Table of Contents

1 Chapter : Introduction....3
1.1 Problem Definition....4
1.2 Motivation....5
1.3 Aims and Objectives....6
1.4 Scope....7
1.5 Outline....8
2 Chapter : Background....9
2.1 Literature Review....11
2.1.1 Open wireless sensor network telemetry platform for mobile phones....11
2.1.2 Sensor network to Android platform to control Rovio Robot....12
2.1.3 Location and Mobility in a Sensor Network of Mobile Phones....13
2.1.4 Conclusion....13
2.2 Analysis....14
2.2.1 Functional Requirements....15
2.2.1.1 User Login....17
2.2.1.2 Add Job....18
2.2.1.3 View/Edit Job....19
2.2.1.4 Synchronize....19
2.2.1.5 Record Service....20
2.2.1.6 View Data....21
2.2.1.7 Email Data....21
2.2.1.8 User Ranking Process....21
2.2.1.9 Server Side Implementation....22
2.2.1.10 Exception Handling....22
2.2.1.11 Service Level Authentication....22
2.2.2 Non-Functional Requirements....22
2.3 Review and Compare....23
3 Chapter : Methodology....24
3.1 Proposed Solution....24
3.2 Design Overview....24
3.3 Front End Technologies....25
3.3.1 Suitable mobile platform....26
3.3.2 Development Environment....27
3.4 Back End Technologies....28
3.5 Implementation Details....28
3.5.1 Recorder Service....29
3.5.2 Add Job....30
3.5.3 Select Jobs....31
3.5.4 Synchronize....32
3.5.5 Upload Data....33
3.5.6 Database Design....34
4 Chapter : Evaluation....35
4.1 Project Plan....35
5 References....35
6 Web References....36

List of Tables

Table 4.1 : Project Plan.....17

List of Figures

Fig 2.1: Structure of a typical sensor network..... 09

Fig 2.2 : Wireless sensor network gateway for transmitting data to nearby mobile phones over a short-range radio link.....11

1 Chapter : Introduction

An ad-hoc or short-live network is the network of two or more mobile devices connected to each other without the help of intervening infrastructure. In contrast to a fixed wireless network, an ad-hoc network can be deployed in remote geographical locations and requires minimum setup and administration costs. Moreover, the integration of an ad-hoc network with a bigger network-such as the Internet or a wireless infrastructure network increases the coverage area and application domain of the ad-hoc network.

An ad-hoc network can be classified into two main types: mobile ad-hoc network and mobile ad-hoc sensors network. Unlike typical sensor networks, which communicate directly with the centralized controller, a mobile ad-hoc sensor network follows a broader sequence of operational scenarios, thus demanding a less complex setup procedure. A mobile ad-hoc sensor or hybrid ad-hoc network consists of a number of sensor spreads in a geographical area. Each sensor is capable of mobile communication and has some level of intelligence to process signals and to transmit data. In order to support routed communications between two mobile nodes, the routing protocol determines the node connectivity and routes packets accordingly. It also helps them to save its limited battery power on the individual nodes as well. This makes a mobile ad-hoc sensor network highly adaptable so that it can be deployed in almost all environments.

The mobile ad-hoc sensor network is a new invention with long-term potential for transforming our daily lives. In mobile ad-hoc sensor networks, each host may be equipped with a variety of sensors that can be organized to detect different local events. Moreover, an ad-hoc sensor network offers less overhead on laying infrastructure and deploying. But in terms of initial cost it will be quite high, because those nodes are comparatively expensive. Therefore depending on the number of nodes

1.1 Problem Definition

Wireless ad-hoc sensor networks are very beneficial in different scenarios in different manner. These networks advance operational efficiency of certain civilian applications. [WWW03]

i. Military sensor networks to detect and gain as much information as possible about enemy movements, explosions, and other phenomena of interest.

ii. Sensor networks to detect and characterize Chemical, Biological, Radiological, Nuclear, and Explosive attacks and material.

iii. Sensor networks to detect and monitor environmental changes in plains, forests, oceans, etc. One environmental application is tsunami alert system.

iv. Wireless traffic sensor networks to monitor vehicle traffic on highways or in congested parts of a city.

v. Wireless surveillance sensor networks for providing security in shopping malls, parking garages, and other facilities.

vi. Wireless parking lot sensor networks to determine which spots are occupied and which are free.

