Offline Indoor Localization leveraging Human Perception of Textual Signs

Abhishek Kejriwal
Graduate Student
Indiana University
107 S Indiana Avenue, Bloomington
Ph. no.- +1-812-369-5807
abhikejr@umail.iu.edu

Shrijit Pillai
Graduate Student
Indiana University
107 S Indiana Avenue, Bloomington
Ph. no.- +1-812-650-2645
pillaish@umail.iu.edu

Vimalendu Shekhar
Graduate Student
Indiana University
107 S Indiana Avenue, Bloomington
Ph. no.- +1-979-676-7082
vshekhar@umail.iu.edu

ABSTRACT
Localization is one of the widely studied areas in mobile computing. There are many systems which solve the localization problem using acoustic signals, visible light, radio signals and using other techniques. Most of these systems require dedicated hardware or infrastructure support. In this paper, we propose an offline software-based approach for facilitating indoor localization that leverages the human perception of textual signs inside a building by storing these signs in a textual format in the database. The key motivations behind this paper are the vast application area of indoor localization and the inability of the other related systems to handle this problem without the need of extra hardware and complex algorithms. Our approach requires no extra hardware or any infrastructural support. This application requires no advanced features and can work on any off-the-shelf smartphone. The key insight behind this work is that every building is equipped with a large number of textual signs to guide the visitors. The knowledge of the location of these signs inside the building and the sign nearest to the visitor can actually pinpoint the location of the visitor on the floor map. The approach is completely offline and does not require Bluetooth, cellular, Internet or Wi-Fi network. As there is no infrastructure support required, the system is energy-efficient. The main challenge here is to determine the desired location if there are identical signs at different locations. We explore the feasibility of this approach by designing an android application that can be installed on any off-the-shelf android tablets or mobile phones. We demonstrate the effectiveness of our approach by running our application on on new devices such as the Google Nexus tablet, Moto X and Moto G smartphones and also on relatively old devices such as the Samsung Galaxy S2. Results of evaluation obtained by testing our application inside one of the buildings in the university campus show that the app correctly identified all the locations nearby the textual sign with 100% accuracy.

Conference'04, Month 1–2, 2004, City, State, Country.

Categories and Subject Descriptors
H.5.0 [Information Interfaces and Presentation]: General

General Terms
Design, Experimentation

Keywords
Offline, Indoor, Localization

1. INTRODUCTION
Today, it is impossible to get lost thanks to the GPS, cellular, Internet and the Wi-Fi. Due to the infrastructure support, it is pretty easy to determine one's location. But take the infrastructure out of the picture and the seemingly easy task of identifying one's location becomes difficult. Such a scenario might arise when a person is in a large building without any access to the Internet, 3G, GPS or the Wi-Fi and would like to find his/her location in the building. In such a scenario, the nearby textual signs would be of great help in determining the location of the user. Our offline indoor localization system exploits the use of these textual signs to determine the user's location without the need for any infrastructure support or dedicated hardware. The system performs offline indoor localization by maintaining a database of the textual signs and associating these signs with their corresponding co-ordinates. The signs are stored in the textual format thereby consuming less space. As each sign is mapped to a co-ordinate, once the user identifies the textual sign in the system, the location of the user can be determined with very high accuracy.

The paper is organized as follows: Section 2 discusses the related work in the field of localization using the different techniques and compares our system with them. Section 3 describes the system design, the rationale behind the design, the challenges faced and their resolutions. Sections 4 and 5 mention the implementation and the evaluation of the system respectively. Section 6 discusses about the limitations, other possible solutions and future work associated with the application. The paper concludes with Section 7 which is the conclusion.

