**CONTENTS**

**EXECUTIVE SUMMARY**

**1.0 INTRODUCTION**

**2.0 DESIGN PROBLEM STATEMENT**

**3.0 DESIGN PROBLEM SOLUTION**

**3.1 MOBILE APPLICATION**

**3.2 REMOTE MONITORING APPLICATION**

**3.3 PROTOTYPE COST**

**4.0 DESIGN IMPLEMENTATION**

**5.0 TEST AND EVALUATION**

**5.1 COMPONENT TESTING**

**5.2 SYSTEM TESTING**

**6.0 TIME AND COST CONSIDERATIONS**

**6.1 TIME CONSIDERATIONS**

**6.2 COST CONSIDERATIONS**

**7.0 SAFETY AND ETHICAL ASPECTS OF DESIGN**

**8.0 RECOMMENDATIONS**

**8.1 MOBILE APPLICATION**

**8.2 REMOTE MONITORING APPLICATION**

**9.0 CONCLUSIONS**

**REFERENCES**

**APPENDIX A**

**APPENDIX B**

**APPENDIX C**

**EXECUTIVE SUMMARY**

This report describes the results of an EE 464 senior design project from the University of Texas at Austin. The project was completed by Nick Carneiro, Merwan Hade, Tim Osborne, Rohan Singhal, and Harish Rallipalli. The project team was advised by Dr. Aristotle Arapostathis and sponsored by Texas Instruments. The goal of the project was to create an indoor tracking and personal navigation system that worked without using the Global Positioning System (GPS) because GPS is inaccurate indoors. The system has two components: a mobile application and a web-based remote monitoring application (RMA). The mobile application tracks a user’s position and communicates its current coordinates to the RMA. The RMA then displays the current location of all smartphone users for its administrator.

The system is designed to run on multiple smartphones with a separate monitoring application for the system administrator. The smartphone application, which was developed for the Android operating system, displays its user’s position in a map of the surrounding area. The project’s original objective was to use inertial sensors to track people’s locations throughout the target area, but the team altered the design to use Received Signal Strength Indicator (RSSI) values instead because RSSI values introduce less error. The new design uses a stored table of RSSI values to determine a user’s location by performing dot products between the user’s present RSSI values and those in the table. The remote monitoring application was originally designed to be a desktop application, but was changed to a web application to be platform independent.

Moving forward, the design team has a few recommendations for both of the team’s subsystems. In the mobile application, adding panning and zooming functionality is critical. Smartphone users expect to be able to do these things when using a mobile app. Another improvement for the mobile application would be adding linear interpolation to the tracking algorithm. As is, the tracking only returns points that are in the table which gives the tracking a jumpy feel. By adding linear interpolation, the mobile application tracking would appear much smoother because it would increase the quantity of points it can find. The mobile application could be further improved by adding navigation for the user. This would allow the user to ask for directions to different places around the target area, which would make it much more useful. At the same time, the RMA can be improved by adding heat maps for the administrators. These could identify the most visited areas of the map, which could be used for marketing purposes. The RMA could also be improved by making it a mobile website. This would enable an administrator to look at it on any device with a web browser.

The prototype developed by the design team showed that an indoor tracking system that uses RSSI values for tracking is a viable option. The result of the senior design project was a system that successfully demonstrates the desired functionality in an unpolished manner.

**1.0 INTRODUCTION**

This report describes an EE 464 senior design project completed at the University of Texas at Austin. The goal of this project was to create an indoor tracking and personal navigation (ITPN) system for Android smartphones with a web-based remote monitoring application (RMA). The system monitors the movements of people within a building. It uses Received Signal Strength Indicator (RSSI) values obtained from the phone’s Wi-Fi interface to track smartphone users as they move in an indoor area such as a shopping mall. Position coordinates will be simultaneously sent to a web application for remote monitoring by mall administrators. The design team for the project consisted of Nick Carneiro, Merwan Hade, Tim Osborne, Harish Rallapalli, and Rohan Singhal. The project’s advisor was Dr. Aristotle Arapostathis, and the project’s sponsor was Texas Instruments. The design described in the report proves that RSSI values can be successfully used to determine location.

