**FEN BİLİMLERİ ENSTİTÜSÜ**
**BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ ANABİLİM DALI**

**BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ PROGRAMI**

# AKILLI BİNA: BİNA İÇİ VE AKILLI YOL BULMA

**YÜKSEK LİSANS TEZİ**

**ALEX GUNAGWERA**

**Danışmanı: Yrd. Doç.Dr. FARZAD KİANİ**

**İSTANBUL**
Kasım 2017

**GRADUATE INSTITUTE OF SCIENCE AND TECHNOLOGY**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

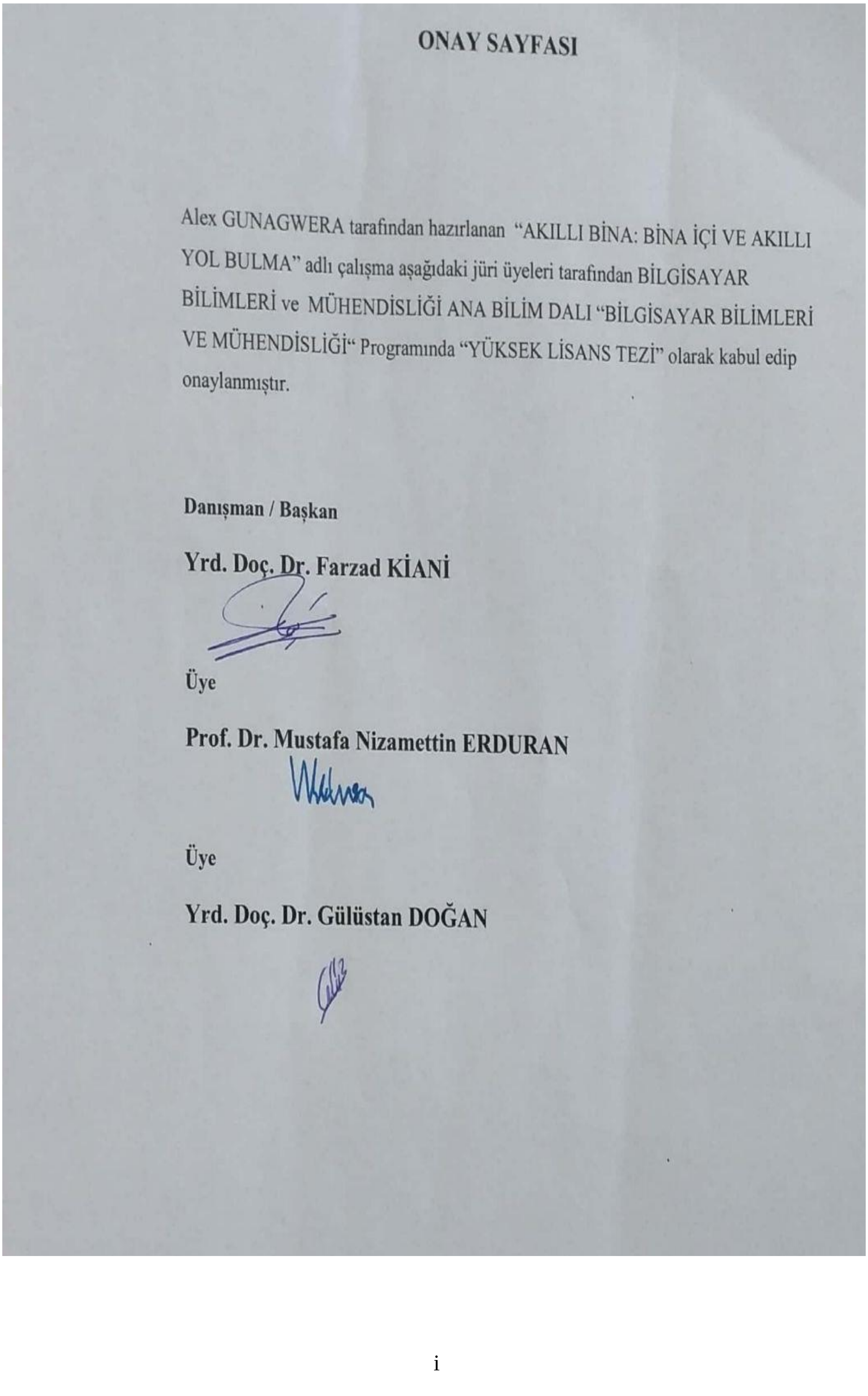
# SMART BUILDING: LOW COST INDOOR POSITIONING AND INTELLIGENT PATH FINDING (ALCIPIPF)

**MASTER OF SCIENCE THESIS**

**ALEX GUNAGWERA**

**Advisor: Asst. Prof. Dr. FARZAD KİANİ**

**İSTANBUL**
Kasım 2017



## ONAY SAYFASI

Alex GUNAGWERA tarafından hazırlanan “AKILLI BİNA: BİNA İÇİ VE AKILLI YOL BULMA” adlı çalışma aşağıdaki jüri üyeleri tarafından BİLGİSAYAR BİLİMLERİ ve MÜHENDİSLİĞİ ANA BİLİM DALI “BİLGİSAYAR BİLİMLERİ VE MÜHENDİSLİĞİ“ Programında “YÜKSEK LİSANS TEZİ” olarak kabul edip onaylanmıştır.

**Danışman / Başkan**

**Yrd. Doç. Dr. Farzad KİANİ**

**Üye**

**Prof. Dr. Mustafa Nizamettin ERDURAN**

**Üye**

**Yrd. Doç. Dr. Gülüstan DOĞAN**

# BEYAN

Bu çalışma İstanbul Sabahattin Zaim Üniversitesi Fen bilimleri Enstitüsü BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI  BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ'ındaki öğrenciliğim döneminde hazırlanmış olan YÜKSEK LİSANS TEZİ tarafımdan yapılmış ve kaleme alınmış tamamen özgün bir çalışma olup bu çalışmamın başından sonuna kadar bilimsel ahlak kuralları uydum. Bu çalışmam süresince elde etmediğim ve tezimde/raporumda kullanmış olduğum bütün bilgiler ve yorumlar için atıf yaptığımı ve kaynak gösterdiğimi, patent ve telif haklarını ihlal edici bir davranışta bulunmadığımı beyan ederim.

**Alex GUNAGWERA**

# ACKNOWLEDGEMENT

I would like to thank my family, teachers, friends and relatives for the steady support they provided me with throughout my programme. Thanks to my advisor for all the guidance, patience and help he offered me. Gratitude to Solicit and Ninho for their aid during the collection of data used in the study. Thanks to Mr. Fahim for the professional advice offered. I thank Şimşek hanım, Hüseyin efendi, and the travelers for making my program not only fruitful but also an enjoyable one.

Yours truly Alex Gunagwera

# CONTENTS

# LIST OF ABBREVIATIONS AND SYMBOLS

| | |
|---|---|
| **Wi-Fi** | : Wireless Fidelity |
| **AP** | : Access Pointd |
| **dBm** | : Decibel milliwatt |
| **GPS** | : Global Positioning System |
| **DR** | : Dead Reckoning |
| **IR** | : Infrared |
| **LAN** | : Local Area Network |
| **IEEE** | : Institute of Electrical and Electronics Engineering |
| **IZU** | : Istanbul Sabahattin Zaim University |
| **API** | : Application Programming Interface |
| **AoA** | : Angle of Arrival |
| **LoS** | : Line of Sight |
| **MAC** | : Media Access Control (Value same as SSID) |
| **MEMS** | : Micro Electromechanical Systems |
| **OS** | : Operating System |
| **RSS (I)** | : Received Signal Strength (Indication) |
| **UWB** | : Ultra-Wide Band |

| | |
|---|---|
| **RFID** | : Radio Frequency Identifier |
| **SSID** | : Service Set Identifier |
| **SQL** | : Structured Query Language |
| **ToA** | : Time of Arrival |
| **TDoA** | **:** Time Difference of Arrival |
| **IDE** | : Integrated Development Environment. |
| ~ | : Approximately |

# LIST OF TABLES

# LIST OF FIGURES

# ÖZET

Mobil cihazların hızla gelişmesine ve iç mekân tabanlı servislerin kapsamlı uygulanabilirliğine rağmen kapalı alan navigasyon sistemi hala bir zor görevdir. Ucuz ama yeterince hassas bir iç mekan navigasyon sistemi oluşturmak çok zordur. Bu çalışmada, düşük maliyetli ve bağlama duyarlı bir kapalı alan navigasyon sistemi teklif edildi. Bu sistem, hibrit fingerprinting ve dead-reckoning yöntemleri kullanıyor. Bu sistem, bütün kullanıcılar için - görme özürlülere veya bedensel engellilere bile- uygundur. Bu sistem üç ana bölümden oluşmaktadır; kullanıcı takibi, bağlam-duyarlı en uygun rota hesaplama ve dinamik yol gösterimi. Sistemin üst tabanı ***0.8-1.3*** metre ortalamasıyla ***2*** metredir. Kullanıcıların tek ihtiyaçları akıllı telefondur.

**Anahtar kelimele**r: Kapalı (iç alan) navigasyon, izleme, algoritma, akıllı telefonlar, sensörler.

# ABSTRACT

Despite the rapid improvement in mobile devices, overall gradual growth in the ubiquitous computing field, the wide applicability, more usefulness of location based services in general and indoor navigation, in particular, building a sufficiently accurate, efficient and relatively cheap indoor navigation system in a GPS-free environment is still a challenging task with a lot of tradeoffs and constraints to put into consideration. In this study a simple yet robust, low-cost, context-aware user-interactive, user-friendly hybrid of finger-printing and dead-reckoning indoor navigation system suitable for both the visually impaired and the physically disable as well that takes advantage of the results yielded by sensor fusion is proposed. The presented system is also designed to allow for efficient evacuation of users in cases of emergences. The prototype is made majorly of the following parts; user tracking, optimal, context-aware and dynamic route calculation and planning and dynamic route representation with an upper bound of ***2m*** and an average of ***0.8-1.3m*** accuracy. All that is required from the user is a smart phone without installation of extra hardware.

**Keywords**: Indoor navigation, tracking, algorithm, smartphones, sensors.

# 1 INTRODUCTION

In this section, an introduction to navigation in general is presented. The history, progress and potential of both indoor and outdoor navigation is provided with more emphasis and greater detail being put on indoor navigation because that is the main subject of this literature.

The Global Positioning System (GPS) has undergone tremendous improvement since the 1900s and it, indeed is considered one of the most successful navigation systems known to date. However, it is still inefficient for sufficiently accurate positioning in both indoor environments and environments with many tall buildings such as skyscrapers since such buildings block or interfere with its signal transmissions.

So many studies have suggested various options for indoor positioning and navigation using various technologies that would aid this cause. These technologies range from Bluetooth, the Ultra-wide Band (UWB), the use of sound (ultrasound), Infrared (IR), Radio Frequency Identification popularly referred to as RFID, wireless sensor networks in general, and visual analysis to magnetic signals. Using one or a combination of these technologies researchers, industries, governmental organizations and universities have come up with indoor positioning and navigation systems. The choice of the technology or technologies is pretty much dictated by the priority of the functionality, availability, cost and importance of the resultant application. Most have had to tradeoff between cost, performance and all availability of other resources such as the required, time, hardware or software.

The ubiquity of mobile devices (such as cell phones) has led to the introduction of

Location Based Services (LBS), or Location-Aware Services. LBS aim at providing information/services relevant to the current location and context of a mobile user.

One of the first several LBS applications, named Active Badge Location System, was introduced in R. Want et al. [1]. This system employed infrared technology for tracking a user's current location and used this location to forward phone calls to a telephone close to the user. Since then, many researchers have studied this topic and as such many LB applications have emerged over time as a result of the increased market of both outdoor and indoor location based services and applications.

Recent technological advance such as the gradual maturation of ubiquitous computing M. Weiser series [2], or pervasive computing, and the evolution of mobile devices (such as PDAs, cell phones, etc.) and wireless communication (3G, Wireless LAN, Wireless Sensor Network, etc.) has further increased the pace of progress A. Butz et al [3]. They also make an overview about map-based mobile guides using the dimensions of Positioning (either GPS, WiFi, UMTS, or other), Situational factors (user or context-related), Adaptation capabilities, interface/use interaction (multi-model or others), Use of maps (2D vector, 2D bitmap, 3D model etc.), and Architecture (client-server, interacting, multi-blackboard or multi-agent system). These dimensions are roughly defined and further subdivision of some of these dimensions is needed. Raper et al. [4] developed a much more complete classification which used the axes of Application (tourism, recreation, transport, and museum), Positioning, Architecture, Presentation, Context relevance, Delivery (pull or push), Use case, and Adaptivity (resource adapted, resource adaptive, resource adapting), and then made an investigation on LBS applications in the published literature.

However, these studies are mainly for outdoor applications. For indoor applications, different positioning algorithms and technologies are needed to replace GPS. As a result, more detailed dimensions on positioning, such as signal (infrared, ultrasonic, radio signals, etc.) and signal metrics (Cell of Origin, Time of Arrival, Time Difference of Arrival, Angle of Arrival, to mention but a few), are needed to evaluate the various positioning technologies. As context-awareness is very important for LBS systems, there should be some dimensions that evaluate the context-awareness of these indoor applications owing to the fact that accurate user tracking is not easy, in fact, it is difficult [11].

Still the serious participation of well-known firms' sectors such as Bing maps and Google maps in [19] and [20] as well as the steady show of interest of companies such as [21] and many more, further stresses the importance and potential of indoor navigation. However, even for such powerful companies accurately mapping all buildings and provided precise navigation is still very hard, if not impossible due to the continuous growth of the construction and business industries. It's simply hard to keep up. In some countries such services are even not heard of yet.

As the range of LBS applications is vast, it is practically impossible to introduce all of them here due to brevity.

Mobile navigation systems, which aim at providing wayfinding services and tracking to the user, are the most important applications of LBS.

However, in indoor navigation and localization, accurately pinpointing the location of a user and correctly and optimally guiding the user to the desired destination is pretty challenging.

Till to date, available techniques and attempts at indoor mapping and/or localization can be distinctly categorized in two major groups; those that employ fingerprinting [5, 6] and those that do not [7, 8, 9]. Most fingerprinting based techniques are faced with the problems ranging from susceptibility to intervention by humans to generating discriminative signatures for the access points (APs). Also the Received Signal Strength Indicator (RSSI) varies over time due to noise, multipath, reflection off and absorption by surfaces and other objects.

