SLOC: System for Location-Based Services

Luís da Costa Carvalho
Telecommunications and Informatics Engineering
INESC ID, IST/UTL, Lisbon, Portugal
E-mail: luis.c.carvalho@tecnico.ulisboa.pt

***Abstract*—Currently, the location of people, animals and objects has become an important component of context information which allied with the exponential growth of mobile devices opens up the opportunity to create a wide range of value-added applications based on location of its users. Location systems are important and required so that, by positioning techniques such as Global Positioning System (GPS), Wireless fidelity (Wi-Fi) and *Bluetooth*, may provide location information to the user. This dissertation addresses the current state of the art of positioning techniques and location-based services, presenting the fundamental concepts and the most featured related works. It is also described the solution for the developed system (SLOC), as well as the respective evaluation. The SLOC system is a location system based on the proximity technique that uses positioning technology Wi-Fi and aims to provide a platform/*middleware* that supports a wide range of applications and location-based services. The SLOC system makes the separation between the applications and the positioning technology to simplify application development by programmers. To evaluate the effectiveness and efficiency of the system we developed a set of location-based applications.**

***Index Terms*—Location Systems, Mobile Devices, *Wi-Fi*, Location-Sensing Techniques, Applications.**

I. Introduction

Mobile computing has grown exponentially and currently holds a prominent place in everyday society. The variability of mobile devices and the lower prices that are practiced in the current technology market contributes to the exponential grown of the percentage of users of such devices (i.e. smartphones, tablets and laptops) relative to a recent past. Currently, programmers who develop applications and location-based services have at their disposal several location systems that use different technologies of positioning, such as GPS [1], Wi-Fi [2], *Bluetooth*[3], Radio-Frequency Identification (RFID)[1], Near Field Communication (NFC)[2] and Real-time Location System (RTLS)[3][4] to get location information of their users. This variety of location systems makes the programmer aware of the properties and base characteristics of each location system [4], [5], So that they can critically select the most suitable system for each specific application, optimizing its role as a developer.

Applications and location-based social networks are having a important growth and begin to emerge applications that offers benefits depending on the user location, such as promotions, discounts, recognition, social integration for being in a certain place or event. With the creation and rise of such applications, the uprising of malicious activity from some users who tried to cheat the system in order to achieve the benefits without being in person at the exact location. It is therefore essential to identify these users and ensure security in the interaction with applications. It is necessary some mechanisms that can verify the user's location accurately and in such a way that is not corruptible [6], [7], [8].

A. Motivation

The variety of contexts in which location-based services can be applied, such as health, entertainment and work in general, provide an advantage and extra motivation for the programmer to develop such services. The fact that these applications depend on the access to information about the location of people, animals or objects,it is important that they are supported by a type of system that has the ability to get that information through one or more positioning technologies. As stated earlier, there are various types of positioning technologies that use different techniques to obtain location data, so it is up to the programmer to choose the one that best fits the needs of the application that he wants to implement.

A location system that incorporates different positioning technologies has the advantage of providing location information acquired using different techniques, which are able to offer more options to develop applications that combine such different types of location data to represent the position more accurately and not limiting the creativity and objectives of the programmer.

B. Goals

The main goal of this work focuses on developing a location system (**SLOC**) that is compatible whit a wide variety of positioning technologies, such as GPS, Bluetooth and RFID, but initially it will be based on proximity techniques and implemented with the support of technology Wi-Fi. For this system, it will be necessary the implementation of a platform/middleware that will support a wide range of applications that provide location-based services.

[1] http://www.oosa.unvienna.org/pdf/icg/2010/VIC-Workshop/UN_Ho-We_Part_2.pdf

[2] http://compnetworking.about.com/od/wireless/tp/how-wifi-works_useful-facts-about-wireless.htm

[3] http://www.bluetooth.com/

[4] http://www.ekahau.com/real-time-location-system/technology

C. Requirements

The system **SLOC** must have the ability to obtain and store information about the user's location in a correct and consistent way, the location obtained can not suffer any changes or errors when stored in the system. The **SLOC** intends to offer a intuitive and easy to use Application Programming Interface (API) for developers of location-based applications. It is intended that the integration with the equipment that gives location support (i.e. Raspberry Pi with an Universal Serial Bus (USB) wireless adapter) has to be effective and efficient in order to give a quick and accurate response to requests for location. As for the security component, the system is designed to verify the identity of users through the user login.

II. Context and Related Work

There are a set of properties that characterize the location systems. These properties directly influence the behavior and performance of the mentioned systems[4], [9], [10].

A. Properties of location systems

When we intend to determine a location, we can choose between two groups of positioning techniques: active and passive.

Active - active positioning techniques determine the position through signs or beacons[5] sent by the devices and transmission infrastructures, these techniques are divided into two types:

1. Proximity: which consists in measuring the proximity to a set of points whose position is known. This type of technique is increasingly used by location systems based on RFID[11] and Quick Response Code (QR-Code)[6] technologies that acquire location information through tag and chips readers installed in specific and well-known location (the tags are attached / incorporated into an object and serve as identifier).

2. Triangulation: It is a technique that uses geometric properties and mathematical formulas for calculating the position. It can be done by *lateration*, which uses distance measures to a well-known points to obtain a location, or by *angulation*, which consists of measuring angles on points with well-known separations.