The report begins by describing the problem the ITPN system solves. It defines the problem, overviews the design and its two subsystems, details a possible use for the system in a mall environment, and then describes the system requirements and constraints such as the spacing of Wi-Fi access points. The report then describes the system’s design in detail. It gives an overview of the entire design and describes other options for the design such as using inertial sensors or Wi-Fi fingerprinting and a web-based RMA or a desktop application, then details the design of each of the two subsystems: the mobile application and the RMA, then itemizes the cost of the prototype the project produced. Next, the report describes how the project was implemented. It explains why the team chose to use RSSI values to determine user position and why the team decided to use a web application instead of a desktop application. Next, the report describes the methods and results of the design team’s testing and evaluation. It details group’s tests of each individual component, and of the prototype system. The report then discusses the time and cost considerations of the design team. It details the decisions the team made to speed up the design as well as setbacks, like a faulty battery in a loaned phone, that slowed down the design. It also explains how the cost of the prototype totaled $0. Next, the report addresses the safety and ethical considerations of the design. Since the project is a pure software one, there are not many concerns in the area of safety, but the report addresses the team’s ethical consideration of invasion of privacy as well as a few others. The report concludes with recommendations from the design team. The team describes many improvements that could be made to the mobile application such as pinch zoom, panning, and interpolation, as well as improvements to the RMA such as heat maps.

**2.0 DESIGN PROBLEM STATEMENT**

While GPS signals are excellent for navigational and tracking purposes outdoors, they are weak and unreliable indoors. Tracking positions indoors can allow users to infer which areas are occupied during the course of a day from a commercial and security perspective. The ITPN system provides a comprehensive solution for indoor tracking and personal navigation using an Android powered mobile application and a web-based remote monitoring application.

The ITPN system was designed with mall shoppers and administrators in mind. The remote monitoring application will display shopper positions, showing markers for each. From a marketing point of view, data about shoppers’ positions in a mall can be used for advertising purposes. Signboards and electronic banners can be tailor-made for various locations of the mall and changed throughout the day to best match the people who frequent those areas. Shoppers must have an incentive to download the ITPN mobile application. So, in exchange for their positional data, the application will give shoppers digital maps that display their locations in real-time. In a large mall, it is often difficult to pinpoint one’s location without the aid of a map. The ITPN mobile application will solve this by helping shoppers find their bearings and destinations by displaying their positions.

The team decided to make a mobile application and web-based remote monitoring application to solve the above problem of accurate indoor tracking and navigation. The mobile application finds an accurate location fix on a smartphone in real time using Wi-Fi access points, and plots these coordinates on a digital map. Furthermore, it simultaneously transmits the coordinates to a remote monitoring application, which collects and displays data from multiple smartphone users. The team chose the Android platform because of its widespread use, availability of test phones, and collective experience with programming in Java. As for the remote monitoring application, initially the team had planned to implement a desktop based solution, however a web-based application allowed for greater platform independence.

The ITPN System had specific design parameters for each of its subsystems. For the mobile application, these included an accurate position fix, frequent position updates, and a friendly user interface. The position fix needed to be accurate to within a three meter radius and updated five times per second, accounting for a running shopper. The vast majority of mall-goers generally do not move faster than five meters per second (eleven mph). By computing position five times per second, the system will track shopper movement with a resolution of better than three meters. Additionally, Wi-Fi fingerprinting, which uses signal strength to pinpoint device location, will need to function with 80% accuracy even when there is interference in the Wi-Fi spectrum. This would result in four out of five locations determined from fingerprinting to be accurate within a three meter radius. Additionally, the user interface needed to adhere to Android Human Interface guidelines and provide easy-to-use buttons and menus for navigation [1].

For the remote monitoring application, the team ensured that the application displayed data from multiple users simultaneously. The team also ensured a one to one correspondence between the position fix shown on a mobile device and the corresponding position on the remote monitoring application. The original RMA specifications also defined heat map functionality, to facilitate data analysis for the administrator.

The ITPN system must operate in a standard mobile phone environment with enough Wi-Fi access points to provide network connectivity. The physical environmental requirements of the mobile system are the same as those of any Android phone. The environment will need to contain sufficient Wi-Fi coverage for the ITPN system to perform successful fingerprinting and transmission of location data to the remote monitoring application. The Wi-Fi fingerprinting technique becomes more accurate when more access points are in range of the smartphone. Since 802.11G access points have a typical range of 38m, mall administrators will need to install enough access points to cover the area to be monitored . These access points do not need to be on the same network, but they must remain stationary. However, if a Wi-Fi network already exists in the mall property, the ITPN system will only need a few additional access points rather than an entire network. Even if network connectivity is lost and Wi-Fi signals are too weak to transmit data, they can still be used for fingerprinting. When Wi-Fi connectivity is reestablished, the new position data can be relayed to the remote monitoring application. The mobile device running the application may access the Internet through these Wi-Fi access points or through a mobile carrier.