The techniques that use fingerprinting, say [6], include Wi-Fi RSSI and FM broadcast fingerprints to help map the indoor environment in question. Indoor locations are attached to signal strength values thereby mapping the environment. Techniques that do not involve fingerprinting generally the dead reckoning technique to calculate the user's current position in reference to pre-known values. There is also the issue of presenting the route to the user in the most efficient way possible as users do not understand signal strengths in decibels (say -45dB) or coordinates in form of (x, y) or (x, y, z) and instead objective descriptions such as “to the left of the canteen right in front you” or suitable graphical presentation and so on and so forth are much easier to understand and hence make more sense. All these and many more problems are considered and addressed in our study.

The rest of this literature is organized as follows. Section 2 presents related works, section 3 introduces the methodology used in this work, section 4 presents the model of the proposed prototype, section 5 shows the field experiments and results performed and discussions concerning the results. Section 6 presents the future works and possible improvements to the proposed system. Finally, a summary of the study is provided in section 7.

# 2 RELATED WORK

In this section, past and recent trending works related to or similar to this study, to the best of our knowledge, will be represented. Such works include those with varying parameters, methodologies application or accuracy of the results obtained in the presented studies.

Indoor navigation has, of recent, become a serious and popular field of interest owing to its wide applicability, usefulness and the improvement in ubiquitous computing and the development of mobile devices in general and hand held smart phones in particular. As mentioned above, previous attempts to map indoor environments can be generalized in two main categories; those that employ Wi-Fi-fingerprinting and those that do not. Those that do not use fingerprinting exploit techniques such as crowdsourcing [10], triangulation and trilateration. This can be more broadly categorized in two again; those that use wireless and sensor technologies (such as most of those discussed later on) and those that use image recognition and processing [16].

[12] Proposed an enhanced form of GPS for indoors and urban area usage. However, the accuracy attained was still not impressive as GPS signals were still being lost through absorption or reflection from walls and other objects – the so called multipath problem. This coupled with other GPS related issues on its application indoors render it not the best of choices. The complex nature and structure of the indoor environment, however, still poses a practical solution to the indoor navigation problem using GPS – even with the Chinese Satellite system [13] cooperating with the recent Galileo Positioning System, despite the potential they present. We bypass GPS related issues by not relying on the GPS system for positioning or user tracking, we instead use Wi-Fi fingerprinting coupled with dead reckoning.

Sensors mounted on feet coupled with ultrasound beacons together with dead reckoning are used in [66] to help rescue teams navigate in indoor environments. The sensors were attached to the shoe laces of the navigators. Tests were performed with an online implementation of their dead reckoning algorithm and recorded the users' trajectory on a computer during their experiments and analysis. They additionally attached inertial measurement units to the chests of subjects.

In [14], they proposed pressure sensitive floors whereas [15] proposed pressure sensitive shoes. These systems may be able to solve the indoor positioning and tracking problem but not only do they not offer a means for navigation, they are also extremely expensive to setup especially for mass deployment, not to mention the necessity to track the weight of each of person in case of pressure sensitive floors. Setting up these requirements for pressure sensitive floors and shoes can barely be categorized as low-cost. Besides, this system can only, at best, provide tracking, not navigation.

Tactile landmarks and smartphone sensors are used by researchers in Navatar [67] to suggest an indoor navigation system for the visually impaired. The tactile landmarks are used as milestones to help guide the blind people. They also apply particile filters on their data.

Alexey et al. [75] suggest a Kalman filter based Wi-Fi fingerprinting and dead reckoning indoor navigation system as well. The major differences between their work and ours include: first of all they highlight their usage of a Kalman filter. As pointed out by most studies and in section 3.7.2 of this study, a Kalman filter is not really suitable for this kind of work, we use a low pass filter. We also use a text based feedback relay approach coupled with audio feedback. This is in order to make the application helpful during emergence situations and emergence evacuation scenarios and furthermore take the visually impaired and blind into consideration.

Trein et al [10] suggested an approach that does not require the use of Wi-Fi signals at all. They simply use a crowdsourcing approach based on sensor data and a step detection algorithm to determine the position of the user. This different approach is not suited for emergence situations.

[69] Discusses path planning and following algorithms in an indoor navigation model specifically designed for the visually impaired. They discuss the level of accuracy, environmental infrastructure among other necessities required to provide indoor navigation to the visually impaired and the blind. Their path planning uses Dijkstra’s algorithm and A* on their new data structure (the cactus tree).

In [18], the steps to deal with varying position-tracking accuracy in mobile augmented reality systems are discussed. They argue that neither the problem nor the solution is limited to only augmented reality problems only but to all systems that require relatively accurate position tracking – which our system qualifies to be.

The SmartCampusAAU project [70] provides a platform with a backend that makes it easy to *enable* indoor navigation in buildings. It is cross platform i.e. it supports android, iPhone and windows phone. They use crowd sourcing to gather data. They also maintained a backend on the server for the project. It operates more like google maps except that it is available for all major platforms (those mentioned above). Also it is not possible to do device-based positioning with this project.

In [22] a time dependent optimal routing model for emergency evacuations in which the route is made basing on the position of the sensors and not the architecture of the structure itself is proposed. In the model, the geometry of the structure in question is ignored and all operations and algorithms developed depending on the available sensors. Hence a change or misread from the sensors could easily lead to significantly flawed navigation and routing service results. As it was designed to handle emergence situations, such potential sources of errors should best be avoided. Furthermore, their system is specifically designed for micro-scale temporary emergence evacuation. So their simulations included human detection with 5-10 second intervals. Our proposed system is a multi-purpose prototype that provides navigation and optimal path routing no matter the situation.

A project aimed to study the used of mobile devices (smartphones) with android OS platform to aid the localization of robotic systems indoors was developed in [64]. They also applied the sensor fusion technique on the device’s inbuilt sensors i.e. the magnetometer, accelerometer, gyroscope etc. They use the camera to extract data about from and about the surrounding to aid in the positioning speculation of the robot. They claim that their study was successful.

A method to train the Wi-Fi fingerprinting database using sensor-based navigation solutions is presented in [65]. They restrict the time length of available indoor navigation trajectories and perform post-processing to improve the sensor based navigation solution since MEMS sensors provide only a short-term accuracy. They combine middle-term trajectories that move out of and into an indoor building to gather data used in database creation.

One of the difficulties faced throughout recent research is the accurate prediction of signal propagation. This can be approached using a technique called Location fingerprinting, that is based on measuring actual signal strengths from surrounding access points [23, 24, and 25]. Furthermore, [24] provides a purely Wi-Fi fingerprinting based approach to localization and indoor navigation. They directly perform scans and store the RSSI with the corresponding MAC addresses into the databases without further processing and simply read and compare the obtained RSSI from the user with the pre-stored values. If the RSSI is within a given range (in their case a $+4 \geq$ RSS $\geq -4$), then the user is at a correct location. And they thus let the users know so. Else, they are at a wrong location. Major draw backs to this approach, however, include; requirement of constant database update due to temporal variation in RSS over time and incase of change of the environment. Also, sometimes the variation n RSS is way greater or less than the (+4. -4) given interval. Under such circumstances (especially for very large or very small roomed buildings), erroneous navigation is inevitable. We apply dead reckoning using filtered data to overcome and minimize the effect of temporal variation of RSS and use the range of (+5, -5). There is also the issue of latency caused by such variations and multipath. Since calculations are performed much faster, this is also minimized by the dead reckoning branch of the algorithm.

The system proposed in [25] apparently predicts where the user might want to go next. They also use a robot to aid in the data gathering process. Their system requires no user input whatsoever. Whereas, this approach is viable for general indoor navigation, it’s applicability in specific situations is questionable. For example in terms of emergencies, let alone taking care of the visually impaired.

Redpin [71] is a croudsource based system that also uses wifi fingerprinting to enable indoor navigation. It uses a server-client model where the server stores the database that contains the necessary fingerprint information. In this system, the training phase of Wi-Fi fingerprinting was omitted and instead use crowdsourcing. So the system is trained as users use it.

An indoor navigation model that supports optimal length-dependent routing is suggested in [26]. It, however, is limited to PCs, uses a server client model and necessitates extra plugin installation thereby barely qualifying as cost efficient. One needs to have at least a laptop to use the application. This is very inconvenient for navigating indoors especially nowadays that mobile phones provide pretty much all the basic functionality sufficient enough to facilitate navigation.

The CricketNav project [11] proposed the design and implementation of an indoor mobile navigation system using the cricket infrastructure developed in MIT labs. Their project requires installation of hardware developed in MIT labs. They also developed special Cricket beacons and Cricket listeners to aid their cause.

[27] Proposed a resource-adaptive mobile navigation system (REAL). Their complete project had three major components; an information booth that had a 3D graphics workstation, an indoor navigation system based on strong infrared signal transmitters planted into ceilings and PDAs used for presentation and finally a head-set laptop combination for outdoor navigation. The routing information is presented depending on the kind of device used. Route optimization is also dependent on the resources available on the device being used.

CYBERGUIDE [28] is among the first systems that employed location aware information to aid tourists. The project was designed to help tourists both indoors and outdoors. It comprised of two major components; an indoor and outdoor component. This project's indoor component depended on beacons that broadcast a unique ID using infrared signals from infrared beacons. On the other hand, the outdoor system used GPS. Both components functioned independently from each other.

In [30] a probabilistic navigation system for pedestrians based on mainly inertial sensors found in a specially made device [31] is presented whereas The NEXUS [14] system also tries to provide a general framework for mobile and location aware computing. The concept of an augmented world is used to keep information necessary for a user's location. It is the basis model for the virtual information towers that connect information objects in their work.

In iDocent [68], a server-client based user navigation system using fingerprinting and triangulation is presented. The let the server store data and do route calculation. Major drawbacks with this approach include latency. Also, in case the server breaks down. The system is rendered useless, let alone triangulation drawbacks such as being position dependent. So in case the currient position or estimated position is wrong. Accumulative error is inevitable. The system we propose is a standalone self-dependent system.

A framework to enable intuitive navigation guidance in complex large-sized buildings by utilizing a topological way-finding method to generate paths in [46]. They integrate the building information model with their new algorithm to subdivide the spaces. They also claim to improve the visual information by using a new method to render all three dimensional views. This they achieve by using imaging serviced using a client-server design with supercomputer computation power. They apply the image processing approach to help improve visualization.

In our study we present a user-interactive context-aware hybrid system of dead reckoning and fingerprinting that provides navigation services to users. It also puts into consideration visually impaired and physically incapacitated users by provided audio directions and a list of options that are aimed to make the user’s navigation experience less troublesome. For example, a user on a wheel chair can set the system such that stairs are excluded from path/route presented to the user.

# 3 SMART BUILDING: A LOW COST INDOOR POSITIONING AND INTELLIGENT PATH FINDING (ALCIPIPF)

In this section, the general methodology and architecture of the proposed approach is presented. The algorithms (path calculation, step detection etc.), feedback presentation mechanism major sensors used (accelerometer, gyroscope, magnetometer) and tools used as well are also presented in this section.

At the end of the section is the system architecture of the ALCIPIPF prototype.

## 3.1 Methodology

Given that attention Indoor navigation has attracted, so many studies about and around the topic have been carried out. Alongside these studies a plethora of approaches to achieve this feat have been presented in previous literature.

The approaches for localization range from the use of ultrasound [66], pattern matching techniques such as Computer Vision [72, 77, and 74], signal distribution techniques such as in NAVIO [79] and LaureaPOP [76], Bluetooth e.g. UCPNavi [73], mere dead-reckoning e.g. NAVIO [78], Nakamura et al. [81] and so on and so forth.

After localization comes path planning and calculation, the main algorithms used being A* and Dijkstra, finally provision of feedback from the system to the user. Feedback modality can be haptic, speech, Audio and visual, just visual or audio cues or a combination of speech and haptic e.g. in Tadokoro [80].

In this section, we present the complete methodology that we propose, accompanied with the more important algorithms involved.

### 3.1.1 Important Definitions

#### 3.1.1.1 Accuracy

In this study, the term accuracy will be used to refer to how close the measured value, during navigation, is to the value stored in our knowledge base.

#### 3.1.1.2 Positioning and localization

These two terms will be used interchangeable to refer to the absolute coordinates such as position (x, y, z) or to relative locations in the building such as class room EG117 as returned by either the DR algorithm or fingerprinting algorithm respectively.

There are various methods and techniques used to locate or estimate the position of devices indoors. These techniques are grouped differently from study to study but they all boil down to the same thing. But as time goes and more research in made the narrower the techniques are classified. E.g. In [47], Y. Gu et al. categorize these techniques into four, namely; triangulation, visual analysis, proximity technique and fingerprinting. Some studies cover them as three major groups including; triangulation, trilateration and fingerprinting.

We shall consider the three major techniques of determining the location/position of the device/user in this study;

- **Triangulation**

  In this method, the AoA of a signal from a transmitter to a receiver is measured and is used to determine the location of the user [48]. This done by forming triangles from the transmitter (say user's device) to known points (APs) – hence the name *triangulation* and then calculating the AoA is performed. This technique has got limitations though such as suffering from multipath, requiring LoS among others [49] which might be hard to obtain indoors.

- **Trilateration**

  This technique is also referred to as multi-lateration. It is a technique that exploits the ToA or TDoA to calculate the distance from known nodes (APs) to client devices (user's device/smartphone) by utilizing the RSS [50, 51]. ToA finds the device's location by measuring distance of the transmitter (user's device) from the receiver (APs whose location is known). It uses the travel time of the signal from the transmitter to the receiver through the formula;

  $$Loc = Time\ x\ Speed$$ **Equation 3.1**

  Where *speed* is a contant and *Loc* is the location to be determined. Now only time needs to be measured and all the parameters are set [49] in order to determine the location *Loc*. While using ToA, both receivers and transmitters have got to be synchronized.

  Unlike ToA which requires synchronization of both receivers and transmitters, TDoA requires synchronization of receivers [48].

- **Fingerprinting**

  This technique basically comprises of measuring, recording and storing signal strengths (in dBm) received from APs in range at given Points of Interest (PoIs) and storing them in a suitable fashion in a knowledgebase or database. Extra data such as coordinates of the PoIs can also be stored as seen fit by the one performing the fingerprinting.

  The stored data stored can be of two main categories; probabilistic [55] or deterministic [41]. [57] provides a detailed comparison of the probabilistic and deterministic *methods* for indoor localization in general.

  Previous studies [53, 54] show that this method is of paramount importance when it comes to indoor localization.

  This technique comprises of two phases

  - **The Offline/training phase**

    During this phase suitable information about and at given PoIs is recorded and stored in a database. This information may include, the RSSI at a given PoI, the MAC address(es) of the APs that are within range at the PoI, the coordinates *(x, y)* or *(x, y, z)* of the given PoI, descriptive features that may help guide the navigator toward their target such as ***"printing machine next to the stairs on the left of the exit"***. In this study an extensive offline phase is carried out.