2. RELATED WORK

Localization systems using different techniques have been proposed recently. CAPS (Cell-Id Aided Positioning System) [7] leverages the mobility and the position history of the user rather than relying on GPS and cellular-tower based methods to improve the accuracy and the energy efficiency of the system. However, it would not be helpful if the user is indoor and would like to determine the location at a finer granularity. Our system is designed to find the location in an indoor environment with high accuracy. Peer assisted localization approach was proposed in [2] using acoustic ranging estimation among peer phones which then maps their locations against a Wi-Fi signature map. This approach requires that peers accept the signal transmitted by the requesting phone. This might not be possible if there are no peers nearby or the peers are far away or if the request is not accepted by the peers. Our system is dependent only on the textual signs which are present at almost all locations in an indoor environment. Localization using Acoustic Background Spectrum was proposed in [3] in which the current acoustic fingerprint of a room was compared to its previous fingerprint stored in the database to determine the location. However, there are can be scenarios wherein the acoustic fingerprint of the location might change over time or multiple places might have the same fingerprint. In such cases, the technique would not be useful. Our system being based on textual signs will not encounter such scenarios. S. Nirjon et. al[5] proposed localization by receiving GPS signal in an indoor environment. However, the technique cannot be implemented using off-the-shelf devices due to the dedicated hardware requirement. In contrast, the indoor localization using our system can be implemented on off-the-shelf smartphones running on Android and does not require any hardware component for localization. The localization technique in [6] uses image for localization. The technique requires the storage of images which makes the database size large and also requires cloud for the computation. On the contrary, the technique proposed by our paper does not require any infrastructure support and as signs are stored in textual format, the database is compact. The paper [11] makes use of GSM signal for indoor localization. The technique however cannot be applied on off-the-shelf devices due to certain requirements. Our solution can be applied on any off-the-shelf device and is completely offline.

3. DESIGN

The main components of this application are the User Interface Module, Database Module, Co-Ordinate Shifting Module and the Location Render Module. Figure 1 shows the high level design of the system. The Database module

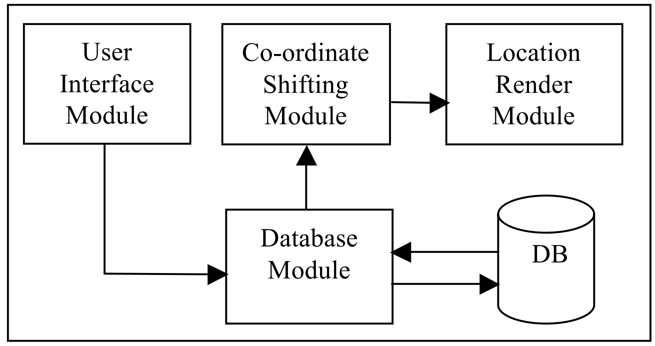

Figure 1. High-level System Design

consists of a table which stores the textual signs available to the user. Each of these textual signs is mapped with their corresponding locations in the form of co-ordinates. The User Interface module presents the user with options to enter the textual signs from a pool of signs for a particular floor. The sign selected by the user is then sent as a query to the Database module which returns the location to the Co-Ordinate Shifting Module. The Co-Ordinate Shifting Module on receiving the co-ordinates, shifts the co-ordinates according to screen size of the device and transfers the final co-ordinates to the Location Render Module. The Location Render Module finally generates a colored dot at the desired location on the screen.

3.1 User Interface Module

As simple as the proposed solution looks, architecting and designing the interface with simplistic and user friendly input mechanisms was one of the most challenging parts of the application. One of the greatest challenges was to come up with a solution to the problem of allowing the user to type in the textual sign. Problems involved with letting the user to enter the textual sign manually are as follows:-.

i. The user may not enter the sign correctly or completely. For example – Herman Wells Lab108 may be entered as Wells Lab108, thereby causing no results to be fetched from the database.

ii. The user may enter a sign that is not present in the Database.

iii. Presence of the sign in more than one location on the same floor will create ambiguity and confusion.

To address the above mentioned problems, we eliminated the idea of allowing the user to manually enter the textual sign. Instead, now we make the user select a textual sign from a list of preloaded signs. Android provides us with three different ways of making a list of any type using data- Spinner, List View and AutoComplete Text View.