The processing power required in each part of the project is minimal compared to the capacities of smartphones and web servers. Therefore, the system is I/O bound. The most likely location of a bottleneck is the network because it must support hundreds of smartphones transmitting their locations. However, the network infrastructure’s capacity will exceed the system’s requirements even for large numbers of phones. For example, 802.11G links support speeds up to 54 Gb/s. If 1000 phones transmit the smallest Ethernet packet size (64 bytes) every second, the bandwidth usage only comes to 500 Kb/s. This leaves ample bandwidth for other network traffic.

**3.0 DESIGN PROBLEM SOLUTION**

The completed ITPN design consists of two subsystems. The first is an application for the Android mobile operating system, and the second is the RMA. The smartphone application sends real-time position information to the RMA, which then displays the position as a maker on a map. The system overview block diagram in Appendix A enumerates the inputs and outputs of each subsystem and the interaction between them. The mobile application accepts user input as well as environmental values and uses this data to output positional data. The RMA takes input from users as well as the mobile application, and uses these inputs to output positional data. The mobile application also calculates its phone’s position using a technique called Wi-Fi fingerprinting and displays the result to its user and simultaneously relays it to the RMA over a network link. When location data is received by the RMA, it updates the location information being displayed on its map.

**3.1 MOBILE APPLICATION**

Over the course of the project, several decisions were made to change the original system design in order to obtain higher tracking accuracy and finish the project within time constraints. The first decision was to do away with the use of inertial sensors (accelerometer, gyroscope, and e-compass), and only rely on Wi-Fi fingerprinting to get a location match. This allowed for a simpler lookup algorithm as well as more accuracy than inertial sensors would have provided. Another significant change was to implement the RMA as a web application instead of a desktop application. This allowed for faster development and easier access to the application.

The mobile application is a self-contained Android software binary that users run on their smartphones. After the user accepts the terms and conditions of the software, the Android application tracks the mobile phone’s changing position using Wi-Fi fingerprinting. The calculated real-time position is displayed on a map within the Android application.

The Android application determines its position, maps it, and relays that position to the RMA. Original design specifications outlined the use of a smartphone’s accelerometer, gyroscope, and compass to do tracking using an alternate method called dead reckoning. This technique requires the mobile user to input a starting location before the application begins tracking. However, using the inertial sensors produces drift error as the mathematical inaccuracies are compounded over time [2]. To compensate for drift error, the team originally planned to use Wi-Fi fingerprinting in addition to dead reckoning to correct the drift error periodically. However, the system design was eventually modified to only employ fingerprinting and the use of inertial sensors was completely eliminated. Research indicated that fingerprinting data is uniform across smartphones (not the case with inertial sensors - each phone has varying sensor accuracy), and moreover fingerprinting is the most accurate method of indoor tracking [3]. Another tracking technique that was considered was Wi-Fi trilateration. This method uses Wi-Fi signal strength value and the knowledge of where access points are located to pinpoint user position. However, this approach was eventually rejected due to frequent fluctuations in Wi-Fi signals as well as the complicated calculations that trilateration required.

To perform fingerprinting, the application compares received signal strength indicator (RSSI) values from nearby Wi-Fi access points to values in a table stored on the mobile device [4]. The table of values is created before tracking is performed by walking around the desired area and recording an average of Wi-Fi RSSI values at specific points. The spacing of these points determines the tracking accuracy; the team spaced out points at approximately three meters from each other. After location is determined by performing a lookup against the fingerprint data, this information is relayed over a network link to the RMA. The mobile phone subsystem block diagram in appendix B shows the internal layout of the mobile application and how its components work together to determine the smartphone’s location. Six of the eight blocks in the diagram are implemented in software, while the hardware and user blocks are self-contained, requiring no work from the team.

Android was chosen as the mobile application platform because of the wide range of software libraries publicly available for it. Also, most Android libraries are open source which means that they can be easily modified. Additionally, the Android Wi-Fi API is easy to use and provides access to all connection details, such as a list of visible access points, each of their ID’s, and their signal strength values [5]. The only other viable mobile platform was iOS, but it wasn't as attractive a choice because it doesn’t have open source libraries and requires each of the team members to learn a new programming language, Objective-C. Android programming, on the other hand, is done in Java, which the entire team was comfortable with.