Even though sensor networks are so important and useful there are some drawbacks inherited with it, which will restrict the usage.

i. Need to deploy hundreds of node

   In order to collect accurate and efficient data it is necessary to deploy lots of sensor node on the selected geographical locations

ii. Considerable amount of initial cost

   Usually single sensor node with generic sensors cost about hundreds of dollars. Therefore it will cost huge amount of initial cost since it is necessary to deploy lot of sensor.

iii. Power constraints

   Those nodes are battery powered. And they can only be active for limited hours of time before battery drained. After that it is necessary to recharge each and every individual node in order to continue further.

iv. Communication constraints

Since those are low powered devices communication need to be done on very carefully without wasting more power. But anyway it need to necessary information over to the base station.

v. Security complexities

vi. Node tracking issues

If it is necessary to make those nodes mobile, then there need to be a proper tracking mechanism to track down each and every node.

Therefore it is clear that, even though it is very helpful and flexible for researchers, there are many drawbacks that it needs to take into consideration before going for it. And if there are any other alternative to overcome these weaknesses sensor networks will become part of human life.

1.2 Motivation

Ad-Hoc wireless sensor networks are becoming an important part of the technology. Its already been using in different scenarios to save lives. And the basic goals of using a wireless ad hoc sensor network generally depend upon the application, but the following facts are the most common motivate to use sensor networks.

i. Determine the value of some parameter at a given location: In an environmental network, it might necessary to know the temperature, atmospheric pressure, amount of sunlight, or the relative humidity at a number of locations. As this example it is clear that a given sensor node can be connected to different types of sensors, each with a different sampling rate and range of allowed values.

ii. Detect the occurrences of 'events of interest' and estimate parameters on the detected event: In the traffic sensor network, one would like to detect a vehicle moving through an intersection and estimate the speed and direction of the vehicle.

iii. Classify a detected object: That is a vehicle in a traffic sensor network to identify as a car, a mini-van, a light truck, a bus, etc.

iv. Track an object: In a military sensor network, track an enemy tank as it moves through the geographic area covered by the network.

With the rise of powerful mobile phone devices and mobile platforms like Google Android, Blackberry OS or Apple iOS, it has become feasible to find alternatives to avoid those problems that are bundle with traditional sensor networks.

i. Mobile phone platforms tend to be widely available, as economies of scale drive their prices down.

ii. Their hardware and software systems are much more homogeneous in nature than most of sensor network architectures as a whole.

iii. They have built-in support for a number of sensor needs, such as cameras, cellular radios, magnetometers, gyroscope, near field communication sensors etc.

iv. Because of these devices are broadly used for personal consumption, they provide a sense of immediate availability not seen in traditional embedded systems. That is most of the time those devices are ideal and can be used for some effective work.

Therefore there are lots of sensor networks deployed all around the world for different purposes, even though there are many drawbacks. And if there are some alternatives to avoid those drawbacks, it can gain the maximum use from these sensor network concepts. After this has been implemented everyone can use this network to collect the relevant magnetic fields data from any geological location as they expected. Hence this can save lot of funds and effort by avoid implementing new sensor network from the scratch. And going forward, if it can create a trend to use mobile phone as sensor node then it will lead to introduce more sensors into upcoming smart phones. Therefore it will make this concept more useful and expandable in future.

1.3 Aims and Objectives

Therefore as it's explained above the main objective of this is to find a way to avoid the drawbacks of the ad-hoc sensor networks that will be by emulating sensor network using mobile phone. And following are the other objectives of this,

- Find an appropriate media to communicate with other nodes. There are many alternatives like bluetooth, infared, wifi, and mobile data connection. This can use different mediums depending on the situation.