Passive - passive positioning techniques determine the position through sensors installed on the devices to obtain information of the surrounding environment. Such techniques are also divided into two kinds:

1. Scene analysis: examines a position through a well-known point by the system, as is the case of a system that use digital cameras to determine the location or calculates the location by analyzing the signal strength of two or more transmission equipment.

2. Movement tracking: calculates the position by tracking the speed and direction over a given time relative to a fixed position, or previously known. It can be used to calculate the location of an object passed a certain period of time relative to its initial position. This technique is also known as Dead Reckoning (DR)[12].

Physical and symbolic position - are two types of information provided by location systems. The physical position corresponds to an real position, such as the GPS coordinates, while the symbolic location involves abstract ideas of the position of something (e.g. in the room, in the car, at Instituto Superior Técnico (IST)).

Absolute and relative position - an absolute location system uses a shared reference grid for all located objects, two devices with GPS receptor lying at the same position have to report equivalent readings about their location. In a relative location system each object can have a set of reference points and these points are always relative to others known points of reference (e.g. John is on the left of the garden and on the right of the main road).

Accuracy and precision - accuracy refers to the margin of error of the location, while the precision indicates the frequency with which achieves the accuracy.

Location computation - in some location systems the device to be located is responsible for computing their own location. This method ensures privacy since other entities can not access your location, unless the programmer or user chooses to make it public. Other location systems need that the located device provides data periodically to external infrastructure in order to proceed to its location.

Scale - localization systems have the ability to locate objects in various scales, some can provide the position of a person, animal or object worldwide, within a metropolitan area, throughout a campus, in a building or even within within a division.

Recognition - there are some applications that are designed to perform specific operations according to the location of objects. These applications need to recognize or classify the located object. For such applications have the desired behavior, an automatic identification mechanism is needed.

Limitations - localization systems have limitations that affect its operation in certain environments and have direct influence on the type of applications that can be developed.

[5]Beacon is a data structure containing information about the communication network and is transmitted periodically to announce the presence of the wireless network

[6]Source: http://www.qrcode.com/en/history/

An example is the case of using GPS technology to implement an application in which the aim is to determine the position of objects indoors, inside a building where a receiver has difficulty in detecting signal transmitted by the satellite.

Cost - the cost of a location system is divided into several parts: time cost, space cost and capital cost.

B. Positioning Technology

The different properties in location systems allows you to perform a review and a critical choice of the system that best suits the needs and requirements of applications to be developed. Then the positioning technologies that support current applications and location-based services will be presented [13].

Location with service provider support - one of the methods of acquiring location information is using the architecture and communication infrastructure features implemented by service providers. The position information can be obtained through the signal from the telecommunications tower which provides signal to the mobile device [10].

Location by GNSS - Global Navigation Satellite System (GNSS)[7] systems are positioning systems that use satellites with global coverage and determine the coordinates of a terrestrial receiver as long as it is in the visual field of four satellites. To determine the coordinates of the receiver it only needs three satellites and the fourth is used to synchronize the time. The GPS positioning system is the best known GNSS system and most used by programmers to develop applications.

Location by short distance - this location method is mainly associated with location systems that use the proximity technique as a way to obtain information about the location of users. Applications using these types of systems allow the user to obtain information based on its location, in particular within buildings and reduced perimeter areas. Technologies such as Bluetooth, Wi-Fi [14], RFID and NFC are used in this location method.

C. Location systems

In order to address some of the flaws of some positioning systems [15], hybrid systems are used. These systems consist in combining the location information acquired by two or more positioning technologies and it is in this group of systems which operates the developed system (**SLOC**).

LifeMap[16] is a hybrid system which aims to determine in a practical and efficient way the exact location information of mobile users in indoor and outdoor environments. However, the use of accelerometer and digital compass gives rise to problems of power consumption and distortion caused by the direction of the user originating location errors.

LOC8[17] is a framework that aims to provide to the developers of applications information about the location of different positioning technologies. The LOC8 uses algorithms that aggregate the information collected to create a spatial model that recognizes the different location information formats. However, the system does not use symbolic location and the merge of different types of information can be an inconsistent result originating location errors.

MiddleWhere [18] is a platform/middleware that aims to separate applications based on location from positioning technologies. The MiddleWhere uses a probabilistic method to address the aggregation of location information of different positioning technologies. However, only has the ability to obtain location information of four technologies (i.e. GPS, RFID, Ubisense® tags and fingerprint recognizers).

BlueBroadcaster® [19] is a proximity marketing and advertising system for mobile terminals. The system uses the Bluetooth technology to send marketing and advertising messages and the Wi-Fi technology to make known their current services and promotions.

BLIPsystems® [20] allows you to check the flow of people and queues in real time through Wi-Fi and Bluetooth transmitters. The correct use of these two systems implies that the located devices must have the Wi-Fi and Bluetooth technologies. Using only these two technologies is a limitation relatively to the scope of action.

Google Location Services® also uses different types of position technologies to develop location based applications, using GPS, Global System for Mobile Communications (GSM) based in Cell ID (CID) developed by Google® and Wi-Fi technology to provide location information.