3.1.1 Solution 1:

Using Spinner and ListView: In this solution we asked the user to select a building in which he was present. Once the user selected the building, then he was prompted to select the floor in which he was located. As soon as the user selects the floor, all the corresponding signs related to that floor are populated in the spinner/listview. Now the user can view the entire list and select the nearest sign thereby getting the appropriate location. But, this solution has its own limitation. Once the entire list of signs is populated, the user has to go through the trouble of searching the desired textual sign in the entire list. In case of a long list, this solution is inefficient as the user has to search through the entire list to find the sign of his interest.

3.1.2 Solution 2:

Using AutoCompleteTextView: To counter the problem introduced by solution 1 we incorporated the use of AutoCompleteTextView and removed Spinner/Listview.

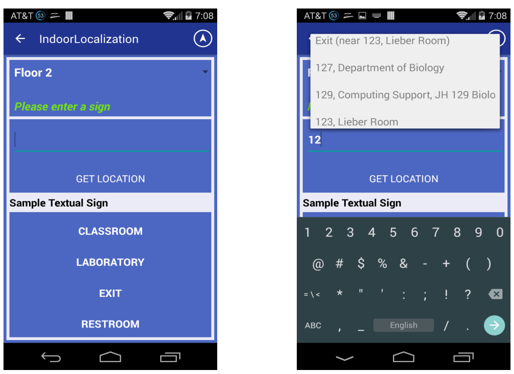

Figure 2. The figure shows two different states of the app. In the left figure, the user is asked to enter a textual sign. In the right figure, as soon as the user starts typing a list of all the signs show up and user has to select a sign from the list.

An important feature of AutoCompleteTextView is that once a user types in two letters of any word, then the AutoCompleteTextView automatically populates a list of all the entries that have those two letters anywhere in the word. Now the user has to choose only from a handful of textual signs rather than selecting from an entire list. It increases the search efficiency to a great extent and is more users friendly. Moreover following this practice restricts the user from entering a wrong textual sign. This also solves the issue of having same signs at multiple locations on the floor as the user would now select only one sign.

The use of AutoCompleteTextView raises a very important question, as to how does the user decide which textual signs to look for in the surroundings and how to enter them. As an answer to this question, once the user selects the floor on which he is located, a section known as Sample Signs is made available to the user. In this section, images of corresponding textual signs are stored. Here the images are classified into categories. For example, Classroom is a category. Once the user clicks on the Classroom, he is shown the image as in Figure 3.

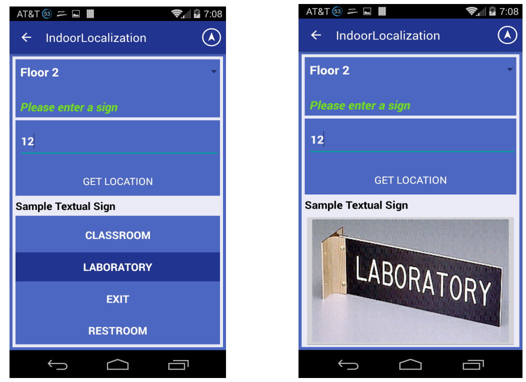

Figure 3. The section sample textual sign guides the user to decide what type of sign to enter in the text box

Once the user clicks on Classroom, he is shown an image with a sample textual sign which self demonstrates, as to how to enter any sign related to a Classroom.

3.2 Database Module

We have used the SQLite Database for our application as it comes embedded in every android device and does not need any setup procedure or administration of the database. SQLite is used instead of content developer as the latter is mainly used to store data that is shared between many applications. As the user is only allowed to query the existing signs, the database is created only once when the application is started for the first time.

Table 1. Table structure for storing the signs and co-ordinates

Building Name	Floor Number	Prev Sign	Current Sign	Next Sign	Location
Jordan Hall	1	Exit	123, Lieber Room	110	804,731
Jordan	1	123, Lieber Room	110	113	904,391

As shown in Table 1, the table contains columns for the name of the building and the floor in which the sign is present, the text of the previous sign, current sign, next sign and the co-ordinates for the current sign. The location of the current sign is entered in the form of x and y components that corresponds to the x and y co-ordinates of the point generated on the screen.

Previous and Next signs will be required for implementing navigation in the current system. We have kept navigation as future work. The database is queried for fetching the number of floors in the building, for fetching the sign on the floor selected by the user and then for retrieving the co-ordinates of the sign selected by the user.