- Model communication hierarchy of the sensor network that is how a single node shares information with its base station and how that base station can send all the collected data to the single centralized location. And how frequently it should upload those data. When deciding those factors it is necessary to consider low power, low memory. Low processing and other constraint on those sensor nodes.
- Elaborate the importance of this by using an innovative application to one of the major Smart phone platform. After take the all smart phone platforms in to consideration it is decided that android is the best suitable platform for this prototype implementation.
- User evaluation to measure the importance of this solution. This will ensure that findings of this effort are useful to the society.

1.4 Scope

- There are many mediums to communicate on mobile phones. Like Wifi, Bluetooth, Mobile data connection. Therefore it is necessary to find what the best medium to serve the purpose.
- Need to model the communication hierarchy. This will be the baseline for communication between node-node and node-base station. This should align with the communication medium selected above.
- Once best communication medium and communication hierarchy defined, a solid architecture design needed upon implementing the application.
- Implement the innovative application on one of the smart phone platforms, to demonstrate the importance.
- Using implemented application user can specify the type of information he needs, the frequency and around which location, based on GPS locations.
- Need to find method to find the closest nodes to given GPS location.
- User will have GUI, as well as a command base interface for advance users to query the necessary information.
- Implement a policy that a user will receive data as a proportion to their contribution to others to gather their information. That will keep everyone contribute to the community.

Therefore in this approach a researcher or any other with the need of this kind of data, do not need any more expensive sensor node. And also deploying an such a sensor network is a huge overhead which will take lot of effort, time and money. Therefore using this approach they can build a sensor network with lesser effort but with closely equivalent capabilities as traditional sensor network.

1.5 Outline

This is the basic introduction to the problem, aims and objectives of the project. The next, second chapter will describe the background on ad-hoc sensor networks, Android platform and literature review of the problem. And it describes the other solutions proposed, and developed for this problem and how far they are successful in solving this problem. It also briefly describes the functional and non-functional requirements that need to satisfy in order to address this problem, the communication methodology and architecture of the proposed solution.

Third Chapter is the Methodology. It describes design and architecture of the system. It includes design diagrams, ER diagrams of the database, deployment diagrams, pseudo codes for complex logics and test plan. Simply this chapter includes each and every implementation details where others can understand the gravity of this project.

Fourth and final chapter is the Evaluation. This includes the critical evaluation of the system. That is how far it is successfully achieved aims and objectives. The importance of this system to the society, strengths & weakness and the future improvements of the system will be discussed. All those points need to be justified in this chapter.

2 Chapter : Background

Although wireless sensor nodes have existed for decades and used for applications as diverse as earthquake measurements to warfare, the modern development of small sensor nodes dates back to the 1998 “Smartdust” project and the NASA Sensor Webs Project. One of the objectives of the “Smartdust” project was to create autonomous sensing and communication within a cubic millimeter of space. Though this project ended early on, it led to many more research projects. The researchers involved in these projects coined the term mote to refer to a sensor node. The equivalent term in the NASA Sensor Webs Project for a physical sensor node is pod, although the sensor node in a Sensor Web can be another Sensor Web itself. Physical sensor nodes have been able to increase their capability in conjunction with Moore's Law. The chip footprint contains more complex and lower powered micro controllers. Thus, for the same node footprint, more silicon capability can be packed into it. Nowadays, motes focus on providing the longest wireless range, the lowest energy consumption and the easiest development process for the user. [WWW05]

The WSN is built of "nodes", from a few to several hundreds or even thousands, where each node is connected to one or more sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a micro controller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoe box down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