Google Location API®[21] aims to simplify the deployment of location-based applications removing the developer concern about underlying details of location technologies. However, the location services provided by Google® are limited in terms of positioning technologies and the API is specially designed for developing applications based on the Android OS® [8].

D. Applications and location-based services

The location-based services, are a class of services and computer programs intended to provide location information to the mobile user. The location information may be provided through some of the systems which have been mentioned above. There are two distinguished groups of location-based services [9], [10], [22]: *i) pull services*, provide the information directly requested by the user; *ii) push services*, provide relevant information for the user, but that has not been requested directly. Such applications have become increasingly important in everyday life of the general population being applied in various contexts, such

[7]Source: http://www.insidegnss.com/

[8]Source: http://developer.android.com/

as: games, location tracking, emergency services, navigation, social networking applications and are even used in industry in order to optimize job functions, such as applications developed by Ubisense® [9] that can make a real-time location and control of developed products in factories.

E. Location-based services privacy

The capture of location information is often related to the tracking of people or to the analyzing of their preferences and history, making the privacy of the location information of users a highly debatable topic. A study on the concern that users have in the privacy about their location information revealed that users have fewer privacy concerns as more useful is the application or location service [23]. Some systems have implemented privacy mechanisms that simplify the development of applications that require a high level of privacy, such as: Jano [24], which allows the configuration of security policies through Security Policy Language (SPL) [25], Confab [26] and pawS [27].

III. System Architecture

The **SLOC** system must have the ability to provide context information based on the location of people, animals or objects and in addition must have implemented a platform/middleware that provides support to a wide range of applications.

A. System architecture design

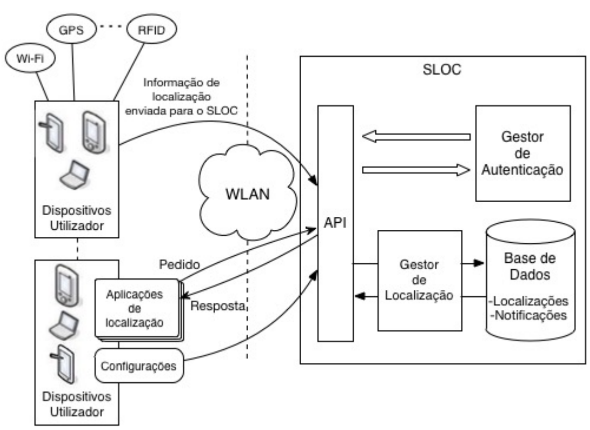

Fig. 1. SLOC Architecture

The **SLOC** system architecture is shown in Figure 1. The system architecture allows through a modular construction to add modules that are focused on a specific positioning technology (e.g. location based on Wi-Fi, Bluetooth, or RFID). Wherein each module performs the conversion of the respective type of location obtained by a positioning technology in a common language used in **SLOC**. Initially the system is developed based on the Wi-Fi module in which the location information about the user is obtained through Wi-Fi technology. Communication between **SLOC** and devices running different applications is accomplished through the Wireless Local Area Network (WLAN) access. The recognition of the geographical position is accomplished by proximity position technique and therefore provides the possibility of an association between the Access Point (AP)[10] (e.g. Service Set Identifier (SSID)[11]) identification and a physical or symbolic position. The interaction with the system is done through the exchange of messages between the user applications and the **SLOC** programming interface (i.e. API).

API SLOC - is the platform/ middleware that provides a set of functions that interact with the system. The offered functions are easy to use for the programmer, it will only have to ensure that applications are able to communicate with the system through the respective interaction functions.

The **SLOC** API offers two groups of services: *i)* queries and administrative functions and *ii)* to acquire location information. The first allows the interaction of user applications (e.g. location requests and notifications) and configuration functions (e.g. associating the identification of a AP to a location). The second allows to interact with positioning technology.

Authentication manager - this security component enables developers to implement an identification system in the users applications.

Location manager - the system's main functions are acquiring context information about the location and associate that information to users. The **SLOC** location manager is in charge of this functionality, serving as a communication element between the API and the system database.

SLOC database - have the function of store all system data. The data and all the location information generated by applications and by the system is stored in files in the **SLOC** file system. The types of information stored in the database are: *i)* associating an identifier to a location information provided by the positioning technology which may be physical or symbolic; *ii)* identification of users associated with a location identification along with a date and time; *iii)* equipment identification (e.g. AP) associated with one location identifier and *iv)* information about settings related to the applications.

B. Scenario of use

To better realize the features that **SLOC** offers, it is to consider the following scenarios:

[9] Source: http://www.ubisense.net/

[10] AP is a device used in wireless network that allows connection between mobile devices and access to the Internet or other networks

[11] SSID is a unique identifier that identifies the wireless network

1º scenario - the registration of the attendance of students in the classroom is a possible application based on location. Often the traditional method of attendance record it makes a unnecessarily miss of a few class minutes, automatic recording made by a simple checkin through an application or access to a web-based Graphical User Interface (GUI) will bring benefit to the teachers that can be able to devote more time to teaching the course material and to simplify the evaluation process of student attendance. Nowadays it is common that classrooms have Wi-Fi access, which allows students who are present in class to communicate with the **SLOC** through an application or web-based GUI, as mentioned above, to make the attendance registering a simple and intuitive process. The teacher can access all the information, since the **SLOC** stores attendance records.