3.3 Co-ordinate Shifting Module

The floor map will be rendered based on the screen size of the device. Hence the position of the desired location on one device will not be the same on another device if the two devices have different screen sizes as shown in Figure 4. This problem is resolved by the Co-Ordinate Shifting Module. The co-ordinates of the locations are stored in the database based on the screen size of the reference device. When the co-ordinates of the desired location are fetched from the database, the module determines the screen size of the device and shifts the co-ordinates to the correct position based on the scale factor of the width and height of the device.

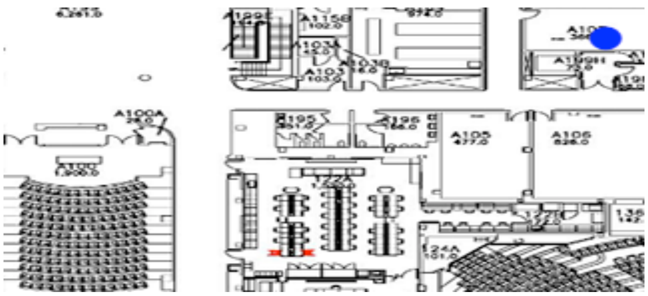

Figure 4.a The location of classroom in the map as seen on Google Nexus tablet

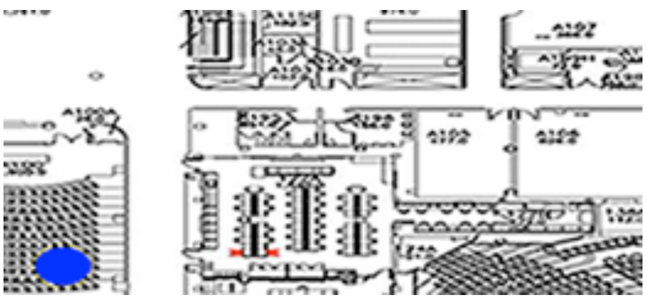

Figure 4.b The location of classroom in the map as seen on MotoX

Let W_n and H_n be the width and height of the device of interest and W_r and H_r be the width and height of the reference device. Also, let X_n, Y_n, X_r and Y_r be the X and Y co-ordinates of the location on the concerned device and the reference device respectively.

Let S_w and S_h be the scale factors for the width and height required to scale the co-ordinates of the location to the correct position.

$$S_w = W_n / W_r \quad \ldots \quad (1)$$

$$S_h = H_n / H_r \quad \ldots \quad (2)$$

For devices with screen size greater than the reference device, the scale factors for the width and height will have a value greater than 1 else the value will be less than 1. For devices with the same screen size as the reference device the scale factors for the width and height will be equal to 1. The new X and Y co-ordinates, X_n and Y_n on the concerned device is given by the following equation:-

$$X_n = S_w * X_r$$

$$Y_n = S_h * Y_r$$

X_n and Y_n represent the co-ordinates on the non-reference device which are shifted based on the scale factors corresponding to the width and height of the device.

3.4 Location Render Module

This module accepts the new shifted co-ordinates from the co-ordinate shifting module and draws a dot at the desired location.

4. IMPLEMENTATION

The application is implemented on MotoX and Nexus 7 running on Android 4.4.4 KitKat and 5.0.2 Lollipop respectively. MotoX is taken as the reference device for determining the desired location on devices with different screen sizes.

The data required for this application was collected manually by going through each and every floor of the building and noting down each sign and its location based on the screen size of the reference device. The signs mainly included Classroom, Laboratory, Restrooms, Library, Stairs, Entry/Exit, and intermediate signs in between. To fetch the exact co-ordinates of each textual sign a unique feature of Android system was utilized. It is known as the pointer location. After the pointer location is enabled, the user can get the X and Y co-ordinates of any point on the screen by simply touching and holding it. As shown in Figure 3, first we uploaded the floor map in the application and then mapped the X and Y co-ordinates corresponding to each sign into the database.