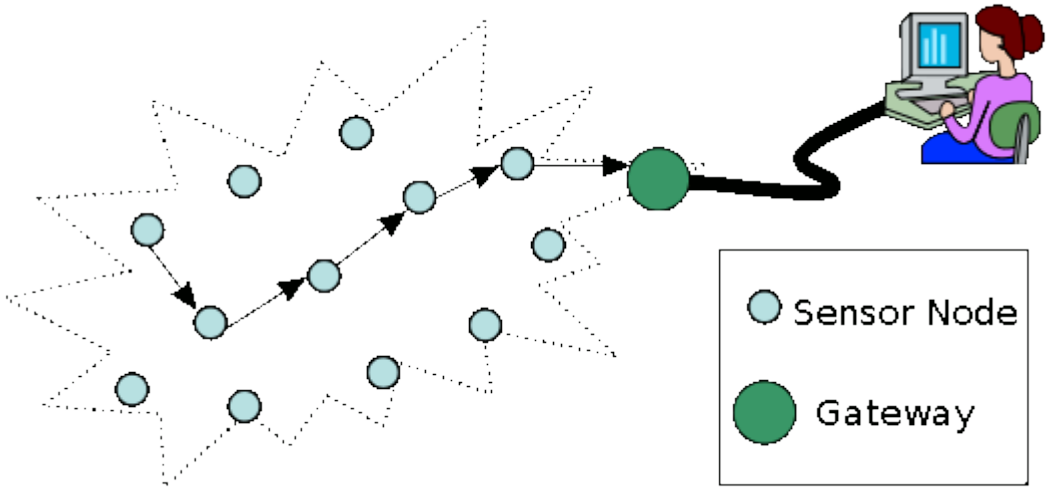

Fig 2.1: Structure of a typical sensor network

Some of the Wireless ad hoc sensor network characteristics are as follows,

Large number of sensors: Aside from the deployment of sensors on the ocean surface or the use of mobile, unmanned, robotic sensors in military operations, most nodes in a smart sensor network are stationary. Networks of 10,000 or even 100,000 nodes are envisioned, so scalability is a major issue.

Low energy use: Since in many applications the sensor nodes will be placed in a remote area, service of a node may not be possible. In this case, the lifetime of a node may be determined by the battery life, thereby requiring the minimization of energy expenditure.

Network self-organization: Given the large number of nodes and their potential placement in hostile locations, it is essential that the network be able to self-organize; manual configuration is not feasible. Moreover, nodes may fail, and new nodes may join the network. Therefore, the network must be able to periodically reconfigure itself so that it can continue to function. Individual nodes may become disconnected from the rest of the network, but a high degree of connectivity must be maintained.

Collaborative signal processing: Yet another factor that distinguishes these networks from MANETs is that the end goal is detection/estimation of some events of interest, and not just communications. To improve the detection/estimation performance, it is often quite useful to fuse data from multiple sensors. This data fusion requires the transmission of data and control messages, and so it may put constraints on the network architecture.

Querying ability: A user may want to query an individual node or a group of nodes for information collected in the region. Depending on the amount of data fusion performed, it may not be feasible to transmit a large amount of the data across the network. Instead, various local sink nodes will collect the data from a given area and create summary messages. A query may be directed to the sink node nearest to the desired location.

2.1 Literature Review

Sensor network are been very helpful to the society throughout the last decade. Therefore many people try to avoid those drawback them to make maximum use of them. And make it less complicated for people who want to build up new sensor networks. In order to do so many people try to use public mobile phone to enhance the capabilities of sensor networks. Some of those efforts are described below.

2.1.1 Open wireless sensor network telemetry platform for mobile phones

Mobile phones are an underutilized resource for connecting low-power wireless sensor networks to the Internet. WSNs typically expend most of their battery power on data transmission. Mobile phones carried by the public could enable a hybrid approach where data makes a low-power short distance hop to phones in the vicinity using Bluetooth or a similar short range protocol, then uses the phones' long distance connectivity to upload to the Internet. Because a large fraction of mobile phones have Bluetooth short-distance radio, this effort is to find low-cost hardware for generic WSN-to-Bluetooth gateway and open-source software that allows a wide subset of mobile phones to download and save WSN data.

This concept was proposed by C. K. Harnett and his effort is to makes available an open-source "opportunistic telemetry" system that creates the long-distance data link using mobile phones carried near the sensor network, reducing the power burden on the WSN. The overall system is illustrated in Fig 2.2 below.