2º scenario - the customer order registration of a restaurant through an application that communicates with the **SLOC** API. The application will allow customers to view the restaurant's menu, showing the various options available to the customer. When a customer request is made, the **SLOC** system receives a message with all data necessary for the processing of the request by the restaurant backoffice. The restaurant backoffice is another application that communicates with the **SLOC** API. This application will manage all the information related to applications, menus and clients (e.g. customer seat and table). The user application even has a login system to identify customers and that allows to offer discounts and promotions, according to the number of times the customer was present in the restaurant.

3º scenario -an application that can be developed with the support of the **SLOC**, is an application that allows the location of friends and notifies the user whenever a friend is near. The user can select a list of friends who can be notified when any of the friend of that list is near, also allows users to define which friends can be equally reported with its location.

IV. Implementation

The SLOC system is a location system with the support of Wi-Fi technology and for its development, a low-cost equipment was used,a Raspberry Pi[12] with the behavior of an AP through the usage of an USB wireless adapter[13], as presented Figure 2.

A. Raspberry Pi as Wi-Fi access point

In order for applications to be able to communicate with the SLOC system, was necessary to make the pass Raspberry Pi acting as an AP[28], [29]. For that we had to install the

[12]Source: https://www.raspberrypi.org/

[13]Source: http://www.edimax.com/

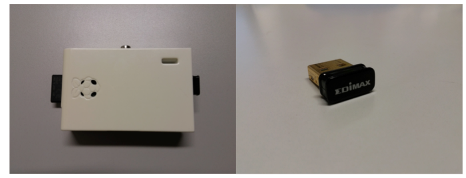

Fig. 2. Raspberry Pi and USB wireless adapter.

hostapd and *isc-dhcp-server* software. The *hostapd*[14] software is a *daemon*[15] which transforms the Raspberry Pi in an AP.

The *isc-dhcp-server*[16] is a *open source* software that implements a Dynamic Host Configuration Protocol (DHCP) server and is responsible for assigning Internet Protocol (IP) addresses to the user terminals that wish to communicate with the **SLOC** system.

The configuration of the file **/etc/dhcp/dhcpd.conf** allowed to create a local network and define the range of IP addresses that can be assigned. The server can assign addresses ranging from **192.168.2.2** to **192.168.2.254**, which gives a total of 253 terminals that can be connected simultaneously to the **SLOC** system. In order for the DHCP server be able to do the management of the network addresses in the wireless interface, the interfaces line in the file **/etc/default/isc-dhcp-server** was changed to **INTERFACES="wlan0"**, where **wlan0** is the network interface related to the USB wireless adapter..

The last step to complete the Raspberry Pi configuration as an AP was assigning a static IP to the USB wireless adapter. This setting allows applications and services to communicate with the **SLOC** system through the IP address **192.168.2.1**.

B. Raspberry Pi as web server

The **SLOC** system was developed based on the client-server[17] model that must have the ability to receive and respond to messages/requests made by user's applications and that are sent over the wireless network.

The Raspberry Pi must acquire the features of a web server[30] and for that the **lighttpd**[18] software was installed, as it is a fast and flexible web application server open source. We also used the **PHP**[19] for web and **API SLOC** development.

C. Time synchronization

The Raspberry Pi device does not have the RTC module as computers and laptops commonly have (to keep updated hours even when they are turned off). It is necessary that the Raspberry Pi is connected to the Internet in order to

[14]Source: https://w1.fi/hostapd/

[15]Daemon is a computer program running independently in the background independently

[16]Source: https://help.ubuntu.com/community/isc-dhcp-server

[17]Client-Server is a distributed application architecture that divides the client processes that send requests to the server, which in turn processes the information and sends a response

[18]Source: http://www.lighttpd.net/

[19]Source: http://php.net/

update the hours by Network Time Protocol (NTP)[20]. As it is intended that the machine only has the interface **wlan0** active and working as an AP, it was necessary to create a script (**wifi.sh**) that is executed when the Raspberry Pi starts and change the **wlan0** interface settings so at first can connect to a *time server* to update the time and then pass to function effectively as a AP.

D. SLOC API

The SLOC API was implemented based on **client-server** architecture and uses **PHP** as development language, as stated above. It works as a platform/middleware that receives Hypertext Transfer Protocol (HTTP) requests of location-based applications, performs the processing of requests and sends a HTTP response back to the user application.

The side of the user application to communicate with the **SLOC API** must create a HTTP request where the parameters required for the processing of the request are indicated in the Uniform Resource Locator (URL). It is necessary that the IP assigned to **SLOC** system (**192.168.2.1**), the filename **slocAPI.php** that implements the API and the parameters for the request are indicated in the URL of HTTP requests.

The **SLOC API** recognizes a set of intuitive and easy to use parameters, which are:

func - It serves to define the function that the user application wants to trigger in the **SLOC** system.
app - identifies the application that made the request.
path - indicates the path/location in the data structure, where the action will be made. There is also the **path1** and **path2** options that indicates two different paths if necessary.
filename - to indicate the name of a handled file.
id - identifies the user.
pass - user password.
data - the data that is sent to the **SLOC** (may be location information, or other information).
hour - the hour to assign to a user location.
min - the minutes to assign to a user location.