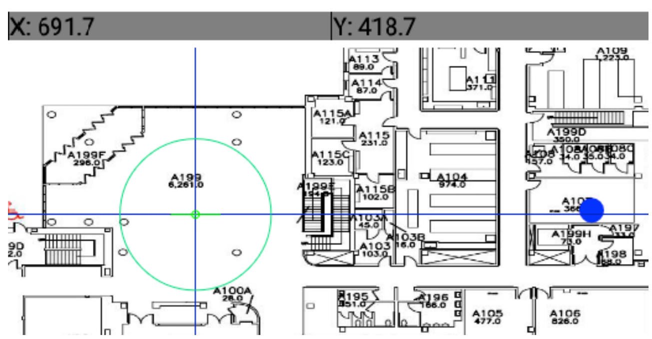

Figure 5. Pointer Location showing the X and Y co-ordinates on the reference device.

The data is maintained in a linear data structure. Data is then retrieved from this data structure and inserted in the database when the application loads for the first time. Instead of using an array for maintaining the data, we have created a hash map for each floor of every building that contains the name of the signs on that floor as the key and location of the sign as the corresponding value. We preferred the hash map as it enables a constant retrieval time. An adapter is created that identifies the incoming queries, performs the execution of the queries and returns the co-ordinates which will be passed to the Co-ordinate Shifting module. The Co-Ordinate Shifting module determines the scale factors using equations (1) and (2) and passes the new co-ordinates to the Location Render module which renders the co-ordinates as a dot on the screen. To improve the visibility of the dot and the nearby locations on the floor map, the pinch zoom and the pan feature has also been implemented.

5. EVALUATION

5.1 Size of Application

The size of the application will increase as we add new buildings under the purview of this application. This is because storing the textual signs in the buildings will increase the size of the database. It is found that with 103 signs, the size of the database is 48 kb, and increases by 4kb for every new 120 signs added as shown in Figure 6.

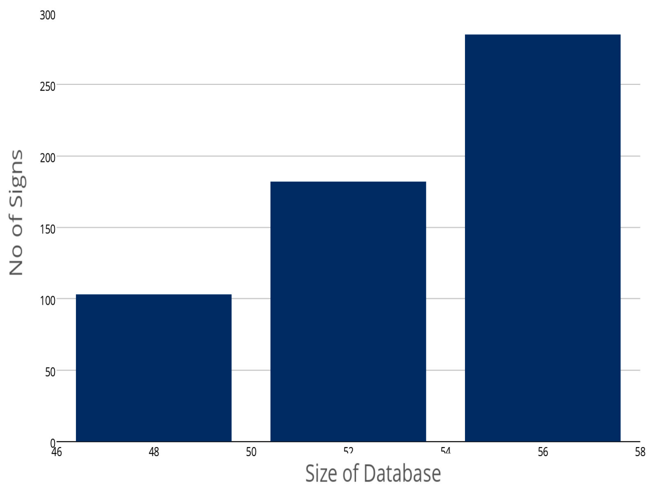

Figure 6. Bar graph showing how the database size increases with an increase in the number of signs

We were able to determine this proportionality by temporarily disabling the primary key of our table and inserting duplicate signs into the table. We added 30 signs at a time and then checked the size of the database to determine the increase.

5.2 Accuracy

The application was evaluated on smartphones with different screen sizes and configurations. We used the Motorola MotoX running on Android 4.4.4 Kitkat as the reference device. The co-ordinates of the locations stored in the database is with respect to the screen size of the reference device. The application was evaluated on the reference device as well as on other devices such as the Google Nexus tablet, Motorola Moto G and also on Samsung Galaxy S2. The application was tested on two floors of a building in the university campus. Figure 7 shows a graph of the number of locations on Floor 2 and Floor 6 of the building pinpointed by the application within a radius of 3 meters on the reference device. We visited each floor and for the available textual signs, checked for our location in the application. The results of our evaluation are summarized in Figure 7.