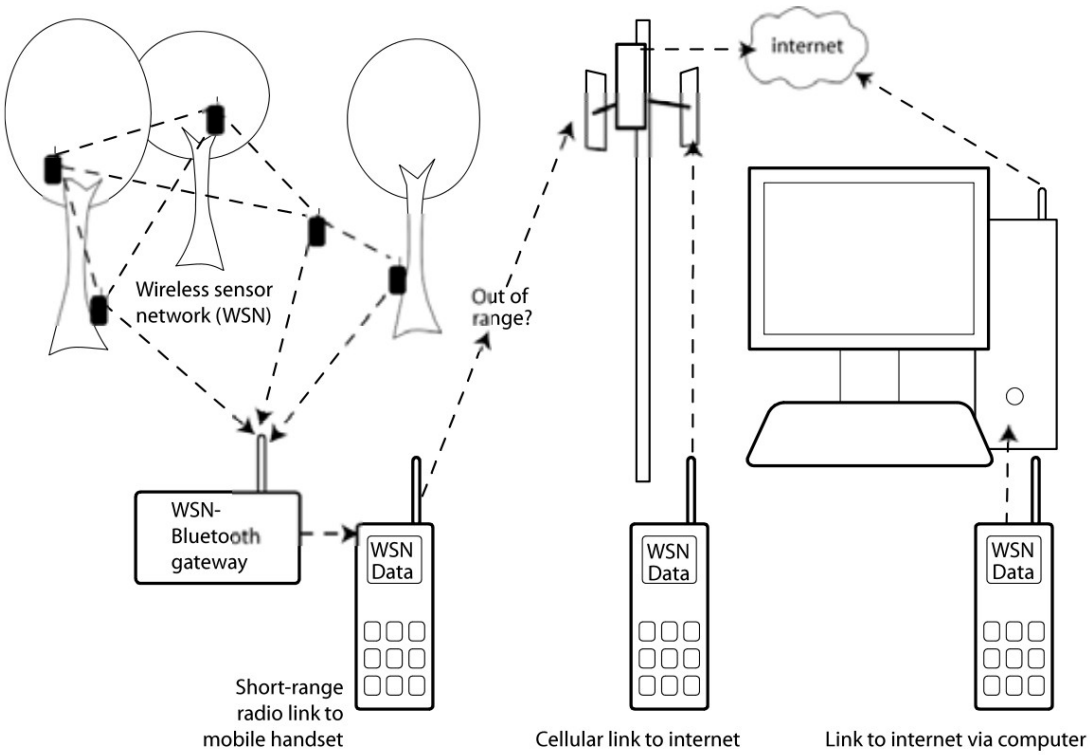

Fig 2.2 : Wireless sensor network gateway for transmitting data to nearby mobile phones over a short-range radio link

2.1.2 Sensor network to Android platform to control Rovio Robot

This effort was done by Yusuf Simonson, Robert Fowler at Renaissance Computing Institute and Edgar Lobatón at The University of North Carolina.

They believe that new smart phone architectures like Android and iOS will play increasingly important roles in sensor network design. Because of this, they wish to investigate the state of development for sensor network-based applications on the Android framework. They introduce an application for Android that allows for the semi-autonomous remote control of Rovio robots. It coordinates with a sensor network of cameras to provide a live stream of camera images and predicted robot locations. Furthermore, it provides functionality for moving the robot to user-selected destinations without the need for manual control. The application acts as a straw man for working with traditional sensor network architectures, and provides important insight on some of the challenges related to sensor network development, especially on the Android platform. They outline these challenges and provide some suggestions on semantics that could alleviate development effort.

2.1.3 Location and Mobility in a Sensor Network of Mobile Phones

This research was done by Aman Kansal and Feng Zhao at Microsoft Research center.