In the **SLOC API** are also implemented the functionalities of the login manager and location manager components of the **SLOC** system.

E. SLOC database

As the system must have the ability to store information generated by the interaction between the system and the applications, the solution was to keep all the information in files and folders in the web server root directory (*/home/pi/www*).

Each application has an associated folder as database structure that is well known by the respective application programmer. To make an HTTP request, the programmer must specify the path/location on the path parameter. The data are stored in form of text without restriction to the size of information, the programmer only have to take into account that the character **;** is the separator for different data types.

F. User application

Aiming to evaluate the **SLOC** system, it was developed a set of applications based on some scenarios described above that will be presented below.

Attendance registration in the class - the application for students attendance registration in the classroom is the continuation of *locDev* project [31] performed by students of IST and similarly to SLOC is also a near location system based on Wi-Fi technology. In the *locDev* system the classes attendance registration is done through interaction with the *PHP* page associated with the *web* server of the system where the students can access through a web browser.

In order to demonstrate compatibility of the **SLOC** system, besides being possible to integrate the *locDev PHP* page, it was developed an application based on the Android Operating System (OS)® with the same attendance registration features. So that the application can exchange HTTP messages with the **SLOC** system was used the library **HttpURLConnection**[32] in its implementation.

Tables and chart of class attendance - this application is also a continuation of the *locDev* work and aims to request information about the classes attendance registration to create charts and graphs of simplistic and intuitive reading according to data gathered by the **SLOC** system. It is an web application implemented with the use of HyperText Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript development languages, used of *Bootstrap*®[21] framework to implement the application User Interface (UI) and the *D3.js*®[22] library to draw the tables and the attendance chart.

The application can be accessed by the vast majority of web browsers installed in any OS, demonstrating the generality and great compatibility of the developed system. As the Android® application mentioned above, this application also has a login system so that only students who have created an account on the system can access information about class attendance.

Restaurant order registration - in order to demonstrate the applicability of the **SLOC** system two applications were developed based on the restaurant order registration scenario. Both applications were developed based on HTML, CSS and JavaScript web development languages and used the *Bootstrap*® framework. Applications are divided into a user application, which allows placing orders from the restaurant menu and an application to the restaurant backoffice to be used

[20]Source: https://wiki.debian.org/NTP

[21]Source: http: // getbootstrap.com
[22]Source: http://d3js.org/

by the staff that make the restaurant's menu management and the customer orders management.

The client-side application is accessed by reading a QR-Code strategically placed on restaurant tables which identify each customer seat. Through the client-side application the customer will be able to view the restaurant's menu and select the options he want, but initially the customer must do the register and the login to be identified by the **SLOC** system so he can take advantage of restaurant promotions and offers. When finished selecting options from the menu, the customer can view the entire order and the related amount through the application on the *Order Menu* option and then can validate the request.

The backoffice application is accessed by the restaurant staff through a web browser and is divided into restaurant menu management and customer order management. In menu management we can add and remove the options that are on the menu and displayed in the client application.

The order management displays orders that have already been requested by customers as well as verify which requests that have already been done (when the *Oder Show* button is green), requests which are being served (when the button *Show Order* is yellow) and requesst that are not done yet (when the *Show Order* is not changed color). In each order that is displayed in the order management menu, we can access another menu through the *Show Order* button that shows the options selected by the customer, the total amount of the order and where we can make the change of the request state.

V. System evaluation

To evaluate the system it was necessary to create a separate set of applications, so that it was possible to evaluate the **SLOC** compatibility and functionality.

Communication with the **SLOC** is performed with the aid of a Raspberry Pi with AP functionality, which gives data needed for evaluation.

The system evaluation metrics are the effectiveness and efficiency, response time, system scalability and the correct storage of the data.

A. Methodology and evaluation scenarios

The methodology and scenarios to evaluate the **SLOC** system focused on the simulation of multiple HTTP requests based on some of the requests made by the developed applications. The types of simulated requests to evaluate the system were as follows:

PA1 - request by the application of class attendance, to request confirmation of the presence in the classroom. The HTTP request uses the **POST** method and has the student number as the data to store in the **SLOC** system. The response of the request is **1** if the request has been processed correctly or a warning in case of error.

PA2 - request on the application of tables and graphic attendance in the classroom. The HTTP request is to request information about registering attendance in class. It used the **GET** method and its response its an attendance list related to one or more previously selected shifts and a list of students enrolled to the course.

PA3 - regarding the use of the application for orders registration in a restaurant and that is to simulate the loading of the application on the client mobile device when making the request of the restaurant's menu. It is an HTTP request using the **GET** method and receives back an JavaScript Object Notation (JSON) object with all the restaurant's menu options.

To analyze the response times of test requests, the *Postman*®[23] software was used and a laptop computer *MacBook Pro* with the processor *2,3 GHz Intel Core i5*, *4GB* Random Access Memory (RAM) memory, running the operating system *OS X Yosemite*. The application *Postman*® can be installed free of charge from the *Chrome Web Store*[24] and allows you to test web API by creating and sending multiple HTTP requests. After being made a connection between the laptop and the **SLOC** system (Raspberry Pi) through the wireless network with the SSID **sloc**, the *Postman*® application is used to get the system response times.