  - **The online phase**

    After all the fingerprinting is done during the online phase, as the users can traverse the mapped/fingerprinted environment with/without the MAC measurements are taken. These are then compared with the values in the knowledge base and, ultimately, an estimation of the navigator's location is returned.

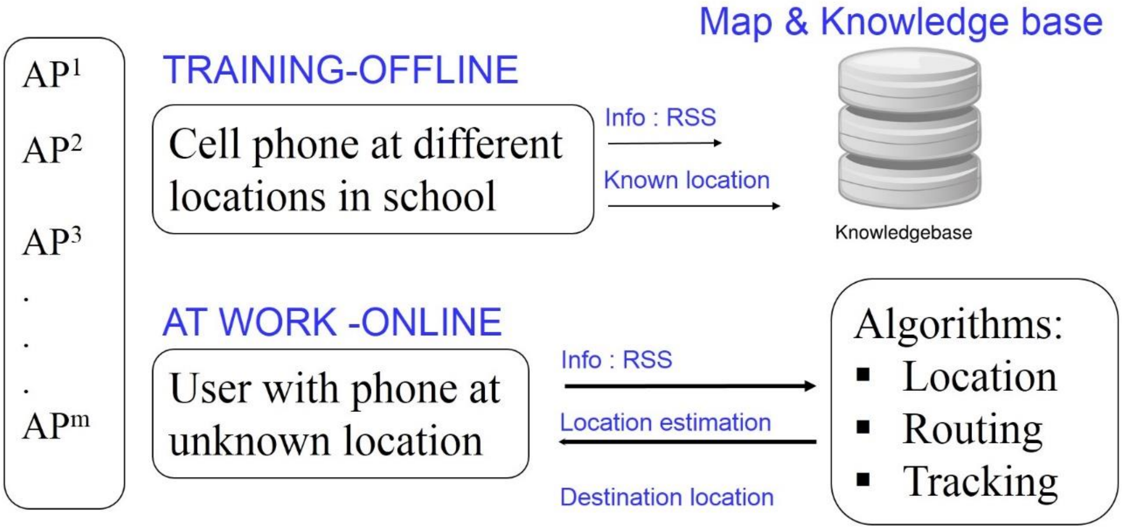
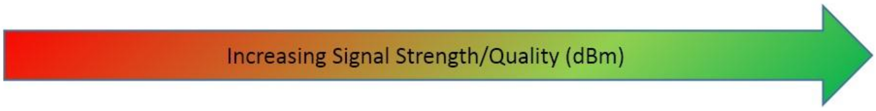


**Figure 3.1** Wi-Fi Fingerprinting

**Table 3.1** Example Fingerprint data

| MAC | Class EG104<br>Coords(120,15,2) | Class EG103<br>Coords(155,17,2) | Class EG102<br>Coords(163,16,2) | Class EG101<br>Coords(171,19,2) |
|---|---|---|---|---|
| 9c:1c:12:06:73:31 | -77 | -62 | -58 | -46 |
| 9c:1c:12:06:9f:80 | -60 | -58 | -52 | -43 |
| d2:7e:35:be:35:fc | -55 | -50 | -42 | -37 |
| 9c:1c:12:ff:77:00 | -51 | -48 | -39 | -35 |



The table above shows example fingerprint data

In the table above are the classroom names, their respective coordinates, the MAC addresses in range at given classroom locations and their respective strengths. For demonstrative purposes these values have been altered for their true values. As one moves from left to right, the signal gets *better*. The closer to 0 the RSS is, the better the signal quality [58, 59].

### 3.1.2 Tools Used

Now we discuss the major tools used. The tools used can be categorize in two main groups; software and hardware.
The main hardware components include; the smartphone (the nexus 5 in our case study [60]) – which will be the device we use to locate, navigate and track the user, and the APs. These we assume are installed in the facility where the application will be used.
Major software components include; the programming language (we used Java Version 8 [61]). Android (Marshmallow [62]) was the platform used for tests and experiments.
We used SQL [63] database for data storage and manipulation. The android platform comes equipped with a free light version of the SQL database.

Wi-Fi fingerprinting and smartphone's inbuilt inertial sensors are used in our study. This choice is made because Wi-Fi Access Points (APs) are readily available and are already installed throughout not only the school campus (the test case in this study) but also in most indoor facilities nowadays such as industries, airports, hospitals to mention but a few and pretty much everyone owns a smartphone. Given these conditions, a low cost functioning, robust navigation system can be built.

The smartphone's accelerometer provides x, y and z coordinate values that, with data processing, can not only provide a lot of information but also provide pretty amazing results. Information it can provide ranges from the relative position of a user in general and the user's device in particular, the distance travelled over time and so forth. However, the accelerometer values are not used to determine the distance travelled by the user in this application because that would require double integration of the achieved values whose error bounds increase tremendously with the operation. We instead use fingerprinting, dead-reckoning [42] whilst employing the classic Euclidean distance formula in ***Equation 3.5*** below to estimate the distance travelled by the user to a sufficiently estimate the pose of the navigator. Using the magnetometer and gyroscope, the user's orientation and direction can be and are determined and it's much easier to determine whether the user is on or off track as smartphones have got Wi-Fi receivers and scanners. It should also be pointed out at this point that smartphone technology nowadays has improved significantly. So smartphones are capable of carrying out computation tasks of this scale without negatively affecting the phone's daily/expected tasks.

### 3.1.3 Sensors

In this section a brief overview of the sensors most relevant to the project will be made.

### 3.1.4 Accelerometer

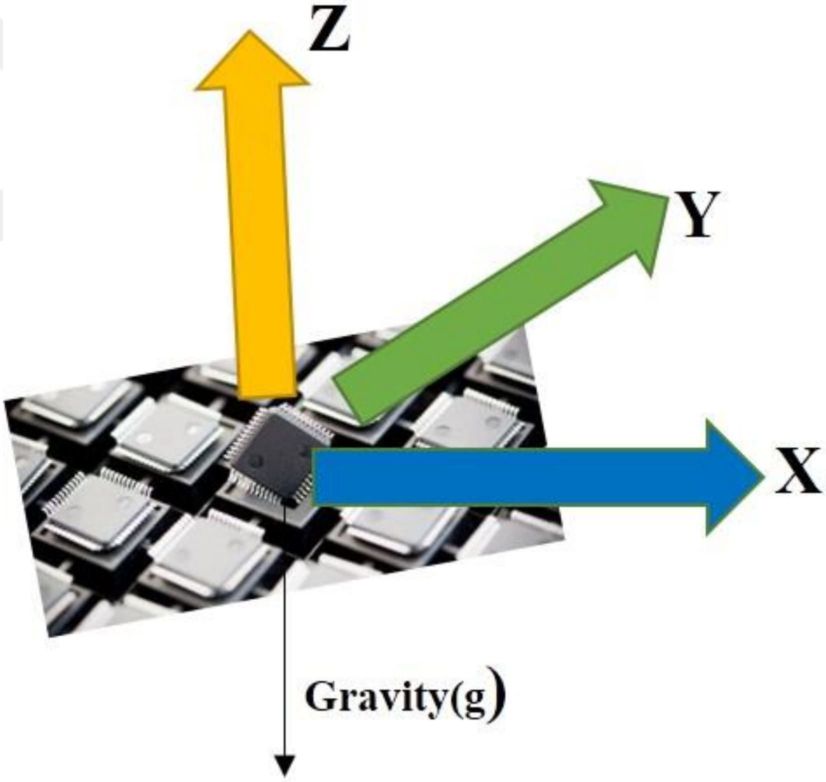


**Figure 3.2** The Accelerometer

The acceleration in most smart phone devices is basically related to the phenomenon of weight experienced by a test mass that's found on the reference frame of the accelerometer (device). An accelerometer is thus a device that measures this proper acceleration and hence, this acceleration is not necessarily the change of velocity of the smart phone in space/coordinate acceleration. Given the general structure of this accelerometer, a device at rest relative to the earth’s surface would show roughly 1*g* upwards due to its weight. Where *g* is the gravitational force whose unit is $m/s^2$. The accelerometer of a smartphone measures both dynamic motions such as movement, phone tilting in the x and y axes and static forces such as the gravitation force (in the z-axis) [33].

### 3.1.5 Gyroscope

**Figure 3.3** The Gyroscope

A gyroscope is a device that, according to the principles of angular momentum, sensors [34], measures orientation. Unlike a mechanical gyroscope which consists of three two gimbals onto which a spinning a wheel that resists changes in orientation is attached, conventional gyroscopes in smart phones generally are made of Micro Electro-Mechanical Systems that measure angular rate, hence the name rate-gyros, while mechanical gyroscopes observe the change in the angle of adjacent gimbals as the spinning wheel remains at a constant global orientation.

### 3.1.6 Magnetometer

A magnetometer is an instrument used to measure the strength and/or directions of the magnetic field in the surrounding area of the instrument.
Magnetometers can be divided into two basic types: scalar magnetometers that measure the total strength of the magnetic field to which they are subjected, and vector magnetometers (the type used in this project), which have the capability to measure the component of the magnetic field in a particular direction, relative to the spatial orientation of the device [35].

### 3.1.7 Steps Involved

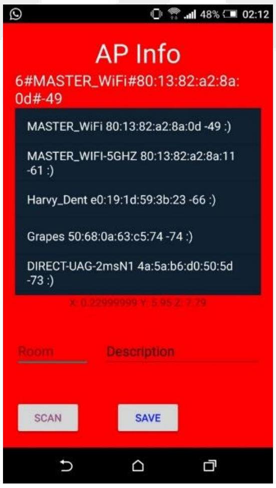
The system's fingerprinting mechanism set up was made in phases, an online phase and an offline.

During the online phase, fingerprints were recorded and stored for reference. It's of paramount importance that all stored fingerprints have distinct values from each other. Fingerprints basically have signal strength value at a given location. This is coupled with a short vivid description of the location (referred to as node during route planning and calculation) for assistance in routing. For example; to the left of the canteen right ahead of you, since indoors mere signal strength values (say -75dB) or coordinates such as (x, y, z) carry no comprehensible meaning to the user who may be a temporary guest at the institution in question. An appropriate *floor identification (floor id)* value is also added to the AP's identifiers. This is very useful for multi floor buildings. It helps to distinguish one floor from another with ease. Assigning a value of say 11 to the first floor, 22 to the second floor, 33 to the third floor and so on and so forth, helps distinguish floors that receive sufficiently strong signals from identical APs since the height from one floor to the other is significantly.

After data collection, then we can go onto navigating the user(s). An instance of the node's coordinates is also stored – this is mainly for usage by the dead-reckoning algorithm as explained later. The dead reckoning service runs in background and, basing on the user's concurrent location, the user's entered parameters (main source and destination), aids in the timely warning of the user in case the user is off route. And, furthermore, triggers route dynamic recalculation thereby reducing the delay these operations would cause in real time.

### 3.1.8 Data Acquisition

One of the most challenging and time consuming tasks in any fingerprint based approach of indoor mapping is the generation of discriminate fingerprints for given points of interests (POIs/nodes). This process is not only time consuming but also requires that empirical results be obtained a couple of times due to signal instability of the Wi-Fi networks resulting from interference, multipath and so on and so forth. As such, we felt the need to develop, as part of the project, an application that will aid to ease the process. This we believe, will not only benefit us, but also future researchers in the field.



```
6#MASTER_WiFi#80:13:82:a2:8a:0d#-49
MASTER_WiFi 80:13:82:a2:8a:0d -49 :)
MASTER_WIFI-5GHZ 80:13:82:a2:8a:11 -61 :)
Harvy_Dent e0:19:1d:59:3b:23 -66 :)
Grapes 50:68:0a:63:c5:74 -74 :)
DIRECT-UAG-2msN1 4a:5a:b6:d0:50:5d -73 :)
X: 0.22999999 Y: 5.95 Z: 7.79
```

**Figure 3.4** Data Acquisition and Configuration Assisting App

Shown in **Figure 3.4** above is part of the smart system that helps in the data acquisition process. The system keeps scanning the environment at a constant pre-specified interval – the default is 2 seconds. The user only needs to walk through the indoor environment while stopping at significant POIs (nodes). If the user stops at a place for a period greater or equal to 3 seconds, the system will automatically save the place in the application database. Saved will be the relevant information for both the dead-reckoning and fingerprinting algorithms. Such information includes filtered sensor data, room name, ranked RSS (Received Signal Strength) from nearby Access Points (APs) and a short description of the node. This data is displayed to the user before it's actually used in the auto Graph (to be described later) generation phase. This gives the user time to make necessary changes to the data before it's saved. The user (one doing the mapping) then presses the saved button if he/she's satisfied with the current information. The saved information is then used in the generation of an undirected connected graph [36] used in route calculation and path presentation.

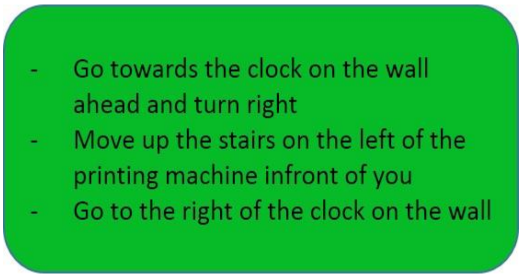
### 3.1.9 Route Calculation and Planning

A modified and optimized form of Dijkstra's shortest path algorithm is used to generate an optimal route from the user's current position to the desired destination. We modified Dijkstra's algorithm [37, 38] to suit our needs in various ways, ranging from enabling physically challenged users to dodge staircases in tall buildings to being able to evacuate as fast and safely as possible through emergency exits. Most path planning algorithms and applications use mainly either graph-based approaches or cell-based approaches [39]. Our system belongs to the former.

Dijkstra's algorithm basically finds the short path, given a graph $G$ comprising of a set of Vertices, $V$ and a set of Edges, $E$ so that the graph can mathematically be represented as $G = \{V, E\}$. Attached to each edge, $E$ is a weight $d$. A special vertex $s$ can be fixed and considered as the source. The algorithm then finds the shortest path from $s$ to each $v \in V$. We modified the algorithm such that the shortest path is found from a specified source to a specified destination. Extra parameters are also fed to the algorithm so that directions such as *left & right* are catered for depending on the user's current location and the desired destination. If the user digresses from the right path, the system automatically notifies the user and recalculates the route/path updating the source as the user's current position whilst keeping the destination fixed.