**3.2 REMOTE MONITORING APPLICATION**

The RMA receives position data from smartphones and uses this data to display all of these positions on a map. It also stores all the data received, so that tables and graphs can be potentially constructed to do data analysis.

The only inputs to the RMA are position data and user manipulation of map controls. Output is displayed on the RMA map in the form of markers indicating smartphone users’ location. The block diagram in appendix C details the flow of position data from the network, through the application to the end user. When the RMA receives position data from mobile devices, it stores the coordinates in a database. The RMA’s plotting module then pulls the coordinates from the database and renders a marker on a map.

The original RMA design also called for functionality involving heat maps and data analysis. In this design, the plotting module would use data generated during the desired time range to generate a bar chart or heat map. These results from the plotting module would then be pushed to the view in an interactive form with user controls. However, given project time constraints, the team was unable to implement this functionality and decided to focus on making the core tracking part of the system more accurate.

The RMA is implemented as a platform-independent web application. The node.js framework was used to program the RMA. Node.js was chosen for its ability for rapid prototyping as well as its compatibility with MongoDB (the database schema the team used). Furthermore, the team had prior experience working with JavaScript and Node.js, and since the web application could be implemented completely in JavaScript, this reduced the learning curve associated with learning new programming languages.

The original design for the RMA defined a Java application that would communicate with a web server to get the position data. However, the team chose to implement a web application instead of a desktop application for several reasons. First, a web application allows for greater platform independence than a Java application, as well as the ability to use the RMA on a wide host of devices (tablets, smartphones, or any other device with a functional browser). Additionally, running a web application doesn’t require any setup beforehand such as installing a desktop application or requiring a computer to have a Java environment installed.

**3.3 PROTOTYPE COST**

Since the team developed the project in a university setting, the cost for developing the ITPN system was $0 because the materials were provided and the students were unpaid. All software used by the team was open-source and free of cost. The only hardware required was an Android smartphone with Wi-Fi. Since the team already had access to multiple Android phones before the project, hardware costs were also zero. The only other cost was the fee for maintaining a web server, which is a monthly charge. However, since the team already had one set up for different projects, this cost was not included in the development of the ITPN system. Additionally, the team was able to use the existing Wi-Fi infrastructure at the University of Texas at Austin instead of purchasing off-the-shelf wireless routers.

**4.0 DESIGN IMPLEMENTATION**

While the original design called for a personal inertial device for indoor tracking, the final design implemented tracking using Wi-Fi signal strength only. The decision to not use inertial sensors was driven by several factors. First, the sensors were found to be relatively inaccurate. Even when the phone was lying stationary on a table top, the accelerometer would give oscillating values. Furthermore, research indicated that the quality of the accelerometer readings vary widely between Android phones. The team did not want to build a solution specific to one phone. Moreover, the algorithm required to incorporate readings from the accelerometer, gyroscope, and e-compass was far more complicated than that required in a table look-up of Wi-Fi signal strengths. Finally, the team learned that a funded startup was attempting to commercialize the Wi-Fi fingerprinting technique, lending social proof to the method [6].

In practice, the different implementation meant using the Android WifiManager class instead of SensorManager in the location component of the mobile application. The network and UI components of the mobile application were left unchanged. Furthermore, the architecture of the RMA subsystem required no modifications. The modular nature of our system’s architecture meant that one localized change would not affect the entire system in a harmful manner.

The position lookup itself has O(n \* m) space and time complexity, where n is the number of rows in the fingerprint table and m is the number of access points detected. The table contains all previously recorded RSSI values and XY coordinates. Each row represents a different coordinate with associated signal strengths. For each row in the table, a loop computes the sum of dot products between the current RSSI values and those in the table [7]. At the end of the loop, the point with the minimum difference is considered to be the closest point and sent to the RMA.

Abandoning the inertial approach means that the new design requires existing Wi-Fi infrastructure wherever the product is deployed. The team decided that this was an acceptable requirement given that most malls have wireless networks. Additionally, the ITPN system does not require any cooperation from the administrators of the Wi-Fi network because the mobile application does not need to associate with or access the network in any way. The phone only needs to detect that the wireless signals are present. If connectivity through Wi-Fi is not available, then data is sent through the phone’s wireless carrier (3G or 4G). If the phone can access the Wi-Fi network, then there is less latency between the mobile application and the remote monitoring application. All modern Android phones handle this transition seamlessly for the user.