Mobile phones have two sensors: a camera and a microphone. Their goal in this position paper it to explore the use of these sensors for building an audio-visual sensor network that exploits the deployed base of millions of mobile phones worldwide. Among the several salient features of such a sensor network, this focus on mobility. Mobility is advantageous since it yields significant advantage in spatial coverage. However, due to the uncontrolled nature of device motion, it is difficult to sample a required region with a given device. They propose a data centric abstraction to deal with this difficulty. Rather than treating the physical devices as their sensor nodes, they introduce a layer of static virtual sensor nodes corresponding to the sampled data locations. The virtual nodes corresponding to the required region to be sensed can be queried directly to obtain data samples for that region. They discuss how the locations of the virtual sensor nodes can be enhanced, and sometimes derived, using the visual data content itself. Experiments with real data are presented to expose some of the practical considerations for their design approach.

2.1.4 Conclusion

It is believed that smart phones or platforms with similar capabilities will play an increasingly prevalent role in sensor network architectures given their commodity pricing and rich sensing hardware. And with their increasing capabilities can used in more meaningful manner to build up simple but useful sensor network.

All above mentioned solutions are working in some way and tried to address some of the drawback appear on sensor networks. But there are more areas to cover in this aspect.

2.2 Analysis

Detail requirements need to be identified and analyzed before going into detail design of the solution. Therefore following segment describe the requirement analysis and the design to the problem described above.

Even though sensor networks are there for quite a some time, still there are lots areas that needs to be improved to make the better use of it. This effort is also to reduce the infrastructural overhead of sensor network using public mobile devices as sensor motes. That will avoid many difficulties that were with traditional sensor motes. In the meantime this should not affect users mobile device usage.

These day mobile device are equipped with many important sensors like magnetometers, NFCs, gyroscopes, accelerometers, GPS modules, Bluetooth, cameras, radio links and there are lot more sensor waiting to be introduced into them. And most of the time those sensors are idle on those devices. Therefore in this context those idle sensor can be use to get some important work. That is we can build expensive sensor network out of those idle mobile devices by avoid initial overhead and cost.

If we can build application that can convert a mobile device into sensor node, then it can collect data and at the end it can upload all the data to the centralized location using long distance data connection. This will be the data connection of that mobile device. The other important fact is once user starts the application this should be able run on the background without affecting usual user operations until he intentionally stops it. Until that it will collect, communicate and upload data as requested. Anyone who is running the application will be a user as well as a contributer to the sensor network. And it will always send data as per others user requirements. But if a user need to collect data then he should be able to make a request to the system with his data requirement criteria. Then as for that request others should upload data to the system.

In order to make this system a reality, there should be a mobile device application which runs on user devices and a server side back end. Mobile application will become front side which interact with the user and it always communicate with the back end to cater user requests.

2.2.1 Functional Requirements

Any coherent and reasonable project must have requirements that define what the project is ultimately supposed to do. A requirement is an objective that must be met. Analysts cast most requirements in functional terms, leaving design and implementation details to the developers. Functional requirements describe what the application is supposed to do by defining functions and high-level logic. Following diagram fig:2.3 illustrates overall functionalities of the mobile application and users interaction with the system. Moreover this use case diagram only shows user interaction with the mobile application. There is service back end, implemented on Axis2 web service platform deployed on Apache tomcat web server.

There are three major type of users and there are,

i. New Users
ii. Users who add new jobs
iii. Users who executes jobs

New user can only create user account which it is necessary to login to the system. And other two type of users can do almost same tasks. But depending on their role, there are some tasks they can perform and can not. As a example, only the user who add a particular job is capable of viewing or retrieving the data from server.