### 3.1.10 Route Representation

After successfully calculating an optimal route from the user's current location to the desired destination, it is very crucial that this information be clearly presented to the user in the simplest way possible. Visually impaired and physically challenged users should also be put into consideration while choosing the display mechanism of the route information to the user. Either a ***dynamically updated visual 2D or 3D map-model*** [44] or simple ***precise list directives*** in form of text instructions coupled ***with audio*** assistance would do just fine. This is because we would not want the user to always be locked on to the screen while travelling, but the updated information should be always available when needed, say for consultation. We therefore integrate text instructions in our system with audio assistance to the users. The instructions/directives are kept as precise, accurate and understandable as possible. *The navigator is free to choose the time interval over which the audio directives are issued in seconds.* This is done because to some people it might be too fast while to others it appears to be too slow. Of course as the navigator moves from one node to another, as the route is updated, the audio directives are also updated. This choice can be made in the application's settings. By default, this is set to 10 seconds.



- Go towards the clock on the wall ahead and turn right
- Move up the stairs on the left of the printing machine infront of you
- Go to the right of the clock on the wall

**Figure 3.5** Route representation as simple precise textual directives

In this study, we choose the later because it is easier to comprehend and follow – especially during emergencies. Some people are biased towards map reading. Furthermore, we think it is easier to listen to directives or quickly comprehend written directives during an emergence than try to understand and follow a map in an unfamiliar environment.

### 3.1.11 Position and Tracking

Signal strength received from the APs (RSS) and data received from the sensor fusion operations are used for user positioning and tracking. Whichever set yields the least error compared to the values stored in our data base is considered as the current user position. This is done in order to compensate for temporal variations in signal strength or random noise received from the smartphone’s inertial sensors – though most of the time the two are used to support each other.

#### 3.1.11.1 Detecting User’s Steps (The algorithm)

Recent versions android (4.4 and later) support the step counting feature which can easily be used to detect user steps. However, the accuracy of the API used varies greatly from device to device. During the course of this study, it was found to be more accurate while using Google’s nexus 6 phone than with the HTC M8S phone. Once in a while the step detection was delayed in the nexus device as well. This delay was more pronounced in the HTC device rendering the current API alone unusable for our purposes. Need for a more consistent, faster algorithm arose. The approach for step detection in this project is pretty similar to that employed in [10] and [40].

By applying a few operations on the accelerometer readings we can get the user’s steps.

The algorithm to detect the user’s steps is as follows. Empirically determine a suitable window to pass over filtered sensor readings to categorize them as either a step or not. When a user takes a step, the accelerometer magnitude values show a peak. This window’s value is of paramount importance to the accuracy of the results yielded as a very small value gives too many steps since noisy data or even slighter motions can be detected as steps and, conversely, an extra-large window will give less steps than those actually taken by the user.

Let $R_a$ represent accelerometer readings, $W$ be the chosen window, $M(R_a)$ be the median accelerometer reading, $T_{elap}$ be the elapsed time value, $D(W)_{cur}$ be the current standard deviation of the window and $D_{thres}$ be the standard deviation threshold value of the window.

We say the window represents a step if;

$$Max|R_a| \ = \ M(R_a)\forall R_a \epsilon W \qquad \textbf{Equation 3.2}$$

$$and\ D(W)_{cur} \geq D_{thres} \qquad \textbf{Equation 3.3}$$

Over a given time $T_{elap}$ for any two successive steps.

With most Points of Interest (POI) being fingerprinted studies such as [41] find it sufficient to use the fingerprints to estimate the user's current location. In this study we further supplement this practice with a dead-reckoning service to *improve* accuracy and performance of the application.

For example assuming a user moves from a start position to a point $p(x_1, y_1)$, his position can be defined in terms of the distance ($d_1$) travelled and the direction (α) can be easily determined using the Magnetometer sensor. Now, if the user moves to point $q(x_2, y_2)$. The successive position can be obtained using the dead-reckoning [42] technic as follows:

$$y_1 = d_1\, sin\alpha$$
$$x_1 = d_1\, cos\alpha$$
$$y_2 = y_1 + d_1\, sin\beta$$
$$x_2 = x_1 + d_1\, cos\beta \qquad \textbf{Equation 3.4}$$

The Euclidean distance, $d$, between the two points can be obtained using the equation;

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad \textbf{Equation 3.5}$$

Dead-reckoning basically requires the starting position, the change in direction/orientation and the actual or an estimate of the distance travelled by the navigator. If, however, any of these parameters is not sufficiently accurate, dead-reckoning yields accumulative errors [42, 43].

The results yielded by the dead-reckoning algorithm are always checked and compared with the results yielded by the fingerprinting algorithm and the result with the least error is considered the true result, thus correcting and updating the algorithm of the more erratic algorithm – assuming that the result return by the more erratic algorithm is catastrophic, else no need to update the parameters. Say if there's a difference greater than 2 meters between the returned results.

#### 3.1.11.2 Filtering Data

The need to filter data arises, mainly, in order to;

1. Improve accuracy
2. Minimize errors

Since the measurements taken by sensors are inevitably noisy. Causes of noise range from the calibration techniques used in the sensors to the very materials that make up the sensor hardware, among others.

Also, there's the problem of RSS fluctuation; there is temporal variation in the RSS values obtained e.g. the RSS reading obtained when the building is not crowded (empty) are different from those obtained during work hours. There is 5 or so dBm difference. Not that it will cause a disastrous change in the results, but it will have an impact on the database records and thus, should be taken note of.

##### 3.1.11.2.1Fingerprinted data

Filtering data is done in order to improve overall accuracy of the fingerprinting technique. E.g. in [54], they applied a Kalman filter [52] during the offline phase so as to improve the positioning accuracy. The Kalman filter, however, has not been considered suitable for filtering this kind of data, e.g. in [56]. They further suggest that a particle filter would be better during the same phase or fingerprinting.

In [57], however, they claim that results with highest accuracy would be obtained if a nearest neighbor filter is applied to the data instead.

#### 3.1.11.2.2 Inertial Sensor Data

Raw sensor readings can yield disastrous results. This is because the data obtained by these sensors contains noise. It is thus necessary to filter the data before usage in the algorithms to avoid errors.

Below is the first-order discrete low-pass filter applied to the sensor data. It is a recursive filter and can easily be implemented (in Java for this project):

$$y(t) = y(t_{-1}) + \mu(x(t) - y(t_{-1}) ) \qquad \textbf{Equation 3.6}$$

s.t: $$\mu = \frac{dt}{TC+dt} \qquad \textbf{Equation 3.7}$$

Where;

$dt$ **:** is the period of sampling the sensor.

$TC$ **:** cutoff frequency or time constant

Doing this yields relatively smoother less noise readings as shown in **Figure 3.6** and **Figure 3.7** below.

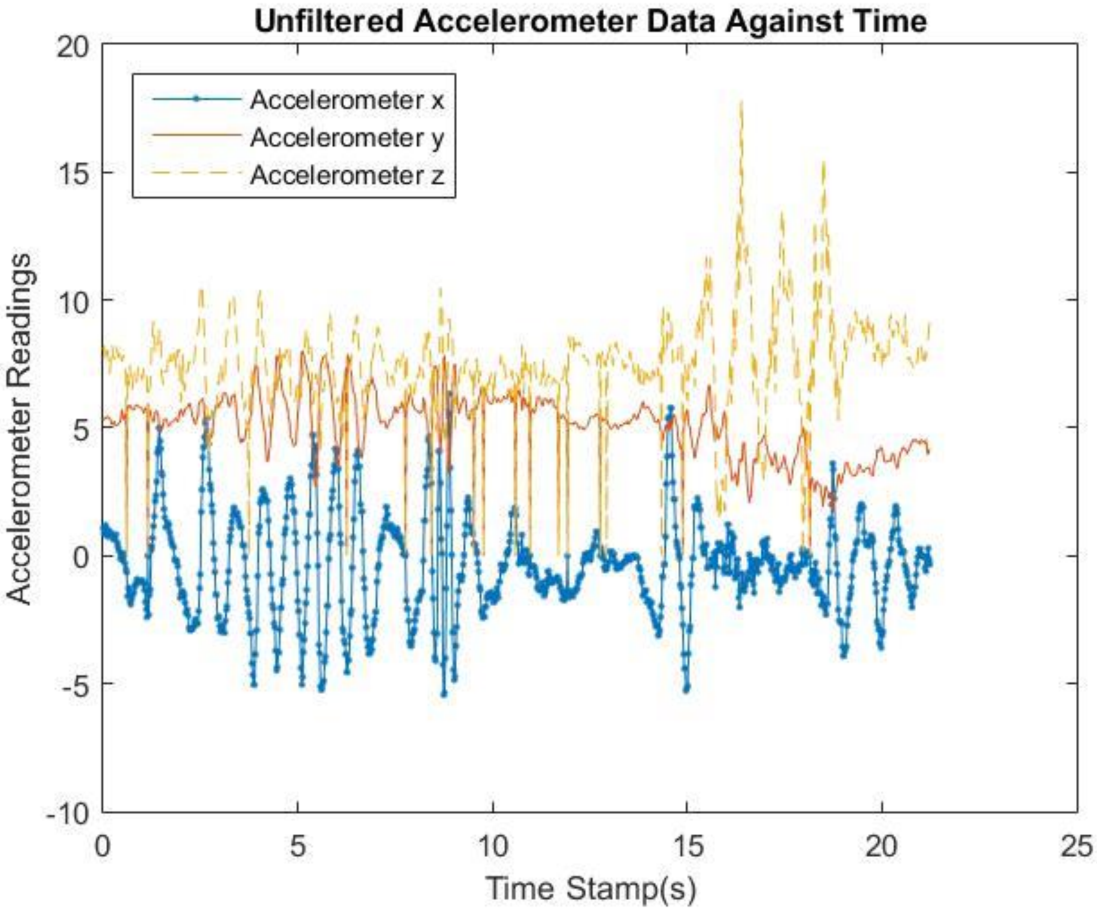


**Figure 3.6** Unfiltered Accelerometer Readings

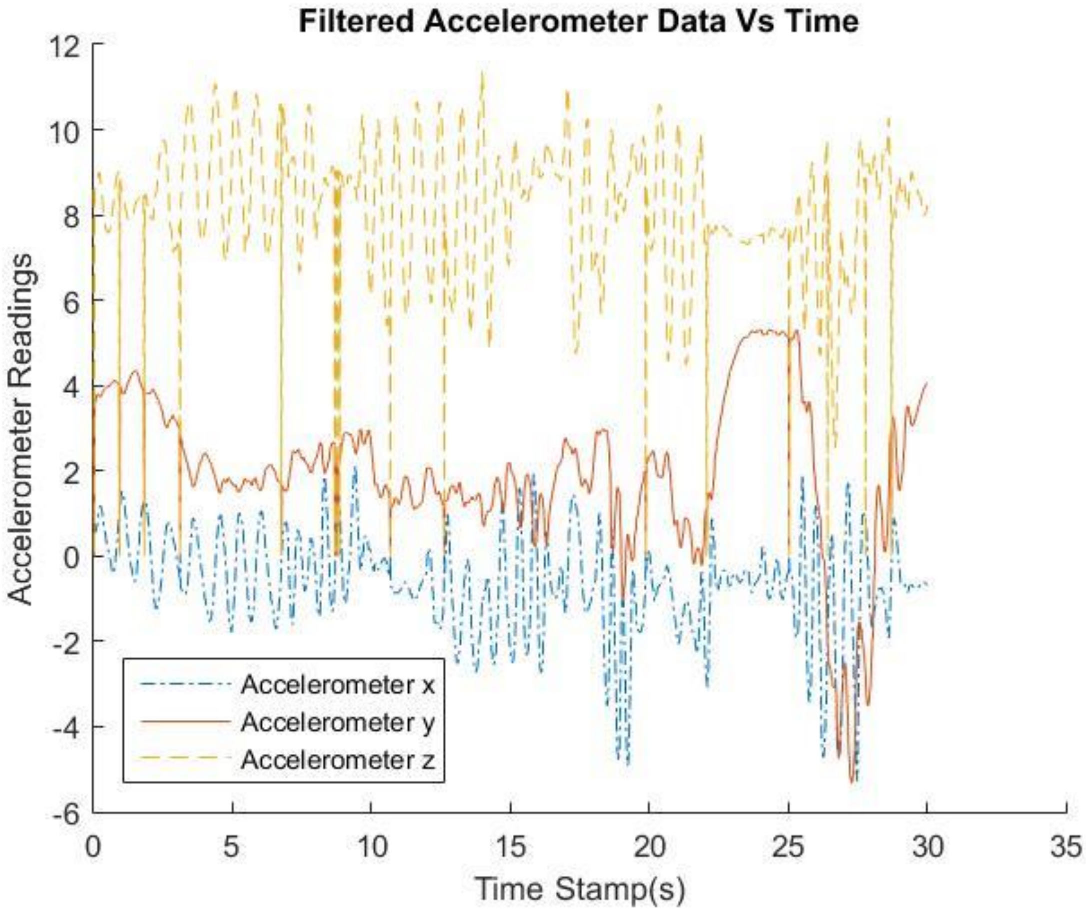


**Figure 3.7** Filtered Accelerometer Readings using a low pass filter

### 3.1.12 Route Confirmation and Context Awareness during Navigation

Indoor navigation can be tricky sometimes as the app user can either easily miss the destination or go off track. It is, thus of paramount importance to let the user know once they are off the right track. As soon as the route has been calculated, the user is constantly notified whether they are on or off the right track. In this project, fingerprinted nodes and the results from dead-reckoning are used to determine whether the user is on or off the right track. If the user is not on the right track, an appropriate notice is issued and the route is re-calculated.

A few of the advantages of the fingerprinting, dead-reckoning hybrid algorithm are briefly described below;

1. If an AP breaks down or is changed/replaced, the fingerprinting algorithm will

give erroneous results. This is compensated for by the dead-reckoning algorithm.

2. RSS varies over time. This could lead to errors in the routing algorithm. In such cases, the dead-reckoning algorithm *backs up* the fingerprinting algorithm. Likewise, dead-reckoning results are prone to error accumulation as stated above. Periodic comparison of results from both algorithms helps address this issue. Since some rooms/nodes use one AP, if there happens to be a temporary significant variation in RSS, the navigator may easily be routed to the wrong location.