The second large change in our design was switching the RMA from a Java desktop application to a JavaScript web application. This decision was driven by several factors. First, a web application is accessible from any Internet-connected computer; it does not need to be installed locally. Mall administrators can simply navigate to a URL to monitor shoppers. Second, there was excellent publicly available code for mapping in a web browser. After surveying the options, the team found that the OpenLayers library was a mature, well-documented project that we could easily adapt for indoor real-time use. Third, having a web application makes it easier to manage historical data in one central location. The team used MongoDB to persist all location data reported to the RMA. And finally, the use of a domain name means that the mobile application doesn’t require any configuration to connect to the RMA. If the team had built a desktop application, it would have required tedious entry of the RMA’s IP address.

Opting for a web application instead of a desktop application meant that the system required a dedicated web server. Fortunately, the team already had one set up. Low-end virtual private servers are available at very affordable rates and the aforementioned benefits clearly warrant this nominal expense.

**5.0 TEST AND EVALUATION**

Testing the ITPN system involved basic testing of three main components: tracking, network, and GUI. The mobile application and the RMA subsystems were tested concurrently. Testing was concluded by integration and system tests that were used to verify that the two applications can communicate with low latency and that information can successfully be transmitted between them. The main criteria considered during testing were tracking accuracy and the successful transmission of location from the mobile application to the RMA.

**5.1 COMPONENT TESTING**

The five core components tested were the mobile application’s tracking, network, and GUI components as well as the network and GUI components of the RMA. Testing the tracking component involved verifying that fingerprinting is performed within desired accuracy of three meters. The network component performs data transmission between the mobile application and the RMA. Testing included verifying that network lag is within the acceptable time range, and checking successful transmission of data without any data corruption. Finally, the GUIs for both applications were tested; the GUIs were assessed on fairly basic criteria such as having functional buttons, ease of access for all menu items, and a user friendly interface.

The main benchmark for the ITPN system is the margin of error in the mobile application’s tracking component. The desired level of accuracy was at most three meters deviation from actual location in an area where there is ample Wi-Fi coverage (three or more Wi-Fi access points). Unfortunately, the finished system was only able to report locations with about six meters of accuracy. The system is still usable but with this visible degradation in accuracy.

The team confirmed that areas with fewer access points reduce tracking accuracy up to six meters since the fingerprinting algorithm does not have sufficient data to perform a lookup. This was evident during open house day on the second floor of ENS where the system was much more accurate in the hallway, but far less accurate inside the room. The attenuation of Wi-Fi signals traveling through the walls made it difficult to maintain the same accuracy everywhere on the floor.

Another important measure of accuracy was the lag imposed by the network component of the two applications. Network lag created delay between the location being updated on the mobile application and the RMA. For the most part lag was not a problem; 99% of packets were completed in under one second, but the remaining packets took up to ten seconds. The accepted measure of accuracy for this criterion was a one second lag between the two applications. The system fulfills this requirement because the slow packets are rare. Although network lag was minimal, the frequency of location updates was lower than the original target of five updates per second. Scanning Wi-Fi hardware on Android is a time-consuming procedure, which meant that the mobile application was only able to update location once every second. This change in frequency meant that the ITPN tracking component was less accurate since the reported location trailed the true location by a few seconds.

Both the GUI in the mobile application and the RMA were deemed user-friendly. The best proof was the fact that visitors during open house were able to use the system. They panned the map easily because it behaves like other pieces of popular mapping software.

**5.2 SYSTEM TESTING**

System level testing required two team members, one with a smartphone and another with a laptop computer. For the first test, the person with the phone verified its position output by looking at the screen and the person with the laptop verified that the map output on the web application was the same. The latter individual sat down on the second floor of ENS and watched the laptop while the other person walked around the floor in a random route. The smartphone holder left markers to track his route. After returning to the stationary individual with the laptop, the smartphone holder asked the laptop user for a detailed oral description of the route. This test was successful because in every instance the oral description matched the actual route taken.

The second test used the same setup, but the individuals stayed in the same corridor so the mobile user’s route was visible to the laptop user. While the smartphone holder walked around the room, the laptop user looked at the laptop screen to confirm that it showed the mobile user’s location. The results of this test demonstrated that the system did not attain the desired three meters of accuracy, but it did confirm that the accuracy was no worse than seven meters. This test also confirmed that the product has excellent latency when used on a fast Internet connection.

The team had originally designed a third test where the smartphone holder would follow a line drawn inside the mobile application, but this was considered unnecessary once the accuracy was determined by the two previous tests.