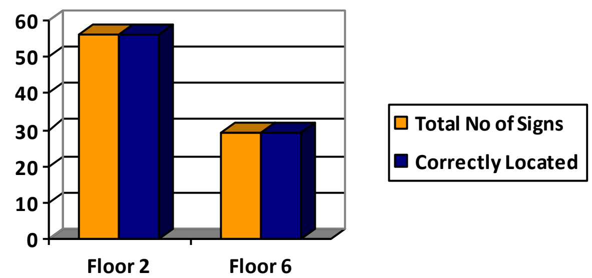

Figure 7. Graph showing the accuracy of the app

The graph shows that in Floor 2, there were a total of 56 signs, out of which the application was able to locate all 56 of them correctly. Out of a total of 29 signs in Floor 6, the application correctly located all of them. Thus for the two floors, the accuracy of the application is 100%. The results are not surprising as the co-ordinates of the textual signs were manually entered into the database. Thus as long as the co-ordinates are correctly entered into the database, the application will always pinpoint the correct user location.

6. DISCUSSION

In this section we discuss about the limitations of our current system and potential directions for future work.

6.1 Overheads

6.1.1 Memory Overhead

The signs inside the building and their co-ordinates are stored in a textual format in the database that is created when the application is loaded for the first time. As data from new buildings are added to our application, the size of this database will increase and may take up several MBs of memory on the phone. One potential solution to this problem could be to ask the user to select the buildings that he frequently visits or the most prominent buildings in the neighborhood and only insert data for those buildings in the database and erase data for all the other buildings. The other solution could be to upload the data on a cloud accessible to the users so that they can download the data according to their need. Both these solutions are potential future work for this system.

6.1.2 Time Overhead

Large size of the database can increase the time required to fetch the query results. This problem can be solved by indexing the appropriate columns and maintaining efficient primary and foreign key relations.

Another possible solution to reduce the time complexity would be to compress the data related to buildings the user hardly visits. The frequency of visit can be determined based on the number of times the user has selected the building in the application. Then using the Least Recently Used (LRU) algorithm, the data for the corresponding buildings can be compressed. This solution would also require a decompression module in case the user selects the building for which the data is compressed.

6.1.3 No Signs Nearby

There may be a situation when the user is in a section of the building where there are no signs like a long corridor. In this case our application will fail to identify the location of the user. This problem could be solved by identifying and marking such sections of the building beforehand and guiding the user to the nearest sign. Time constraint has prevented us from implementing this solution in our present system, so we identify it as future work.

6.1.4 *Entered Sign not present in the database:* In rare cases, the sign entered by the user may not be present in the system. In this case also, our system will fail to identify the location of the user. This situation can be avoided by meticulously carrying out the data collection process, making sure that no sign is missed. Even after thorough data collection, if this situation occurs the user can be asked to enter the next nearest sign.

6.2 Other Solutions

The other possible solutions apart from using textual sign for offline indoor localization are as follows:-

6.2.1 Use images instead of textual signs

Instead of creating a database of textual signs, one solution could be to create a database of images. These images would show the user his current location. But the main disadvantage of this system is that the database would consume a lot of memory of the device. Also, in case of identical images on the same floor, retrieval of the location co-ordinates might become difficult.

6.2.2 Use of QR Code/NFC Card

QR Code/NFC Card is a good possible solution as compared to textual signs. The use of QR Code would reduce the work on the part of the developer of manually adding the location co-ordinates into the database. But QR Code has lots of disadvantages. The solution requires you to stick QR Codes on doors and walls of the buildings for which permission would be required. QR Code is only 30% resistant to scratch. So if the QR Code is destroyed then the user may end up getting false results. Moreover in case of educational institutes such as school, children may take out the QR Code and stick it somewhere else. This again would result in displaying the wrong location. From the user's perspective, the user will have to be very patient for scanning the QR Code with a mobile camera. He may also end up scanning the QR Code in the 4^{th} or 5^{th} attempt thereby making it frustrating for the user. The process consumes much more battery as compared to our solution as camera consumes 20% of the battery power. If QR Codes are stuck anywhere on the upper half of the door, scanning them may be difficult for people with disabilities, such as people on wheelchairs. Also, the QR code/NFC reader might not be available on all smartphones.