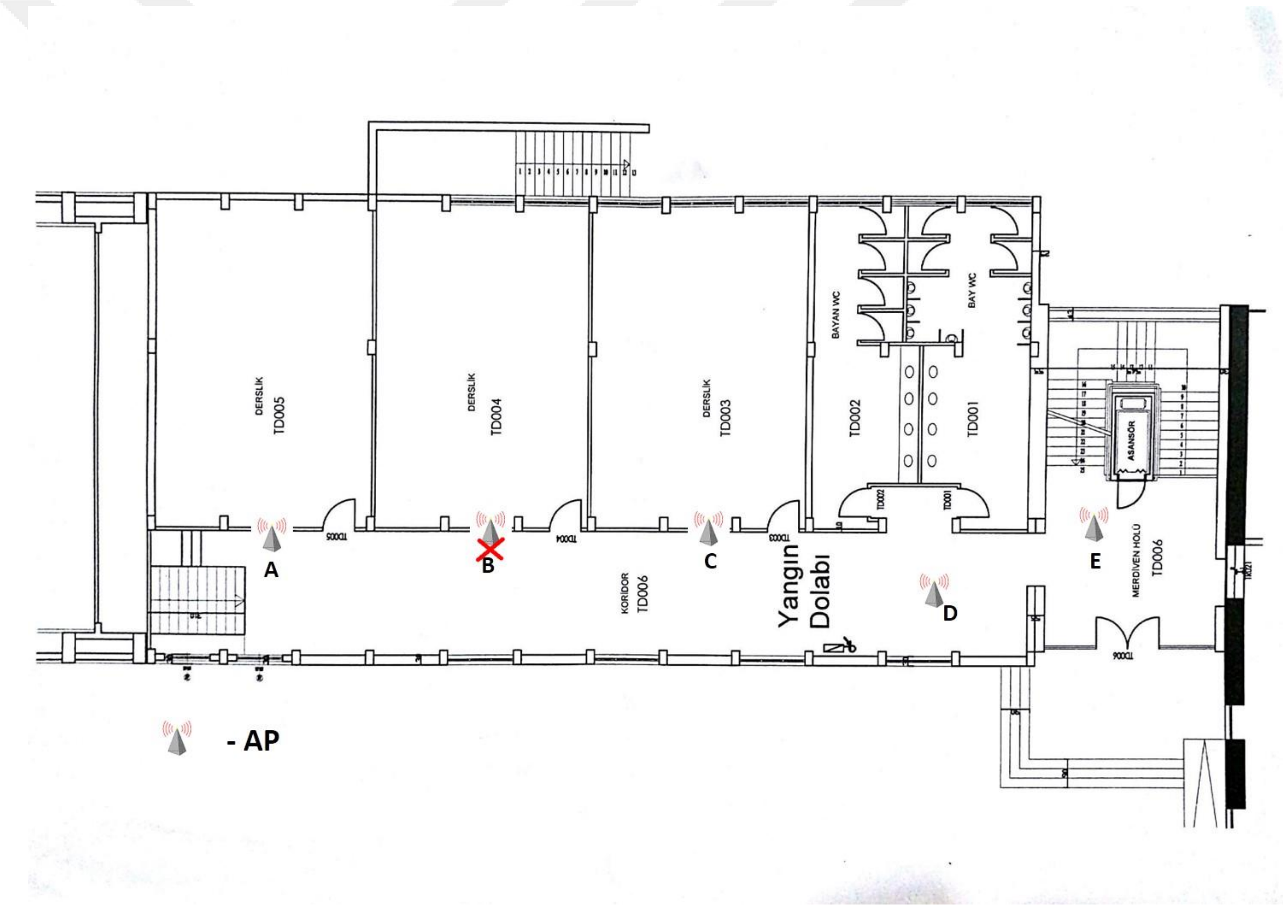


**Figure 3.8** A demonstration of algorithm compensation

For example; assuming the navigator was wishing to move from TD006 with AP *E* to TD005 with AP *A* where *A* and *E* are the base nodes for the WiFi fingerprints stored in the knowledge base, if AP *B* happens to be disabled, or replaced or breakdown (depicted by the red cross in **Figure 3.8** above), the situation described above would arise in which case the results from dead-reckoning would be more than sufficient to seamlessly rectify the issue without troubling the navigator.

## 3.2 Architecture of The Proposed Prototype

In this section, the architecture of the proposed system is presented. It explains how the major components of the entire system are linked to and interact with one another. The system comprises mainly of receivers, transmitters, a knowledge base (database) and a processing unit. The navigator’s smart phone works as a transmitter, receiver, processing unit and also contains the knowledge base. The APs basically are receivers as well. They receive signals transmitted by the smart phone. Then they in turn also transmit data to the smart phones. As mentioned *section 3* above, the architecture set up is in two main phases; the offline and online phases of the Wi-Fi fingerprinting. During the offline phase, reference data is acquired, processed and added to the knowledge base. During the online phase, a user unfamiliar with the environment or wishing to do navigation to a certain location within the environment simply queries the system for a route from a given location to a desired destination. During the online phase, the user given data is filtered to eliminate noise as described in detail in *section 3.7.2*

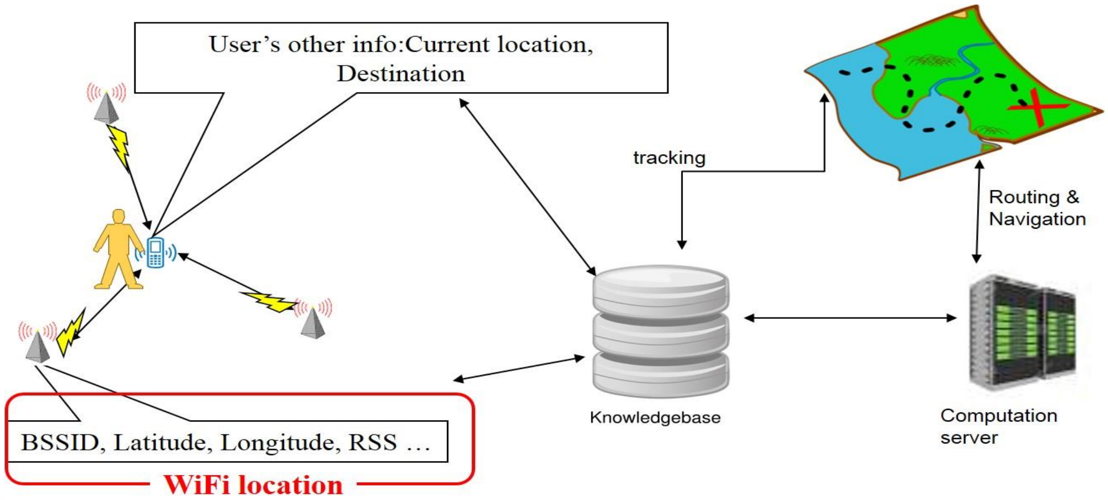

User's other info:Current location, Destination
tracking
Routing & Navigation
Knowledgebase
Computation server
BSSID, Latitude, Longitude, RSS …
WiFi location

**Figure 3.9** The Proposed Model

Generally, as shown in the **Figure 3.9** above, the navigator first of all provides the current position of the location from which they wish to start the navigation guidance to their desired destination and provides the *optimal* route according to the navigator's configurations. Among the options the navigator has are;

- Whether they wish to use stairs or elevators.
- The rate (in seconds) over which the audio directions are given. This is set to 10 seconds by default.
- Whether they are physically disabled …

Then as the navigator moves, the system keeps on periodically calculating and updating the route according to the user's current location in real-time whilst providing notifications whether or not they are on the right track or not.

# 4 EXPERIMENTS AND RESULTS

In this section the experiment tests carried out to test the suggested prototype are explained. Empirical results obtained when testing the system described above are displayed. A Google Nexus 5 using android version 6.0.1 was used throughout the testing procedure at the Istanbul Sabahattin Zaim Campus. These results include the accuracy level, success and the failures with corresponding failure levels.

## 4.1 Tests and results

The following scenarios were tested;

1. User at the very entrance of the floor provided both source and destination and follows instructions from the application in an unbiased way.

2. User provided both source and destination, then followed the instructions while intentionally getting off the suggested route.

3. User provided wrong position as the current location (Check whether and, if so, how fast the program will update the current location).

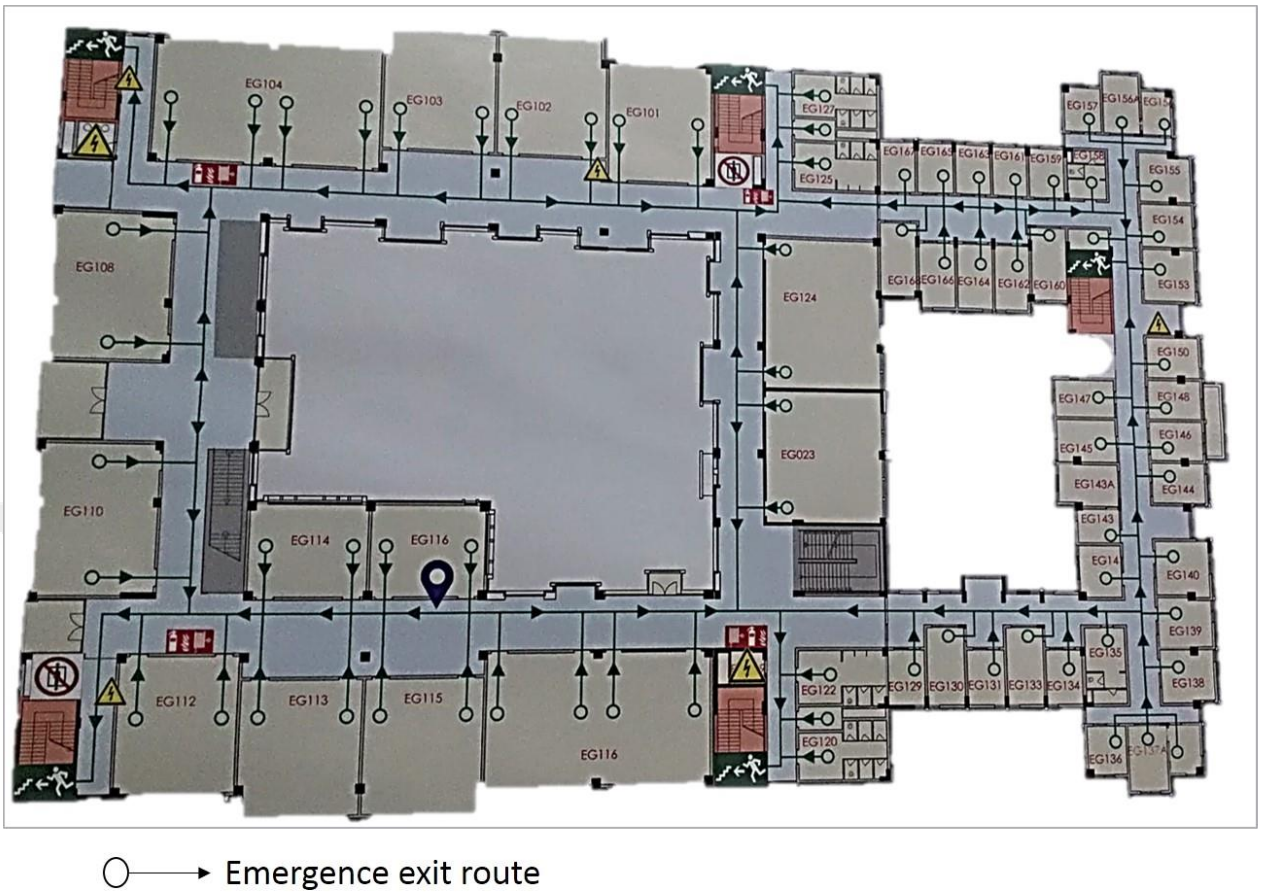


**Figure 4.1** A simplified version of the used map in the project prototype.

The map in **Figure 4.1** above is of the first floor of the education block. The left wing of the map will be used for testing the application.

**Table 4.1** Sample summarized results

| Current Location | Input Location | Destination | Time to calculate route(ms) | Average Update Time If required(ms) | Success | Failure (Deviation in meters) |
|---|---|---|---|---|---|---|
| EG123 | EG123 | EG110 | 16 | Null | 5 | 0 |
| EG110 | EG110 | EG104 | 27 | Null | 4 | 0 |
| EG110 | EG110 | EG108 | 5 | Null | 1 | 1(~1.2m) |
| EG113 | EG113 | EG101 | 27 | Null | 6 | 2(~≤1.33m, *0.4m*) |
| EG124 | EG117 | EG104 | 20 | 46 | 5 | 1(≤*1.5m*) |
| EG104 | EG101 | EG122 | 25 | 37 | 7 | 1(≤*1.28m*) |

**Table 4.1** Summarized Results

Some results denote failures, these errors are obtained when the project prototype gives a wrong node at a given point/node, say if Class *EG117* is shown as *EG116*. Every correctly identified significant node on the path calculated and presented to the navigator is considered as success. The values in the brackets show how far off the yielded results actually are from the intended location. Using the analogy above, this corresponds to how far the user was taken from *EG117*. As for the case of the route from *EG110 – EG108*, it is because both room *EG110* and room *EG108* share the same node and were assigned the same node value. Hence an empty list of results is returned since it was, for test purposes, was configured to do so. The same behavior will also be exhibited in case we try to navigate towards a disconnected node in a disconnected graph. There are so many choices for handling these situations, in this case since the two rooms are right adjacent to one another so we chose to simply let the user know that they have already arrived at their desired destination and add a direction they should turn to depending on their orientation. Given that all the failures are with ≤*2m* range or less – with *2m* being the *upperbound*. The descriptive navigation instructions accurately and very efficiently rectify this offset since all the intended targets are within eye sight so considering the whole system has an average accuracy of about ***0.8-1.3m***. This is sufficiently accurate for indoor building even those with small rooms since room level (~*2m*) accuracy is just sufficient. The average of ~*0.8m* was observed for standard sized rooms (~$4m^3$) such as student laboratories or even larger whilst that of *1.3m* was observed for relatively small rooms (~$2.1m^3$) for example, the rest rooms.