The integration of the two subsystems was flawless. There was no discernible difference between the output of the mobile application and the RMA. Moreover, the math to translate between the coordinate systems was straightforward and precise.

The only problem identified with our system was the inaccuracy of the tracking component in the mobile application. It technically failed the tests originally designed by the team. However, these tests did help us determine the true accuracy which was bounded at seven meters. The system is still usable even with this caveat.

**6.0 TIME AND COST CONSIDERATIONS**

The tools required to construct the ITPN system were readily available and free of charge, largely releasing the team from financial considerations. The Android smartphone was loaned through the check-out desk located on the second floor of the ENS building at The University of Texas at Austin, and the Eclipse development environment was downloaded for free. Since wireless routers were needed for the project, the team used the university's existing wireless infrastructure to save time and money. Project schedule and the delegation of the work are outlined in the Gantt Chart in appendix D. The schedule of the Gantt Chart was closely followed with the exception of accelerometer and gyroscope testing, which became unnecessary after the inertial component of the project was eliminated.

**6.1 TIME CONSIDERATIONS**

Throughout the project, many items affected the timing of the projects completion; some of these sped the timing up, some slowed it down. The initial plan to find an indoor shopping environment or open area with little electromagnetic noise turned out to be unnecessary as Wi-Fi fingerprinting can accommodate for some interference. The geographic placement of and students' easy access to the access points, or routers, inside the ENS building matched the conditions required for Wi-Fi fingerprinting. These conditions allowed for each floor’s area to be covered by a Wi-Fi signal with the typical range of 38m produced by 802.11G access points. Thus, the ENS building was determined to be the best area for testing and prototyping the ITPN system, saving the team expenses and commuting time involved in using an off-campus location.

The initial plan also called for manual mapping of the area of interest, but this was later rejected in consideration of time. The task of manually acquiring dimensions of the floor was unnecessary because accurate floor plans were obtained from the ENS building staff. The image files of the floor plans were scaled and directly overlaid with a coordinate system, avoiding the need to create a map from manual measurements.

The schedule was delayed due to a faulty battery in the first Qualcomm Snapdragon 8655 mobile development platform borrowed by the team from the ECE Department. This delay, along with internship obligations, resulted in no project progress over the summer.

Modularizing the workload also improved the team’s time management. This modularization allowed the team to work on different pieces of the project individually; then incorporate each finished component into the system. The unilateral decision-making regarding the technology used for the RMA and its early completion helped provide the time needed to integrate it with the mobile application. This allowed the team to use the RMA as a helper to their design process, which made this team’s job faster.

**6.2 COST CONSIDERATIONS**

Over the course of the project, budgetary constraints were not a large concern. The team had no financial obligations because the project was designed to use hardware that was available to them. During the system design phase, hardware requirements included three Cisco Aironet 3500 series access points with a total estimated cost of $1,890. By levaraging the university's Wi-Fi, the team saved the cost of creating an independent network of access points.

In April 2011, the Qualcomm Snapdragon 8655 mobile development platform loaned to the team by the ECE Department had problems with its battery. The subsequent delay led the team to discuss the prospect of using iOS as the development platform since three members already had one in their possession. After a cost analysis, the team rejected this idea based on the cost required to purchase Apple Developer Tools and the learning curve for Objective C. Eventually, the team obtained a working Qualcomm development platform from the ECE department and another member of the team purchased an Android phone for personal use; this resulted in the team having no cost for the phone the system was tested on.

**7.0 SAFETY AND ETHICAL ASPECTS OF DESIGN**

The ITPN system addresses several ethical and safety considerations such as invasion of privacy, reliability of data, and anonymity and storage of positional data. The ITPN system requires that smartphone users will agree to have their locations tracked in exchange for navigational assistance. This could be perceived as an invasion of privacy because smartphone users may feel as if they are being watched or followed. This is a valid concern of the users, but the navigation assistance they receive should compensate for any loss of privacy. In order to address this issue, the mobile application users are required to consent to a user agreement which clearly states that their positions will be transmitted to a web application.

A potential concern is the reliability of data. Smartphone users may depend solely on the mobile application for navigation, in which case a user interface lag or momentary incorrect data may cause confusion for the user. The mobile application user agreement provides a disclaimer stating that the user should rely on common sense and good judgement in addition to digital maps.

Additionally, smartphone users may have concerns over how they are identified and how their positional data is used and stored. The remote monitoring application identifies each user using the individual’s Android device id. This information, however, is not accessible via the RMA interface since each smartphone user is shown as an anonymous marker. Furthermore, a history of each individual user is not stored by the RMA so concerns about misuse are mitigated.