6.3 Future Work

Apart from the ideas to overcome various limitations of the system discussed in the above section, the most intuitive extension to the current system would be navigation. The user can enter the sign nearest to him and the destination sign like a room number or an elevator sign and the system can return a set of signs for the user to follow to reach the destination. The path could be traced out on the floor map to guide the user. This feature would require us to implement the shortest path algorithm in order to find the shortest path from the source to the destination. Using navigation user can locate the nearest exits, restrooms, administrative offices inside the building and classrooms.

A second possible future work would be to incorporate computer vision in the system. This would allow the user to capture images of signs and an image processing unit can process the images to extract the textual sign and store the sign in the database. This would eliminate the need to manually enter the signs in the database. The overhead involved in the approach would be the computation required to extract the text from the image.

Another possible future work would be using a speech recognition system so that the user can speak out the textual sign as it is and the application would extract the information and insert the data in the textual format in the database. Although this approach can save more space as compared to storing images, the speech recognition operation would be time consuming. Also another challenge in the implementation of this approach is that if there are identical signs on the same floor, both insertion as well as retrieval would become difficult.

7. CONCLUSION

Indoor localization remains one of the perennial issues, as none of the previously proposed techniques has been able to come with an efficient and flawless solution to this problem. This paper presents a novel idea of achieving indoor localization by leveraging the human perception of textual signs in a building in the absence of any cellular or Wi-Fi network. The system does not require any dedicated hardware and is an energy-efficient solution to the indoor localization problem. This technique can be applied on any off-the-shelf device without the need for any infrastructure support. The storage of the signs in the textual format results in a compact database and the technique is not computation intensive. Our findings show that the system can accurately determine the location in an indoor environment with 100% accuracy. Even though the accuracy of this technique is very high, it has its own limitations and overheads, which we propose to resolve through future work.

8. CONTRIBUTION

Vimalendu Shekhar designed and developed the database module for this project. He implemented queries and methods for creation of database and insertion and retrieval of records from database. He was also involved in the gathering of signs from the buildings, updating the database with the collected signs and their co-ordinates and the evaluation and testing of the final application.

Shrijit developed the Co-ordinate Shifting and the Location Render Module. He also developed the pinch zoom and the scroll feature in the last screen of the app. He was also involved with the gathering of the signs and mapping their location with the corresponding co-ordinates in the database. A part of the evaluation and testing of the app was also done by him.

Abhishek Kejriwal was involved in the development of entire user interface module. He was also responsible for deciding the appropriate features to be used such as using the AutoCompleteTextView instead of ListView. Code integration for all the modules was done by him. He was actively involved in gathering textual signs from the building and mapping the appropriate co-ordinates to the respective signs and also participated in testing and evaluation.

9. REFERENCES

[1] Han, B., Qian, F. and Ra, M. Human Assisted Positioning using Textual Signs. In HotMobile 2015.

[2] Chen, Y., Lymberopoulos, D., Liu, J. and Priyantha, B. FM-based Indoor Localization. In MobiSys 2012.

[3] Tarzia, S., Dinda, P., Dick, R. and Memik G. Indoor Localization without Infrastructure using the Acoustic Background Spectrum. In MobiSys 2011.

[4] Kuo, Y.,Pannuto, P., Hsioa, K. and Datta, P. Luxapose: Indoor Positioning with Mobile Phones and Visible Light. In Mobicom'14.

[5] Nirjon, S., Liu, J., DeJean, G., Priyantha, B., Jin, Y. and Hart, T. Coin-gps: Indoor Localization from Direct gps Receiving. In ACM MobiSys'14.

[6] F. X. Yu, R. Ji, and S.-F. Chang. Active Query Sensing for Mobile Location Search. In ACM MM, 2011.

[7] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan. Energy-Efficient Positioning for Smartphones using Cell-ID Sequence Matching. In Mobisys, 2011.

[8] http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html

[9] http://developer.android.com/training/displaying-bitmaps/index.html

[10] K. Cheverst, N. Davies, K. Mitchell, and A. Friday. Experiences of developing and deploying a context-aware tourist guide: the GUIDE project. In Mobicom, 2000.