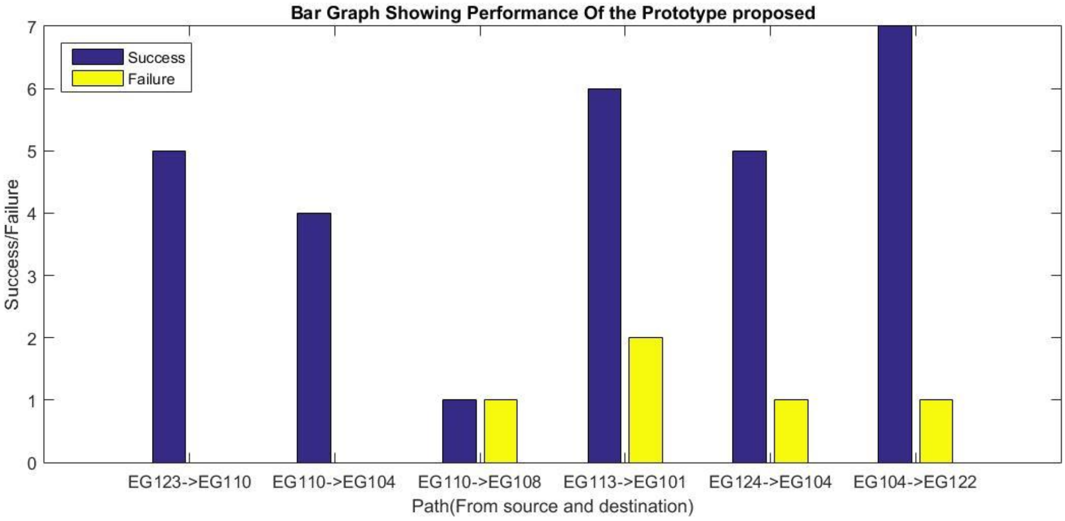


**Figure 4.2** Bar graph reflecting the performance of the suggested prototype.

**Figure 4.2** Above shows the performance of the proposed prototype. The overall accuracy of the system, let alone the fast response time, is sufficiently accurate at room level – which is approximately less than (in most cases) or equasl to *2m* (~≤ *2m*). This accuracy, however, drops drastically if there happens to be a disconnected node in the graph. Hence paramount attention must be paid whilst creating the graph. The graph should not be disconnected. The effects of a disconnected graph on the performance and accuracy of the proposed prototype are demonstrated in **Figure 5.3** and **Figure 5.4** below.

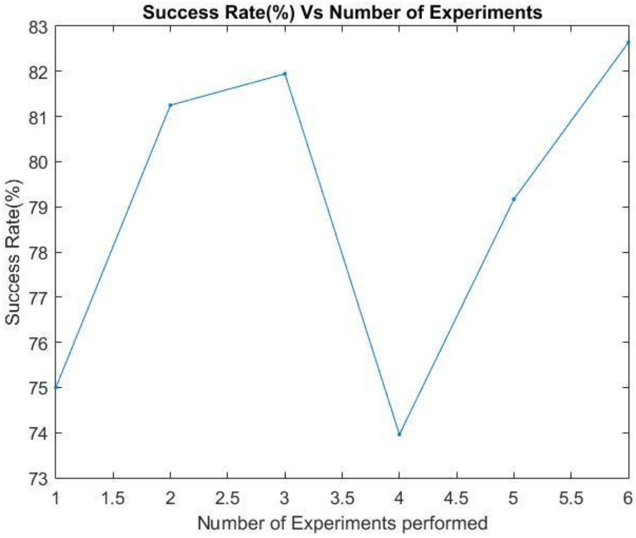


**Figure 4.3** Average success rate of a disconnected graph

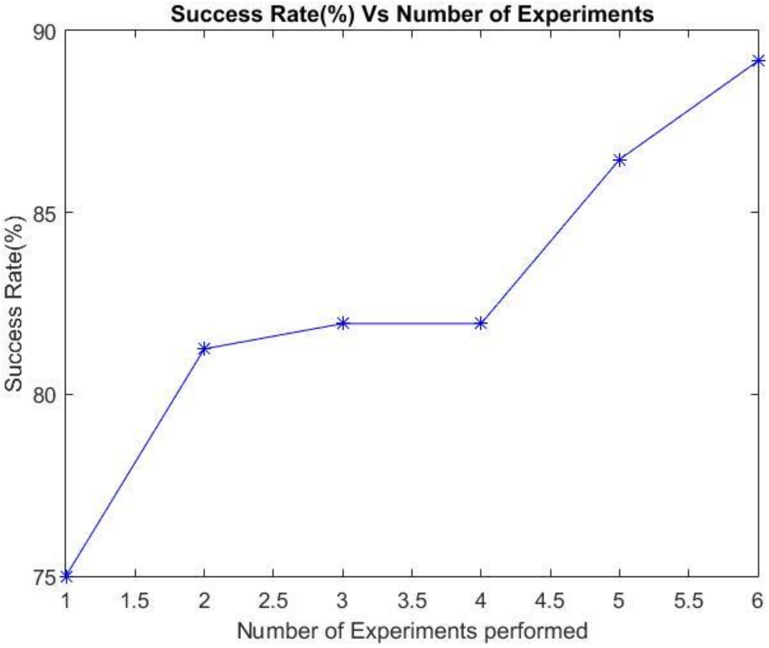


**Figure 4.4** Average success rate of a connected graph.

In the disconnected graph shown in **Figure 4.3** Above, a *single* class room was not connected to any other room/node in the graph whilst in **Figure 4.4** This node was not put into consideration.

The above results demonstrate the general performance of ALCIPIPF. Extensive tests were performed and the average accuracy grows linearly with the number of tests performed as shown in **Figure 4.4** above. Furthermore, the prototype scales with building room or node number size. Latency or calculation delays do not get worse with increasing number of node points. This is clearer shown in **Figure 4.5** below:

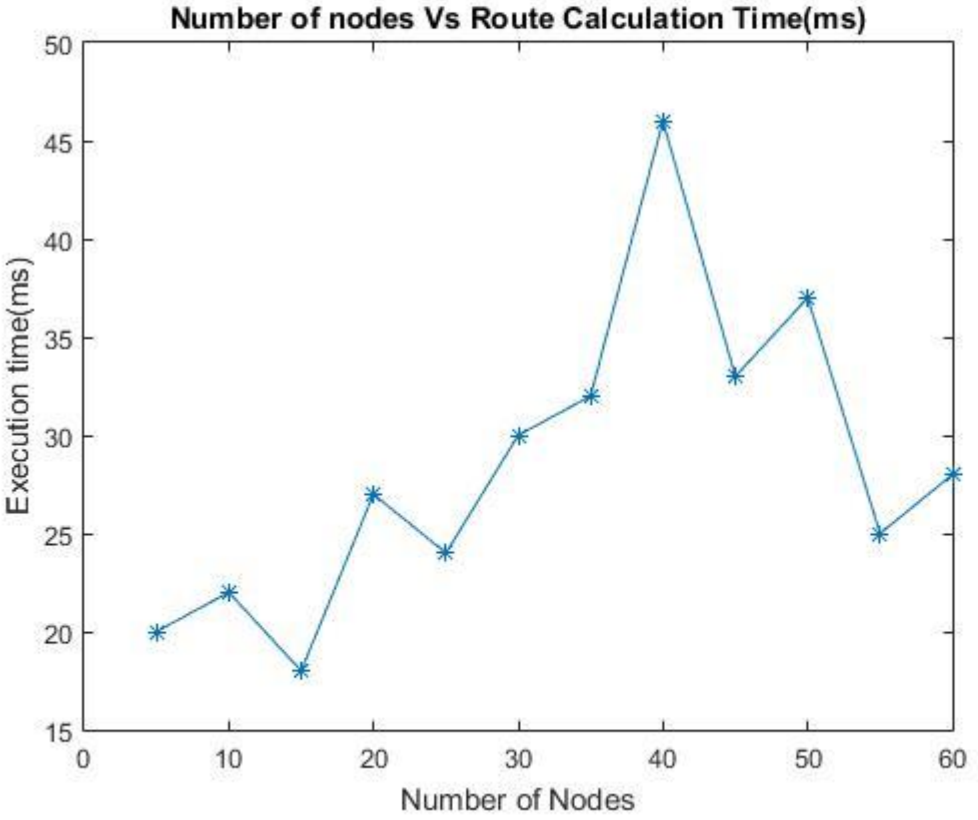


**Figure 4.5** Number of nodes Against Route Calculation time (ms)

**Figure 4.5** above shows the scalability of the suggested prototype. Basically, the number of nodes in the map does not have a significant negative effect on the time take to calculate a route.

## 4.2 General Comparison with trending approaches

In this section, we briefly show how ALCIPIPF fairs with other trending approaches from related works performance wise. On top of providing an optimal route and navigation services for users – including the visually impaired and the physically disabled, ALCIPIPF performs rather well performance wise as shown in below table.

| **Method / Approach** | **Average Accuracy (m)** |
|---|---|
| KNN | 4.37 |
| Probabilistic | 2.78 |
| Fingerprinting | 2-3 |
| Probabilistic and particle filter | 1.96 |
| ALPCIPIPF | 0.8 – 1.3 |

**Table 4.2** General Performance Comparison

# 5 CONCLUSION, FUTURE WORKS AND POSSIBLE IMPROVEMENTS

## 5.1 Future works and Possible Improvements

Satisfying as the results may be, there is still a lot of room for improvement and development. Both as far as performance and providing a much better user experience are concerned. With continuous growth in ubiquitous computing and the performance of mobile devices, even greater room for development is presented.

This section covers such areas that can be improved or enhanced are presented.

i. The system is likely to suffer serious accuracy issues if any AP is taken down, removed or changed especially, assuming the node(s) associated with the AP have got not multiple AP access but single or two AP access as described in [26], yet they are connected to many other nodes. Worst case scenario is if the other APs are also experiencing non insignificant RSS variations. This would throw both the fingerprinting and the DR algorithms off track. This means that periodic checkup of the APs would be in our best interest or constant communication with the ones responsible with AP setup in the facility in case of any changes.

   Furthermore, environmental changes such as infrastructural renovations also have got an impact on the accuracy, but not the performance of the application.

   We therefore plan on implementing automatic AP update or leave it up as a significant future work.

ii. This prototype was designed basing mainly on only one building or infrastructure – from the building/infrastructure's management's point of view (say a university campus such as ours). However, we would also like to think from the users'/navigators' point of view. If the users install this software from a given building, it also implies that they would have to re-install the software when they go to another infrastructure (again say, university) that uses the same software. Hence with a little finance or support we would like to set the system up such that users install the software once and can use the finished project in any place that uses the same software.

iii. Seamless integration with google maps so that the navigator seamlessly travels not only from building block to building block, but can also easily and comfortably use the app outdoors.

iv. Multi-language support. In this prototype, due to brevity, only the English language is supported. However, in the later versions, a wider range of language choices for example French, Turkish, Arabic, Spanish, to mention but a few, will be implemented. A more *locale* language based implementation is not a big challenge.

v. We plan on giving the user a choice of the route/path representation in the future. A 3D/2D dynamic map such the one used by Google maps [45] or the textual representation as shown here.

## 5.2 Conclusion

In this study, a cost efficient user interactive indoor navigation system using Wi-Fi fingerprinting and sensor fusion that requires no installation of extra hardware or third-party software is proposed. The system offers optimal navigation to the user with an upper bound of ***2m*** and an **average** of ***0.8-1.3m*** accuracy. The proposed prototype is also suitable for the visually impaired and physically challenged as it enables the user to choose the kind of configuration they prefer. In any case, the most efficient route is always suggested and presented to the navigator. The system implementation is a hybrid of Wi-Fi fingerprinting and dead reckoning performed using resultant data from the application of a filter on sensor data so as to reduce noise. Prototype test cases were carried out on the Education block of IZU campus. This prototype is of an indoor navigation system that runs on a user's smart phone device and requires nothing else.

The suggested prototype is suitable for navigation and emergence evacuation for both normal, visually impaired and physically challenged users. Audio support is included into the system so users are able to get voice directions as they navigate throughout an unfamiliar indoor environment.

# References

[1] R. Want, A. Hopper, V. Falc˜ao, and J. Gibbons. The active badge location system. ACM Transactions on Information Systems (TOIS), Jan 1992.

[2] M. Weiser. The computer for the 21 stcentury. ACM SIGMOBILE Mobile Computing and Communications Review, Jan 1999.

[3] Butz, A., and Baus, J., and Krüger, A., and Lohse, M.A Hybrid Indoor Navigation System. IUI2001:International Conference on Intelligent User Interfaces, ACM, New York, 25-33.

[4] Raper, J., Gartner, G., Karimi, H. and Rizos, C., 2007. A critical evaluation of location based services and their potential. Journal of Location Based Services, 1(1), pp.5-45.

[5] The Ekahau Positioning System Homepage, 12/03/06, http://www.ekahau.com.

[6] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services, MobiSys ’12, pages 169–182, New York, NY, USA, 2012. ACM.

[7] N Priyantha, A Chakraborty and H Balakrishnan, “The Cricket Location Support System”, Proc. Of the Sixth Annual ACM International Comference on Mobile Computing and Networking( MOBICOM), Aug. 2000.

[8] D. Gusenbauer, C. Isert, and J. Kr¨ osche. Self-contained indoor positioning on off-the-shelf mobile devices. In Indoor Positioning and Indoor Navigation

(IPIN), 2010 International Conference on, pages 1–9, 2010.

[9] J. Link, P. Smith, N. Viol, and K. Wehrle. Footpath: Accurate map-based indoor navigation using smartphones. In Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on, pages 1–8, 2011.

[10] Trein, G., Singh, N. and Maddila, P., Simple approach for indoor mapping using low-cost accelerometer and gyroscope sensors.

[11] Miu, A., 2002. Design and Implementation of an Indoor Mobile Navigation System. Master Thesis of MIT.

[12] Alison K. Brown, "Urban/Indoor Navigation using Network Assisted GPS", ION 61st Annual Meeting, Cambridge, MA, June 2005.

[13] Kim O'Neil, "Galileo-European Satellite Navigation System", http://www.aatl.net/publications/galileo.htm, Sep. 2001.

[14] T. Mori, Y. Suemasu, and H. Noguchi, "Multiple people tracking by integrating distributed floor pressure sensors and RFID system", 2004 IEEE International Conference on Systems, Man and Cybermetics, Volume 6, 10-13 Oct. 2004, pp. 5271-5278.

[15] J. Saarinen, H.Suomela, M. Elomaa, and A. Halme, "Personal navigation system", in Proceedings of 2004 IEE/RSJ International on intelligent Robots and systems, 2004, pp. 212-21.

[16] N. Ravi, P. Shankar, A. Frankel, "Indoor Localization Using Camera Phones", www.research rutgers.edu/ ~spravin/papers/camera-phone.pdf, 2005.

[17] Android Studio IDE, "https://developer.android.com/studio/index.html"

[18] Höllerer, T., Hallaway, D., Tinna, N. and Feiner, S., 2001, August. Steps toward accommodating variable position tracking accuracy in a mobile augmented reality system. In 2nd International Workshop on Artificial Intelligence in Mobile Systems (AIMS'01) (pp. 31-37).

[19] “Bing Venue Maps”, http://www.bing.com/maps/?vmb=1.

[20] “Google Indoor Maps Availability”,http://support.google.com/gmm/bin/answer py?hl=en&answer=1685827.