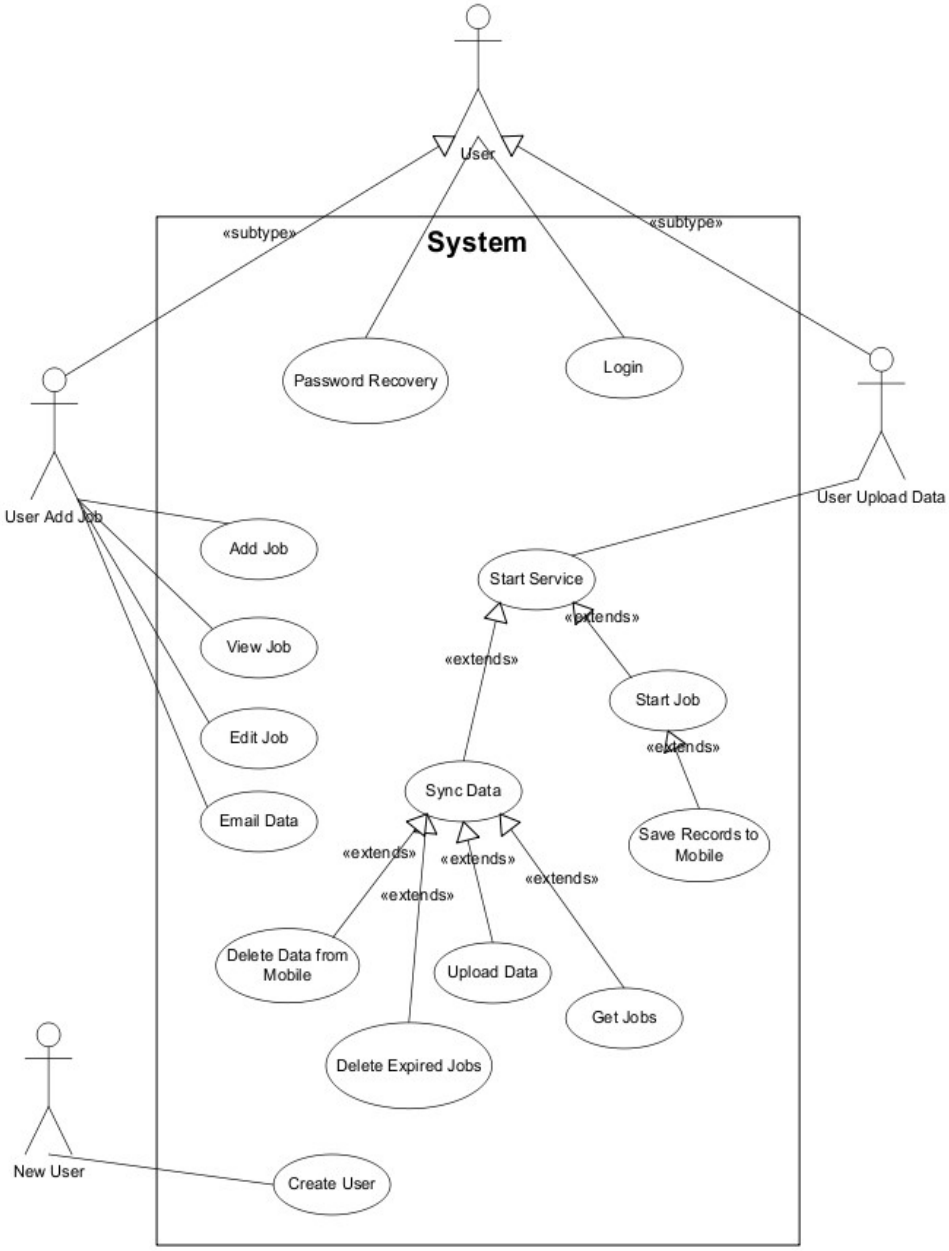

Fig 2.3: Use case diagram for overall system

Below entity relationship diagram illustrates the identified entities of this system and the relationship between the those entities. Cardinality of those relationships are indicated with relationship.