From a public interest point of view, the ITPN system can be re-purposed to track inaccessible people such as children or the elderly. For example, it can be used to track elderly people at a retirement home. The administrators can keep track of the home residents using the remote monitoring application, while the residents can use the application to move around the home. Another example could be tracking children’s locations in a neighborhood. Parents can use the remote monitoring application to track the position of their children and thereby ensure their safety. The ITPN system can also be used to prevent unauthorized access. If all users’ positions are tracked, unauthorized entry into a restricted location can be easily detected.

**8.0 RECOMMENDATIONS**

Although the ITPN system had essential functionality implemented and worked fairly well within acceptance criteria and time constraints, the project has significant room for further expansion in both the mobile application and the RMA components. There are a few user interface features that the team was unable to implement that would be essential in a commercial application. Pan and zoom controls for the mobile application, and heat map and graph generation for the RMA are important functions that the team was unable to implement. Additionally, the tracking module of the system was not as accurate as the value that initial requirements specified. The final system was accurate to within six meters, whereas the target was a maximum three meter deviation from actual location. Finally, there was a noticeable lag between doing the lookup calculation and rendering position on the mobile phone map.

**8.1 MOBILE APPLICATION**

The mobile application’s map view was designed around commercial map applications that all Android phones come with. These commercial applications have controls like panning and zooming defined as simple user gestures, such as pinch to zoom and drag to pan. Although the team was able to load a custom map image, we weren’t able to implement these user controls, which are an expected important feature for any mapping application. Additional improvements to the mobile application would be regarding the tracking itself. First, the tracking algorithm was slightly more inaccurate than what the team expected. This could have been improved by allowing for linear interpolation. The current lookup matching algorithm simply finds a closest match, but a smarter procedure would have the ability to infer location between two stored points based on the real-time data passed in. Another useful tracking feature would have been the ability to distinguish what floor the mobile user is on. Current location matching is only done in two dimensions; matching in three dimensions would allow the application to detect the floor the user is on and automatically load the map for that floor.

An additional important feature to add to the mobile application can be the ability to provide navigational assistance. While the current application only shows current location and navigation is performed manually by the user, a more functional application would display a route overlay on the map that would instruct the user where to go. The navigation could in turn be expanded to reroute the user dynamically based on any incorrect turns the user makes. One last behavior on the mobile subsystem that could be addressed is the lag between lookup calculations and showing updated position on the map. During demo day, there was a three second lag before an updated position was displayed on the mobile phone. This issue was probably related to how threading is performed on Android, but finding a fix would involve further debugging.

**8.2 REMOTE MONITORING APPLICATION**

The RMA was more functional than the mobile application and it fulfilled more of its original specifications than did the mobile application. Useful additions to the RMA would have been the ability to create heat maps and do other data analysis on stored location data. Although this was part of the original design, the team was unable to implement this due to limited time constraints. This feature would have allowed users of the RMA to see data regarding areas most frequented in easy to read heat maps. Heat maps represent data using color codes to show different frequencies. Complete implementation of heat maps would allow the RMA user to specify a time frame for which to view relevant data.

Another important feature for the RMA would be a mobile website. A big advantage of implementing the RMA as a web application is the ability to use the application on smart phone browsers. However, the RMA interface does not translate over very well, which makes the application difficult to use on smart phones. A more functional RMA would detect whether the device accessing it is a smart phone, and render the page accordingly. Additionally, since the RMA was implemented with mall administrators in mind, an authentication procedure to access it would be helpful.

**9.0 CONCLUSIONS**

This report described an EE 464 senior design project from The University of Texas at Austin that created an indoor tracking and personal navigation system. The project resulted in a successful prototype that has room for improvement. The resultant system consists of two subsystems: a mobile application designed for Android smartphones and a web-based remote monitoring application. The prototype proved that RSSI values can be used for indoor tracking, solving GPS’s inaccuracy issues indoors. The design team’s solution requires access points spaced at most thirty-eight meters apart covering the entire area the system monitors.

The team’s recommendations moving forward are specific to each subsystem. Regarding the mobile application, significant improvements could be made by implementing linear interpolation to the tracking algorithm to increase the number of communicable points, adding panning and zooming features to the mapping element, adding a third dimension to the points so a users floor could be determined, and adding navigational assistance so users can decide where they want to go and receive directions. The remoter monitoring application could be improved by adding heat maps and making it a mobile website so administrators could access it through various devices. The final product was less accurate than planned and lacked heat maps which were an important component in the system’s design. The project succeeded in proving that a tracking algorithm can rely on RSSI values. The project culminated in a working, although unpolished ITPN system.