[21] “Nokia, Samsung, Sony, Qualcomm, Others Team Up for Indoor Positioning”, http://www.xbitlabs.com/news/mobile/display/20120824002723 Nokia Samsung Sony Qualcomm Others Team Up for Indoor Positioning.html

[22] I. Park, G. U. Jang, S. Park, J. Lee (2009) Time-Dependent Optimal Routing in Micro-scale Emergency Situation. 10th Int. Conf. on Mobile Data Management: Systems, Services and Middleware, pages 714-719

[23] M. Brunato and K. Csaba,”Transparent location fingerprinting for wireless services”, Proc. Med-Hoc-Net, 2002.

[24] Stook, J.W., 2012. Planning an indoor navigation service for a smartphone with Wi-Fi fingerprinting localization (Master's thesis).

[25] Kannan, B., Meneguzzi, F., Dias, M.B. and Sycara, K., 2013, March. Predictive indoor navigation using commercial smart-phones. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (pp. 519-525). ACM.

[26] Yuan, W. and Schneider, M., 2010. iNav: An indoor navigation model supporting length-dependent optimal routing. Geospatial thinking, pp.299-313.

[27] Baus, J., Krueger, A. and Wahlster, W., 2002. A resource-adaptive mobile navigation system. In:Proceedings of the 7th International Conference on Intelligent user Interfaces, San Francisco, USA, ACM Press, pp. 15-22.

[28] Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R. and Pinkerton, M., 1997. CyberGuide: A Mobile Context-Aware Tour. Wireless Networks, 3(5), pp. 421-433.

[30] Leppäkoski, H., Collin, J. and Takala, J., 2013. Pedestrian navigation based on inertial sensors, indoor map, and WLAN signals. Journal of Signal Processing

Systems, 71(3), pp.287-296.

[31] VTI Technologies, “SCC1300-D04 combined gyroscope and 3-axis accelerometer with digital SPI interfaces,” rev. 1.0.3. Doc.Nr. 82 1131 00A, Data sheet, 2010.

[32] Hohl, F., and Kubach, U., and Leonhardi, A., and Rothermel, K., and Schwehm, M. Next century challenges : Nexus – an open global infrastructure for spatial aware applications. International Conference on Mobile Computing and Networking (MobiCom ’99), ACM Press, 249-255.

[33] Kalra, N. and Bansal, D., 2014. Analyzing driver behavior using smartphone sensors: a survey. Int. J. Electron. Electr. Eng, 7(7), pp.697-702.

[34] Gyroscope, http://en.wikipedia.org/wiki/Gyroscope

[35] Magnetometer, http://en.wikipedia.org/wiki/Magnetometer

[36] Diestel, R. (2005). "Graph Theory, Electronic Edition". p. 12.

[37] E. Dijkstra. A note on two problems in connexion with graphs. Numerische Mathematik, 1:269–271, 1959.

[38] Dijkstra's Algorithm, https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

[39] Fallah, N., Apostolopoulos, I., Bekris, K. and Folmer, E., 2013. Indoor human navigation systems: A survey. Interacting with Computers, 25(1), pp.21-33.

[40] J. Link, P. Smith, N. Viol, and K. Wehrle. Footpath:Accurate map-based indoor navigation using smartphones. In Indoor Positioning and Indoor Navigation (IPIN), 2011 International Conference on, pages 1–8, 2011.

[41] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” in Proceedings of 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’00), vol. 2, pp. 775–784, Tel Aviv.Israel, March 2000.

[42] Y. Jin, H.-S. Toh, W.-S. Soh, and W.-C. Wong, “A robust dead-reckoning pedestrian tracking system with low cost sensors,” in Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on, march 2011.

[43] Errors associated with Dead-Reckoing, https://en.wikipedia.org/wiki/Dead_reckoning#Errors

[44] Huang, H. and Gartner, G., 2009. A survey of mobile indoor navigation systems. In Cartography in Central and Eastern Europe (pp. 305-319). Springer Berlin Heidelberg.

[45] Google maps, https://en.wikipedia.org/wiki/Google_Maps

[46] Lertlakkhanakul, J., Li, Y., Choi, J. and Bu, S., 2009, August. GongPath: Development of BIM based indoor pedestrian navigation system. In INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on (pp. 382-388). IEEE.

[47] Gu, Y., Lo, A. and Niemegeers, I., 2009. A survey of indoor positioning systems for wireless personal networks. IEEE Communications surveys & tutorials, 11(1), pp.13-32.

[48] Woo, S., Jeong, S., Mok, E., Xia, L., Choi, C., Pyeon, M. and Heo, J., 2011. Application of WiFi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR. Automation in Construction, 20(1), pp.3-13.

[49] Zhang, D., Xia, F., Yang, Z., Yao, L. and Zhao, W., 2010, May. Localization technologies for indoor human tracking. In Future Information Technology (FutureTech), 2010 5th International Conference on (pp. 1-6). IEEE.

[50] Yang, J. and Chen, Y., 2009, November. Indoor localization using improved RSS-based lateration methods. In Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE (pp. 1-6). IEEE.

[51] Kotaru, M., Joshi, K., Bharadia, D. and Katti, S., 2015, August. Spotfi: Decimeter level localization using wifi. In ACM SIGCOMM Computer Communication Review (Vol. 45, No. 4, pp. 269-282). ACM.

[52] Kalman Filter, https://en.wikipedia.org/wiki/Kalman_filter

[53] Lim, H. L. Kung, J.C. Hou & H. Luo (2010), Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure. Wireless Networks 16, pp. 405-420.

[54] Yim, J. (2008), Introducing a decision tree-based indoor positioning technique. Expert Systems with Applications 34, pp. 1296-1302.

[55] Youssef, M.A., Agrawala, A. and Shankar, A.U., 2003, March. WLAN location determination via clustering and probability distributions. In Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on (pp. 143-150). IEEE.

[56] Development of an Indoor Location Based Service Test Bed and Geographic Information System with a Wireless Sensor Network

[57] Dawes, B. and Chin, K.W., 2011. A comparison of deterministic and probabilistic methods for indoor localization. Journal of Systems and Software, 84(3), pp.442-451.

[58] RSSI and dBm, http://www.metageek.com/training/resources/understanding-rssi.html

[59] Goodall, D.S., Ryan, P.J., Myles, A.F. and De Souza, L.N., Cisco Technology, Inc., 2005. Selecting an access point according to a measure of received signal quality. U.S. Patent 6,940,843.

[60] Nexus 5, https://www.google.com/nexus/5x/

[61] Java Standard Edition Version 8, https://docs.oracle.com/javase/8/docs/technotes/guides/language/enhancements.

html#javase8

[62] Android Marshmallow, https://www.android.com/versions/marshmallow-6-0/

[63] Stonebraker, M., 2010. SQL databases v. NoSQL databases. Communications of the ACM, 53(4), pp.10-11.

[64] Silva, L.G.D. and Becker, M., 2013. System Based On An Android Platform As A Sensor To Assist The Autonomous Localization.

[10] [65] G. Stein, A. Kling, M. Gottshall, M.Shanblatt, 2011, Indoor Directional Orientation Communication and Enabling Navigational Technology.

[66] Fischer, C., Muthukrishnan, K., Hazas, M. and Gellersen, H., 2008, September. Ultrasound-aided pedestrian dead reckoning for indoor navigation. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments* (pp. 31-36). ACM.

[67] Fallah, N., Apostolopoulos, I., Bekris, K. and Folmer, E., 2012, May. The user as a sensor: navigating users with visual impairments in indoor spaces using tactile landmarks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 425-432). ACM.

[68] Zhang, P., Zhao, Q., Li, Y., Niu, X., Zhuang, Y. and Liu, J., 2015. Collaborative WiFi fingerprinting using sensor-based navigation on smartphones. Sensors, 15(7), pp.17534-17557.Vancouver.

[69] Wu, H., Marshall, A. and Yu, W., 2007, July. Path planning and following algorithms in an indoor navigation model for visually impaired. In *Internet Monitoring and Protection, 2007. ICIMP 2007. Second International Conference on* (pp. 38-38). IEEE.

[70] Hansen, R., Thomsen, B., Thomsen, L.L. and Adamsen, F.S., 2013, June. SmartCampusAAU--An Open Platform Enabling Indoor Positioning and Navigation. In *Mobile Data Management (MDM), 2013 IEEE 14th International Conference on* (Vol. 2, pp. 33-38). IEEE.

[71] Bolliger, P., 2008, September. Redpin-adaptive, zero-configuration indoor localization through user collaboration. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments* (pp. 55-60). ACM.

[72] Golding, A.R. and Lesh, N., 1999, October. Indoor navigation using a diverse set of cheap, wearable sensors. In *Wearable Computers, 1999. Digest of Papers. The Third International Symposium on* (pp. 29-36). IEEE.

[73] Huang, H., Gartner, G., Schmidt, M. and Li, Y., 2009, June. Smart environment for ubiquitous indoor navigation. In *New Trends in Information and Service Science, 2009. NISS'09. International Conference on* (pp. 176-180). IEEE.

[74] Hub, A., Diepstraten, J. and Ertl, T., 2004, October. Design and development of an indoor navigation and object identification system for the blind. In *ACM Sigaccess Accessibility and Computing* (No. 77-78, pp. 147-152). ACM.

[75] Panyov, A.A., Golovan, A.A. and Smirnov, A.S., 2014, February. Indoor positioning using Wi-Fi fingerprinting pedestrian dead reckoning and aided INS. In *Inertial Sensors and Systems (ISISS), 2014 International Symposium on* (pp. 1-2). IEEE.

[76] Rajamäki, J., Viinikainen, P., Tuomisto, J., Sederholm, T. and Säämänen, M., 2007, February. LaureaPOP indoor navigation service for the visually impaired in a WLAN environment. In *Proceedings of the 6th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications* (pp. 96-101). World Scientific and Engineering Academy and Society (WSEAS).

[77] Ran, L., Helal, S. and Moore, S., 2004, March. Drishti: an integrated indoor/outdoor blind navigation system and service. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on* (pp. 23-30). IEEE.

[78] Retscher, G., 2004. Pedestrian navigation systems and location-based services.

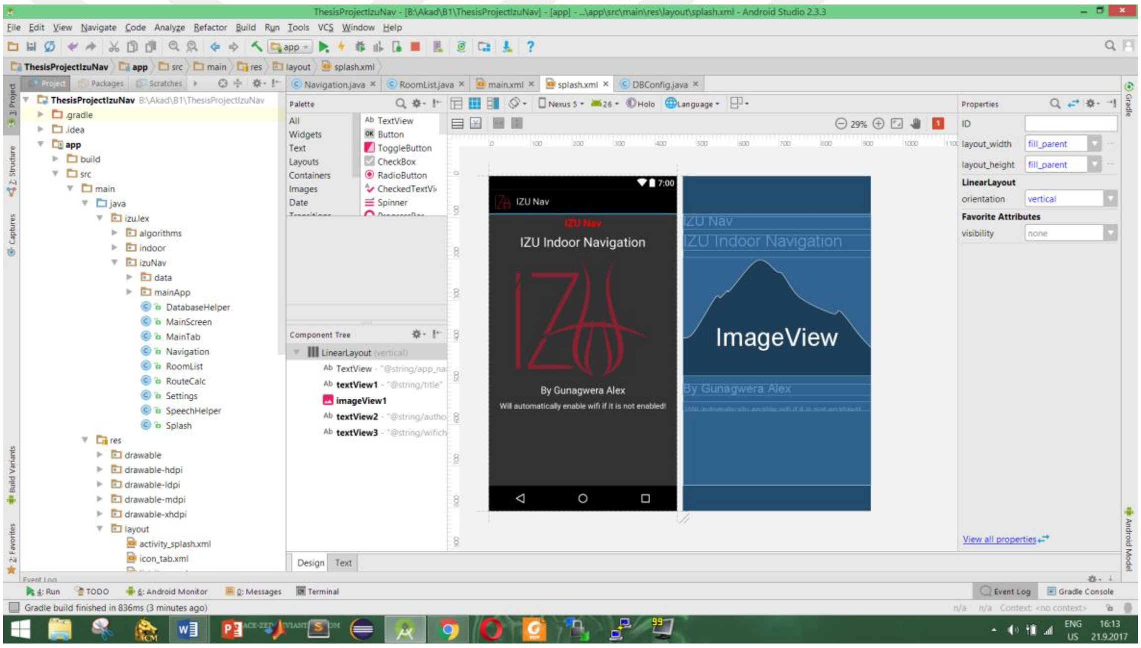
[79] Retscher, G. and Thienelt, M., 2004. NAVIO–A navigation and guidance service for pedestrians. *Positioning*, *1*(08), p.0.

[80] Kaluwahandi, S. and Tadokoro, Y., 2001. Portable traveling support system using image processing for the visually impaired. In *Image Processing, 2001. Proceedings. 2001 International Conference on* (Vol. 1, pp. 337-340). IEEE.

[81] Nakamura, K., Aono, Y. and Tadokoro, Y., 1997. A walking navigation system for the blind. *Systems and computers in Japan*, *28*(13), pp.36-45.

# 6 Appendix 1

In this section sample screen shots from the project are presented. These were taken during the prototype development and/or the prototype testing after development.



