconds. Each scan will be added to scan window until the scan window contains six scans, it implies that it takes one minute for a single scan window to be created. The scan window will be uploaded to the server by making HTTP POST to the URL of the scan window. After the current scan has been successfully uploaded, a new empty scan window will be created for storing the subsequent scans. This process continues until the user signs out of OtaSizzle platform.

Sample Wi-Fi finger prints in JSON Format –

```
{

 {"mac": "94:c:6d:a7:e9:8c", "ssid": "TP-LINK", "rssi": -76},

{"mac": "0:19:5b:e5:9f:fe", "ssid": "Fudan", "rssi": -89},

{"mac": "0:19:db:95:51:61", "ssid": "MSI", "rssi": -62},

{"mac": "0:26:5a:2e:6d:a8", "ssid": "Reddy_King", "rssi": -49}]

},
```

Bluetooth Scan –

The **Bluetooth framework** contains the API you use to perform Bluetooth-specific tasks. With the methods and functions in the Bluetooth framework, you can create and destroy connections to remote devices, Discover services on a remote device, Perform data transfers over various channels, Receive Bluetooth-specific status codes or messages.

These are the three attributes the location application scans from the surrounding access points –

1) **Name**

2) **MAC Address**

3) **Class id**

4) **Major Device Class**

The Background Bluetooth thread performs a scan for all reachable Bluetooth devices every 30 seconds. A list of all detectable Bluetooth devices also called as Bluetooth scan are uploaded to server using HTTP POST. This process continues until user is logs out of the application.

Sample Bluetooth scan –

```
"device_list": [
```

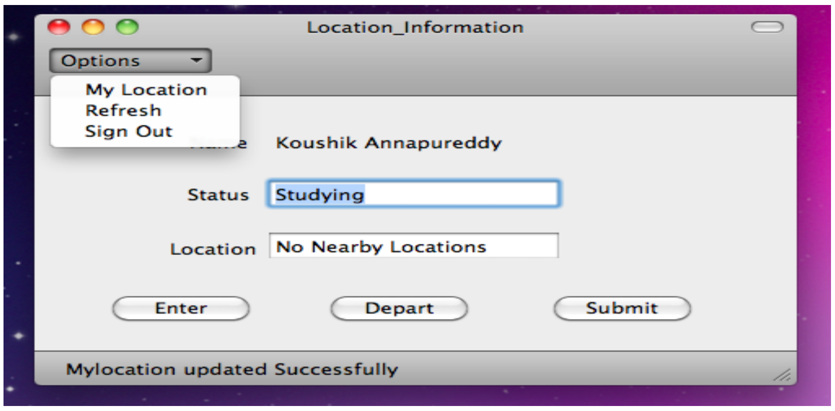
{"name": "KOUSHIK-PC", "address": "00-25-56-D8-84-92", "class_id": "4063492", "dev_class": "Computer"}

{"name": “Nokia E71", "address": "00-25-CF-FD-F5-70", "class_id": “5898764",   "dev_class": “Phone"}

{"name": “Microsoft Wireless Notebook Presenter Mouse 8000", "address": "00-1D-D6-38-75-W8", "class_id": “9600", "dev_class": “Peripheral"}

]

Other Features -



Name Koushik Annapureddy

Status Studying

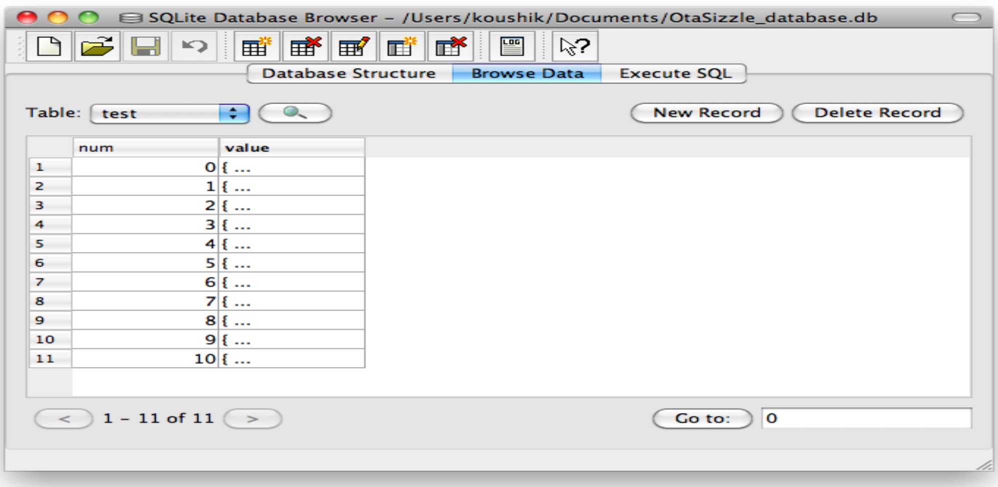
Location No Nearby Locations

Mylocation updated Successfully

Offline Capability –

Offline capability feature allows users to save the Wi-Fi scan windows when they are not connected to Internet and later send all the saved scan windows when ever user is connected to Internet.

SQlite is used for implementing this feature. SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.



SQLite Database Browser - /Users/koushik/Documents/OtaSizzle_database.db

Table: test

	num	value
1	0	{ ...
2	1	{ ...
3	2	{ ...
4	3	{ ...
5	4	{ ...
6	5	{ ...
7	6	{ ...
8	7	{ ...
9	8	{ ...
10	9	{ ...
11	10	{ ...