In order to evaluate the system in a real environment, tests were also carried out on a room using 13 mobile devices in order to test the system's ability to support connections and HTTP requests with different devices. The mobile devices used were 2 tablets, 7 smartphones and 5 laptops.

B. Processing time

To evaluate the system performance in terms of speed of response to requests made by applications, some tests were made to the system processing times, taking into account network latency. The response speed depends not only on the time that the system takes to process the request but also in the network latency, since the communication is performed through the wireless network. The latency test consists in measuring the time a network packet takes to travel from one point of the network to the destination.

To perform the latency test described above was used the **ping** command in the laptop terminal in order to analyze the times of Round-Trip Time (RTT) obtained. The test was to send 20 Internet Control Message Protocol (ICMP) packages to the **SLOC** IP destination (**192.168.2.1**) and has obtained the approximate value of 62ms as RTT average, which is the value used as network latency.

To obtain the system processing time was necessary to measure the time it takes to respond to requests made by

[23]Source: https://www.getpostman.com/

[24]Source: https://chrome.google.com/webstore/category/apps?hl=pt-PT

applications taking into account network latency, was then used the following formula:

$$T\_Processamento(n) = (avg(RTT(n)) - T\_Latencia) * n \quad (1)$$

T_Processamento(n) - the system processing time for **n** HTTP requests.
n - the number of HTTP requests.
avg(RTT(n)) - the RTT average of **n** requests.
T_Latencia - the latency value (62ms).

The Table 1 shows a test consisting in making 50 HTTP requests 50 for each test scenario **PA1**, **PA2** and **PA3**, in order to understand the influence of different scenarios with requests with different packet sizes in the system processing time.

TABLE I
SIMULATION OF 50 HTTP REQUEST MADE FOR EACH TEST SCENARIO

Scenarios	avg(RTT)	Packets size	T_Processamento
PA1	343 ms	180 Bytes	14.2 s
PA2	376 ms	3.6 KBytes	15.6 s
PA3	314 ms	364 Bytes	12.6 s

C. Scalability

An asset to the **SLOC** system is the ability to support and respond to a large number of user and maintain a efficient and constant working.

To test the scalability of the system was simulated multiple user requests in order to analyze the behavior of the system as the number of orders increases. It was also carried out a test in a room where a maximum of 13 devices simultaneously established connection with the **SLOC** system in order to evaluate the system in terms of the ability to communicate with different mobile devices.

With the values obtained by the formula described above, Figure 3 shows the difference of the **SLOC** processing time in relation to the number of HTTP requests performed.

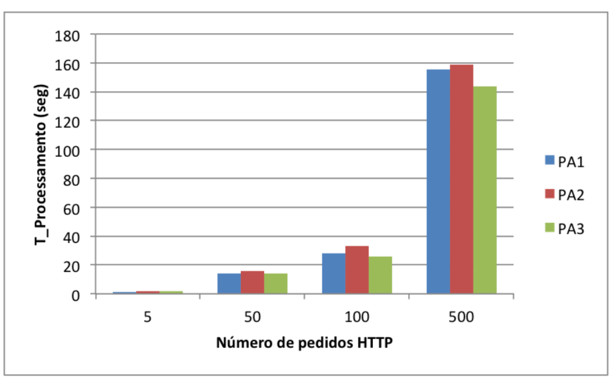

Fig. 3. Processing time of **SLOC** system.

Relatively to the test performed in the room with real users, in the first experiment only 5 devices were used to interact with the system. All that 5 were able to connect to the wireless network and make the request for attendance registration through the interface web using the attendance registration page based on *locDev* project mentioned above. In the second experiment the number of devices was increased (13 devices) to perform the same procedure as in the previous experiment and it was found that the result was the same, the connection to **SLOC** and the attendance registration has been successfully performed.

D. Correct storage of data

One of the most important elements in the **SLOC** system is the database component. It is therefore essential to test the correct storage of the input and output data, so that users requests are not answered with incorrect information.

In order to evaluate the performance and integration of the system database, the following tests were performed:

Test *i*: 13 attendance registration requests were made by web interface and Android® application according to the test carried out in the room, using different mobile devices. In the verification of the file where the data was stored it was found that the storage of data in the system was successfully performed without data loss.

Test *ii*: based on the same number of requests as the previous scenario, but using the restaurant orders registration application for test data storage and the application of restaurant management to verify the correct storage of orders placed, we verified as in the first test that the data storage in the system was successfully performed.

E. Results and evaluation

With the analysis of the results obtained from the tests performed to the **SLOC** system was concluded that the latency time (62 ms) is a good value. In terms of system processing time has been observed that there is no significant difference between the response times of requests with packets of different sizes. With the 50 HTTP requests test performed in accordance with the scenarios **PA1**, **PA2** and **PA3**, we also found that the average processing time is approximately 14 seconds. The requests number increasing the number made the processing time also increased, so the greater the number of users that make simultaneous requests, the higher is the system's processing time and increases the response time of the user requests.