ThesisProjectIzuNav - [B:\Akad\B1\ThesisProjectIzuNav] - [app] - ...\app\src\main\res\layout\splash.xml - Android Studio 2.3.3

ID

layout_width: fill_parent

layout_height: fill_parent

LinearLayout

orientation: vertical

Favorite Attributes

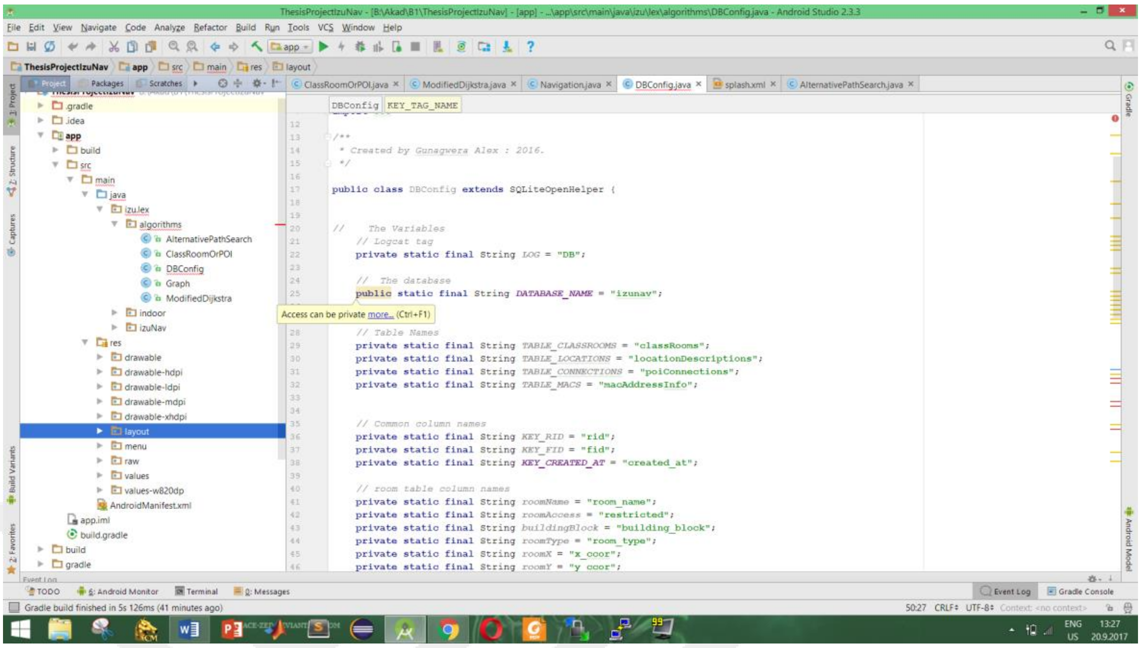
visibility: none

Gradle build finished in 836ms (3 minutes ago)

n/a n/a Context: <no context>

**Figure 6.1** Project Development Environment - Android Studio

This prototype was developed using the Android Studio IDE [17]. The IDE shown in **Figure 6.1** above is opensource and readily available.

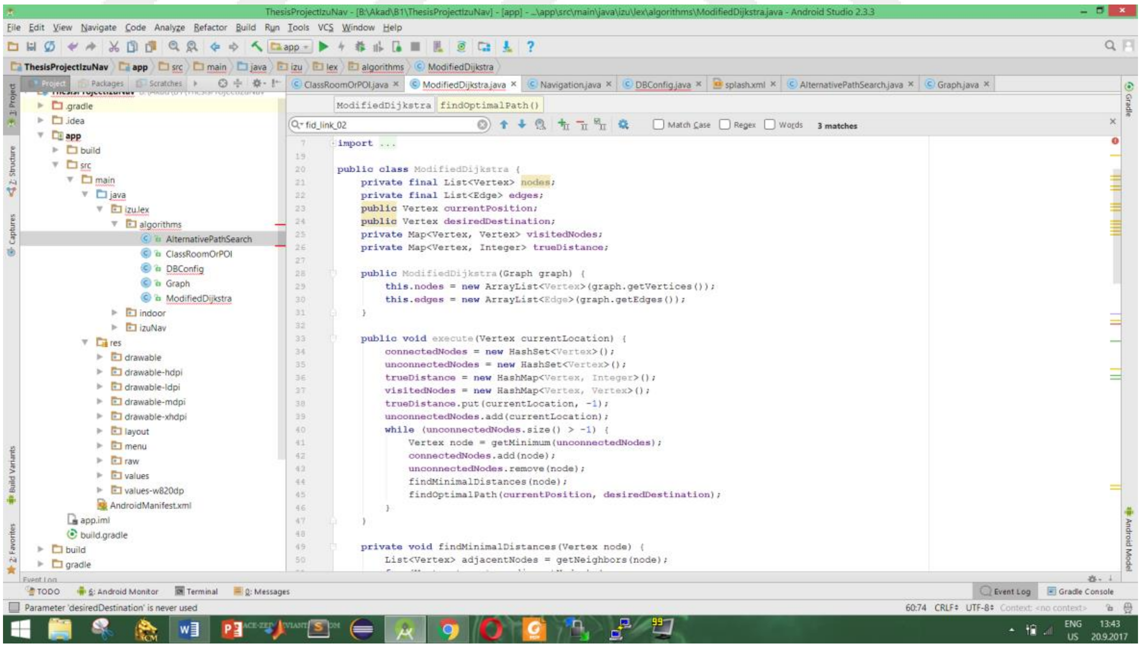


ThesisProjectIzuNav - [B:\Akad\B1\ThesisProjectIzuNav] - [app] - ...\app\src\main\java\izu\lex\algorithms\DBConfig.java - Android Studio 2.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ClassRoomOrPOI.java
ModifiedDijkstra.java
Navigation.java
DBConfig.java
splash.xml
AlternativePathSearch.java
DBConfig
KEY_TAG_NAME
Access can be private more... (Ctrl+F1)

```java
/**
 * Created by Gunagwera Alex : 2016.
 */

public class DBConfig extends SQLiteOpenHelper {


//    The Variables
    // Logcat tag
    private static final String LOG = "DB";

    //  The database
    public static final String DATABASE_NAME = "izunav";

    // Table Names
    private static final String TABLE_CLASSROOMS = "classRooms";
    private static final String TABLE_LOCATIONS = "locationDescriptions";
    private static final String TABLE_CONNECTIONS = "poiConnections";
    private static final String TABLE_MACS = "macAddressInfo";


    // Common column names
    private static final String KEY_RID = "rid";
    private static final String KEY_FID = "fid";
    private static final String KEY_CREATED_AT = "created_at";

    // room table column names
    private static final String roomName = "room_name";
    private static final String roomAccess = "restricted";
    private static final String buildingBlock = "building_block";
    private static final String roomType = "room_type";
    private static final String roomX = "x_coor";
    private static final String roomY = "y_coor";
```

Gradle build finished in 5s 126ms (41 minutes ago)

**Figure 6.2** Project Database Helper - Initialiser Class



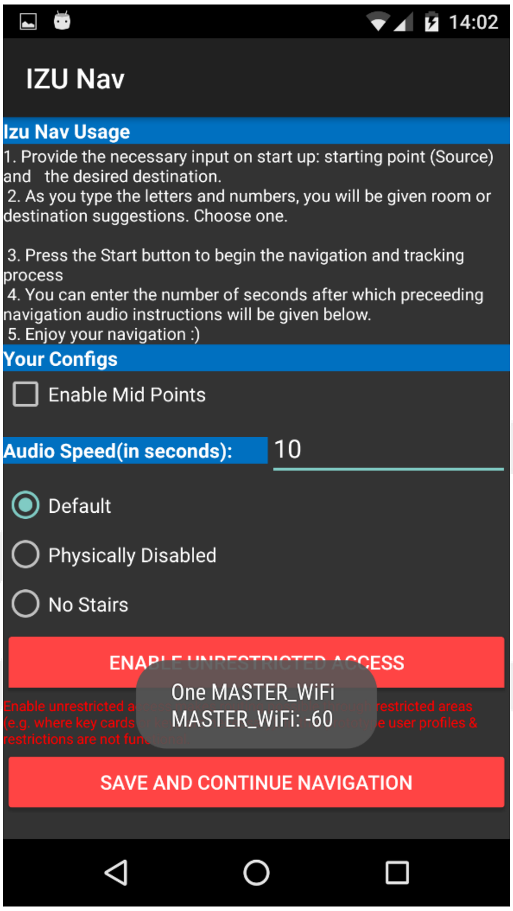
ThesisProjectIzuNav - [B:\Akad\B1\ThesisProjectIzuNav] - [app] - ...\app\src\main\java\izu\lex\algorithms\ModifiedDijkstra.java - Android Studio 2.3.3
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ClassRoomOrPOI.java
ModifiedDijkstra.java
Navigation.java
DBConfig.java
splash.xml
AlternativePathSearch.java
Graph.java
ModifiedDijkstra
findOptimalPath()
fid_link_02
□ Match Case □ Regex □ Words 3 matches

```java
import ...

public class ModifiedDijkstra {
    private final List<Vertex> nodes;
    private final List<Edge> edges;
    public Vertex currentPosition;
    public Vertex desiredDestination;
    private Map<Vertex, Vertex> visitedNodes;
    private Map<Vertex, Integer> trueDistance;

    public ModifiedDijkstra(Graph graph) {
        this.nodes = new ArrayList<Vertex>(graph.getVertices());
        this.edges = new ArrayList<Edge>(graph.getEdges());
    }

    public void execute(Vertex currentLocation) {
        connectedNodes = new HashSet<Vertex>();
        unconnectedNodes = new HashSet<Vertex>();
        trueDistance = new HashMap<Vertex, Integer>();
        visitedNodes = new HashMap<Vertex, Vertex>();
        trueDistance.put(currentLocation, -1);
        unconnectedNodes.add(currentLocation);
        while (unconnectedNodes.size() > -1) {
            Vertex node = getMinimum(unconnectedNodes);
            connectedNodes.add(node);
            unconnectedNodes.remove(node);
            findMinimalDistances(node);
            findOptimalPath(currentPosition, desiredDestination);
        }
    }

    private void findMinimalDistances(Vertex node) {
        List<Vertex> adjacentNodes = getNeighbors(node);
```

Parameter 'desiredDestination' is never used

**Figure 6.3** Find Optimal Route/Path for user

**Figures 6.2** and **6.3** above show parts of the code used in the implementation of the prototype software. It is for the Android platform and hence mainly developed using the Java programming language.



14:02

IZU Nav

Izu Nav Usage

1. Provide the necessary input on start up: starting point (Source) and the desired destination.
2. As you type the letters and numbers, you will be given room or destination suggestions. Choose one.

3. Press the Start button to begin the navigation and tracking process
4. You can enter the number of seconds after which preceeding navigation audio instructions will be given below.
5. Enjoy your navigation :)

Your Configs

□ Enable Mid Points

Audio Speed(in seconds): 10

● Default

○ Physically Disabled

○ No Stairs

ENABLE UNRESTRICTED ACCESS

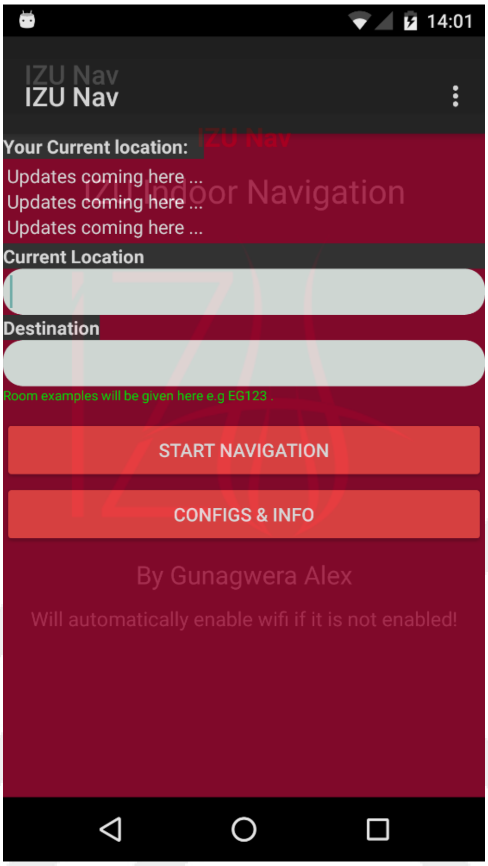
Enable unrestricted access makes routing possible through restricted areas (e.g. where key cards or key [illegible] this prototype user profiles & restrictions are not functional.

One MASTER_WiFi
MASTER_WiFi: -60

SAVE AND CONTINUE NAVIGATION

**Figure 6.4** Options Configuration Screen

**Figure 6.4** above shows a very user friendly option configuration screen of the suggested prototype. The message in the toast is of no significant importance. It is simply a message meant for debug purposes and it displays the strongest received sssignal at the moment the screenshot was taken.



Your Current location:

Updates coming here ...
Updates coming here ...
Updates coming here ...

Current Location

Destination

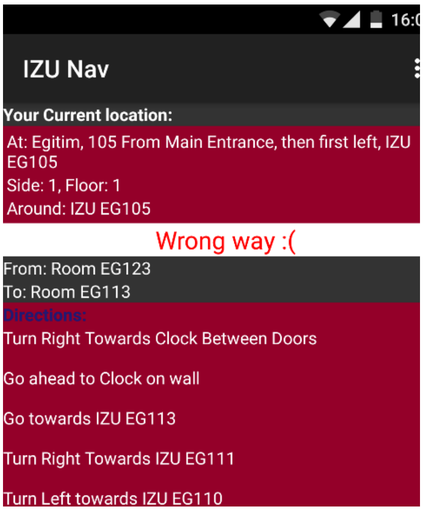
Room examples will be given here e.g EG123 .

By Gunagwera Alex

Will automatically enable wifi if it is not enabled!

**Figure 6.5** Source Destination Input Screen

**Figure 6.5** above shows the screen when the user actually interacts with the prototype. The destination field *must* be filled and the application will not go to the next phase – the navigation phase – without this field being filled. As for the user's current location, as long as the user is within the fingerprinted building or campus, the current location will be auto updated.



Your Current location:

At: Egitim, 105 From Main Entrance, then first left, IZU EG105
Side: 1, Floor: 1
Around: IZU EG105

Wrong way :(

From: Room EG123
To: Room EG113

Directions:

Turn Right Towards Clock Between Doors

Go ahead to Clock on wall

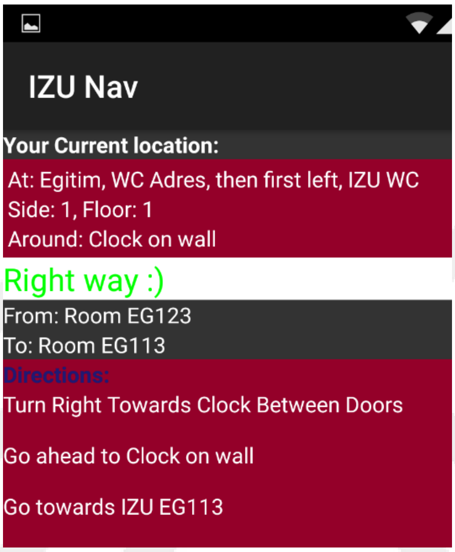
Go towards IZU EG113

Turn Right Towards IZU EG111

Turn Left towards IZU EG110

**Figure 6.6** Wrong Way

**Figure 6.6** above displays the alert presented to the user when they are off the right track. A visual and audio message are both issued. The audio message is periodically read out to the user until they get back on the right track. In this case we intentionally went past the desired target which was room EG 113 of the Education block in IZU campus.



Your Current location:

At: Egitim, WC Adres, then first left, IZU WC
Side: 1, Floor: 1
Around: Clock on wall

Right way :)

From: Room EG123
To: Room EG113

Directions:

Turn Right Towards Clock Between Doors

Go ahead to Clock on wall

Go towards IZU EG113

**Figure 6.7** Right Way

**Figure 6.7** above is the corresponding on the track message. And once again, the navigator is always told that they on the right track.