REFERENCES

[example 1] “Estimated World War II Dead,” [Online]. Available: http://warchronicle.com/numbers/WWII/deaths.htm. [Accessed: 26 November, 2011].

[example 2] Winkler, K, “50 Years Later, the Debate Rages Over Hiroshima,” Chronicle, 21 April 1995, [Online]. Available: http://chronicle.com/article/50-Years-Later-the-Debate/85731/. [Accessed: 26 November 2011].

[example 3] Global Security Institute, “Nuclear Timeline,” [Online]. Available: http://www.gsinstitute.org/dpe/timeline\_2.html. [Accessed: 26 November 2011].

[1] “User Interface Guidelines,” [Online]. Available: <http://developer.android.com/guide/practices/ui_guidelines/index.html>. [Accessed: 26 November 2011].

[2] “Tracking Position Indoors: moving from Hype to Reality,” [Online]. Available: http://www.sensorplatforms.com/tracking-position-indoors-moving-from-hype-to-reality. [Accessed: 26 November, 2011].

[3] “Wi-Fi Localization Using RSSI Fingerprinting,” [Online]. Available: http://www.digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1007&context=cpesp. [Accessed: 23 November 2011].

[4] Raja, Wi-Fi Indoor Postioning System (Wi-Fi IPS), 2008. Available: http://sites.google.com/site/monojkumarraja/academic-projects/wi-fi-indoor-positioning-system. [Accessed: April 3].

[5] “Android Developers,” [Online]. Available: http://developer.android.com/reference/android/net/wifi/WifiManager.html. [Accessed: 19 November 2011].

[6] <http://articles.businessinsider.com/2011-06-02/tech/30057143_1_wi-fi-mapping-buildings>

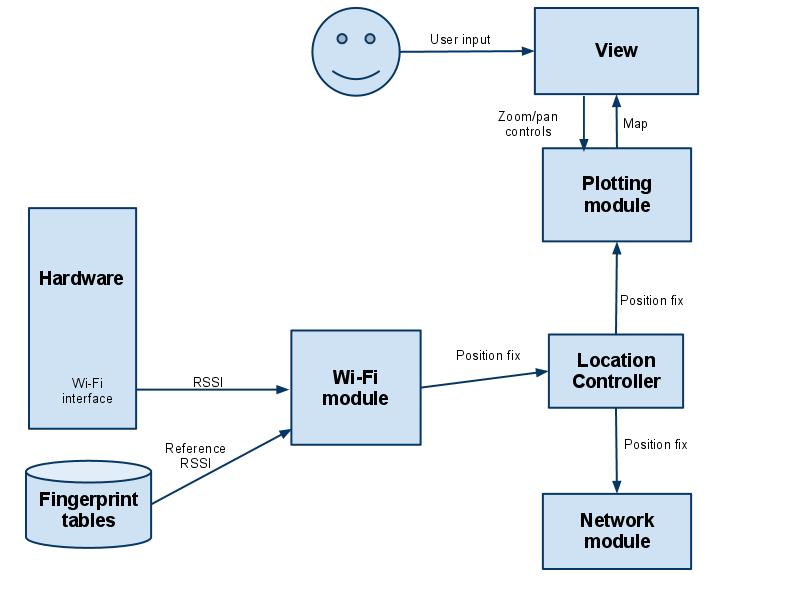
**APPENDIX A - System Overview**

**APPENDIX A - System Overview**



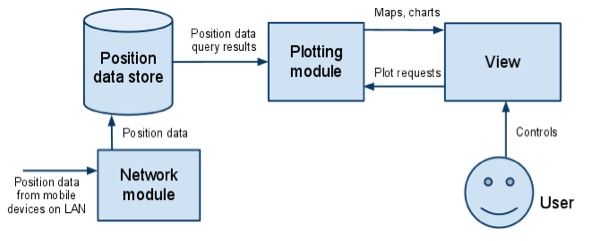
**APPENDIX B - Mobile Phone Subsystem Block Diagram**

**APPENDIX B - Mobile Phone Subsystem Block Diagram**



**APPENDIX C - Remote Monitoring Subsystem Block Diagram**

**APPENDIX C - Remote Monitoring Subsystem Block Diagram**



4.2

APPENDIX D **- System Design Gantt Chart**