Regarding the scalability test, we found that the system is able to address a large number of HTTP requests and that the processing time was proportional to the number of requests. The tests carried out with real users it was found that the system has the ability to communicate with different devices without failure or loss of information.

The tests performed to assess the correct storage of the data show that the data was properly stored in the system without any changes or losses.

VI. Conclusions

In the system development we used a low-cost equipment (i.e. Raspberry Pi with USB wireless adapter) and the positioning technology Wi-Fi for context information about the location. The SLOC aims to simplify the development of applications based on location, offering a API easy to use by the programmer. Communication between the SLOC and location applications is done through the wireless network created by the system with the SSID **sloc** that identifies the network in which users have to connect to exchange messages. The messages exchanged between applications and the system are HTTP requests and responses that are processed by **SLOC API** that runs on the web server installed in the system.

In order to make an evaluation of the developed system, a set of location-based applications has been developed. The implementation of applications took into account different scenarios of usage and different development languages (*PHP*, HTML, CSS, *JavaScript* and *Java* in the implementation of Android® application). It was also created a set of multiple HTTP requests simulation tests based on the applications developed and real environment tests in order to assess system performance in the interaction with different devices.

With the assessment made and the different applications developed, it was found that the **SLOC** system is compatible and allows the implementation of a wide range of location based applications for different devices with different OS's. In general, the system had a good performance in the evaluation that was conducted, however it was observed that can be made some significant improvements to the level of scalability and security.

According to the requirements and objectives presented in I, the following contributions were made:

- Development of a location system with the support of Wi-Fi technology, where we used a low-cost equipment (Raspberry Pi and an USB wireless adapter) that in general showed a good performance;
- Implementation of a platform/middeware serving as communication channel and API easy to use by the programmer;
- Development of a set of applications to test and evaluate the developed system.

VII. Future Work

The system developed is supported by Wi-Fi positioning technology, but with a view to future work, it should have the ability to get location information from other positioning technologies. To make this possible, it is necessary to change the API services group, so that the system can receive and manipulate the new type of location information.

Regarding security, it would be interesting the possibility of integrating an authentication system in the *SLOC* system in order to verify the authenticity of its users. Also, in terms of security, it is important that communication between the *SLOC* system and user applications were made through a secure channel in which the information exchanged between them were not acquired/revealed by malicious third parties.

Another limitation of the **SLOC** system is the storage of data in files. This method of storing the information exchanged between the system and location-based applications in the secondary memory of the equipment used for the development of the **SLOC** system (Raspberry Pi using a 16GB memory card) - dependent on the storage space availability of the equipment memory, might bring scalability problem, since the more users interact with the system, more information is stored in files in the secondary memory.

References

[1] P. Fuhrer, D. Guinard, and O. Liechti, "RFID : From Concepts to Concrete Implementation Table of Contents," *IPSI-2006 MARBELLA, February 10-13, 2006, Marbella, Spain*, 2006.

[2] S. Burkard, "Near Field Communication in Smartphones," *Dep. of Telecommunication Systems, Service-centric Networking Berlin Institute of Technology, Germany*. [Online]. Available: http://www.snet.tu-berlin.de/fileadmin/fg220/courses/WS1112/snet-project/nfc-in-smartphones_burkard.pdf

[3] H. Kim and J. Song, "Real-time u-LBS system for mobile vehicle in 3G networks," *Proceedings of the International Conference on Mobile Technology, Applications, and Systems, ACM*, 2008.

[4] J. Hightower and G. Borriello, "Location Systems for Ubiquitous Computing," *Computer, IEEE*, no. August, 2001. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=940014

[5] S. Dobson, "Leveraging the subtleties of location," *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies, ACM*, no. october, pp. 189–193, 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1107597

[6] S. Saroiu and A. Wolman, "Enabling new mobile applications with location proofs," *Proceedings of the 10th workshop on Mobile Computing Systems and Applications - HotMobile '09*, pp. 1–6, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1514411.1514414

[7] X. Ni, J. Luo, B. Zhang, J. Teng, and X. Bai, "MPSL: A Mobile Phone-Based Physical-Social Location Verification System," *Wireless Algorithms, Systems, and Applications, 7th International Conference, WASA 2012, Yellow Mountains, China*. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-31869-6_42

[8] B. Carbunar and R. Potharaju, "You unlocked the Mt. Everest badge on foursquare! Countering location fraud in Geosocial Networks," *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, pp. 182–190, Oct. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6502516

[9] S. Steiniger, M. Neun, and A. Edwardes, "Foundations of location based services," *Lecture Notes on LBS*, pp. 1–28, 2006. [Online]. Available: http://ftp.jaist.ac.jp/pub/sourceforge/j/project/ju/jump-pilot/w_other_freegis_documents/articles/lbs_lecturenotes_steinigeretal2006.pdf

[10] M. Mohammadi and A. Naserasadi, "A Survey on Location based Services and Positioning Techniques." *International Journal of Computer Applications (0975 – 8887)*, vol. 24, no. 5, pp. 1–5, 2011. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=09758887&AN=74099376&h=tNQwhCqLNXcRaWrKh3WYefkOcPciNbmOurz%2BaS84mN6dh7Bct5ujTbJmuMm%2FiBZ3e1huEm2LaUG4RZpKgG8I0w%3D%3D&crl=c

[11] G. Gomes, "Sistema RFID com tecnologia ZigBee," Ph.D. dissertation, 2008.

[12] S. Beauregard and H. Haas, "Pedestrian Dead Reckoning : A Basis for Personal Positioning," *PROCEEDINGS OF THE 3rd WORKSHOP ON POSITIONING, NAVIGATION AND COMMUNICATION*, pp. 27–36, 2006. [Online]. Available: http://www.wpnc.net/fileadmin/WPNC06/Proceedings/33_Pedestrian_Dead_Reckoning.pdf

[13] J. Lourenço, "A moda dos serviços baseados na localização do utilizador," 2013. [Online]. Available: http://pplware.sapo.pt/tutoriais/networking/a-moda-dos-servicos-baseados-na-localizacao-do-utilizador/

[14] L. Zhang, C. Tao, and G. Yang, "Wireless positioning: fundamentals, systems and state of the art signal processing techniques," *InTech, Rijeka, Croatia*, 2011. [Online]. Available: http://www.intechopen.com/source/pdfs/14749/InTech-Wireless_positioning_fundamentals_systems_and_state_of_the_art_signal_processing_techniques.pdf

[15] J. Han, J. Lee, T. T. Kwon, D. Jo, T. Ha, and Y. Choi, "How to Mitigate Signal Dragging during Wardriving," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 20–27, Jan. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5370803

[16] J. Chon and H. Cha, "Lifemap: A smartphone-based context provider for location-based services," *Pervasive Computing, IEEE*, pp. 58–67, 2011. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5686873

[17] G. Stevenson, S. Dobson, and P. Nixon, "LOC8: A Location Model and Extensible Framework for Programming with Location," *Pervasive Computing, IEEE*, pp. 28–37, 2010.

[18] A. Ranganathan and J. Al-Muhtadi, "Middlewhere: a middleware for location awareness in ubiquitous computing applications," *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1045686

[19] BlueBroadcaster, "Proximity Marketing," 2014. [Online]. Available: http://www.bluebroadcaster.com/

[20] BLIPsystems, "BlipTrack," 2014. [Online]. Available: http://www.bliptrack.com/

[21] Android-Developer, "Location & Places," 2014. [Online]. Available: https://developer.android.com/google/play-services/location.html#

[22] V. Chitre, D. Amritansh, and D. Pandey, "Trendsetter: Location Service using Android," *Department of Computer Engineering, Bharati Vidyapeeth College of Engineering Navi Mumbai-400614*, vol. 2, no. 4, pp. 431–436, 2013. [Online]. Available: http://www.advanceresearchlibrary.com/temp/downloads/ijair/april2013/t81.pdf

[23] L. Barkuus, A. Dey, and L. Barkhuus, "Location-Based Services for Mobile Telephony: a Study of Users' Privacy Concerns," *Proceedings of the INTERACT 2003, 9TH IFIP TC13 International Conference on Human-Computer Interaction*, 2003.

[24] J. Simão, C. Ribeiro, P. Ferreira, and L. Veiga, "Jano: specification and enforcement of location privacy in mobile and pervasive environments," *Proceedings of the 2nd International Workshop on Middleware for Pervasive Mobile and Embedded Computing, ACM*, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1890793

[25] C. Ribeiro, A. Zuquete, P. Ferreira, and P. Guedes, "SPL: An Access Control Language for Security Policies and Complex Constraints." *NDSS Symposium*, 2001. [Online]. Available: http://www.researchgate.net/publication/221655503_SPL_An_Access_Control_Language_for_Security_Policies_and_Complex_Constraints/file/e0b49526675377122d.pdf

[26] J. Hong and J. Landay, "An architecture for privacy-sensitive ubiquitous computing," *Proceedings of the 2nd international conference on Mobile systems, applications, and services, ACM*, p. 177, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?doid=990064.990087http://dl.acm.org/citation.cfm?id=990087

[27] M. Langheinrich, "A privacy awareness system for ubiquitous computing environments," *UbiComp 2002: Ubiquitous Computing*, 2002. [Online]. Available: http://link.springer.com/chapter/10.1007/3-540-45809-3_19

[28] R. P. Headquarters, "How-To : Turn a Raspberry Pi into a WiFi router," 2014. [Online]. Available: http://raspberrypihq.com/how-to-turn-a-raspberry-pi-into-a-wifi-router/

[29] G. Sims, "How to Set up a Raspberry Pi as a Wireless Access Point," 2014. [Online]. Available: http://www.maketecheasier.com/set-up-raspberry-pi-as-wireless-access-point/

[30] Matt, "How To Setup A Web Server On Your Raspberry Pi," 2013. [Online]. Available: http://www.raspberrypi-spy.co.uk/2013/06/how-to-setup-a-web-server-on-your-raspberry-pi/

[31] J. a. Amaral and R. Bruno, "locDev Client Presence Detection," *Mobile Computing, IST(Instituto Superior Técnico)*, 2013.

[32] Android-Developer, "HttpURLConnection," 2015. [Online]. Available: http://developer.android.com/reference/java/net/HttpURLConnection.html