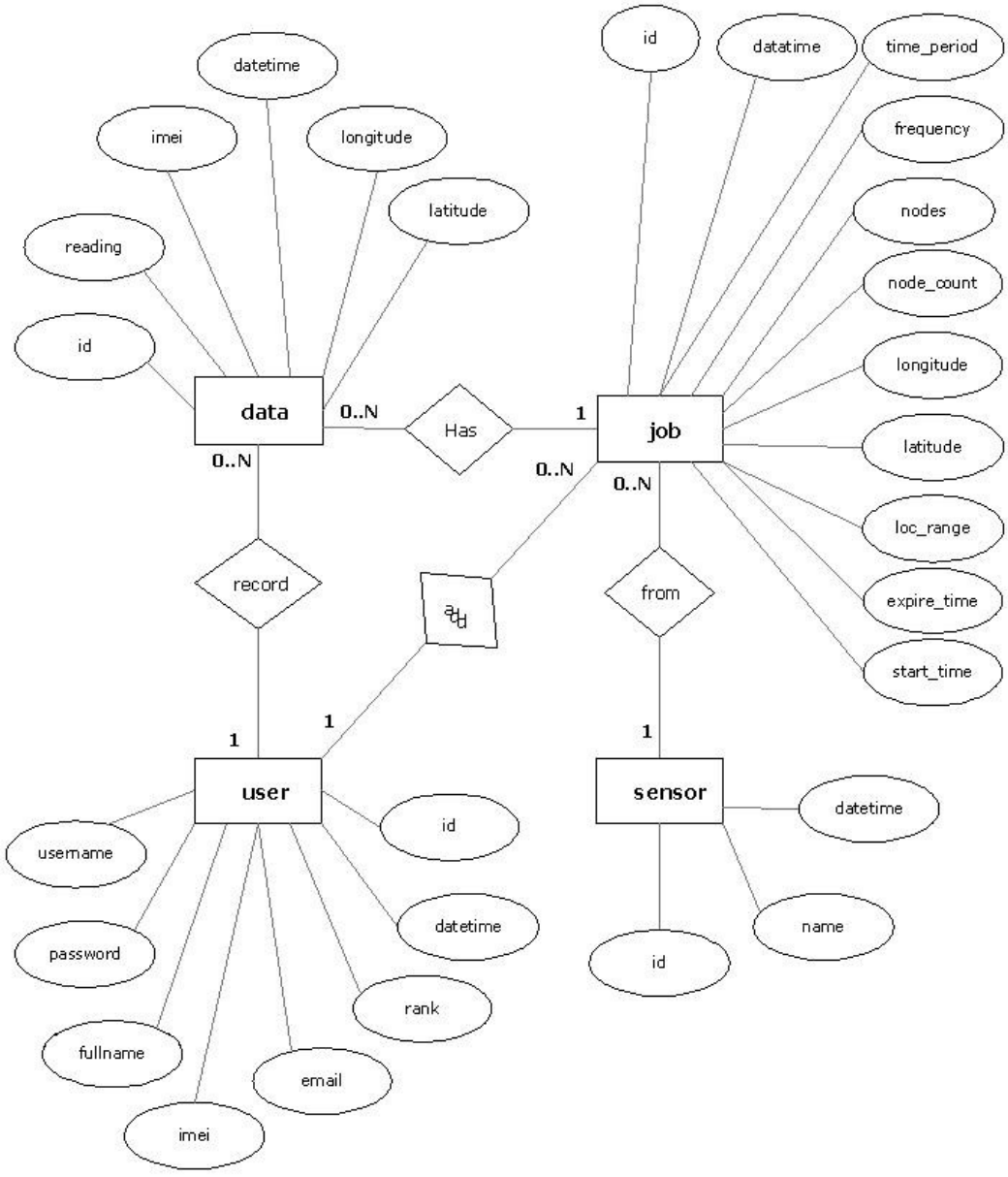

Fig 2.4: Entity Relationship diagram of the system

Following are the high level functional requirements of the application.

2.2.1.1 User Login

This is the first screen of the mobile application. Every user need to have an user account to login to the system. If not they should be able to create a new user account using Register new user screen. And if user forget or lost his password and user name there should be a password recovery methodology in order to gain access to his account again. All the password authentication and create new user account function

should be implemented and the application will invoke the back end service for all of them. This does not use its own local database. This is only for mobile applications security procedure, and there should be separate layer of web service security at back end.

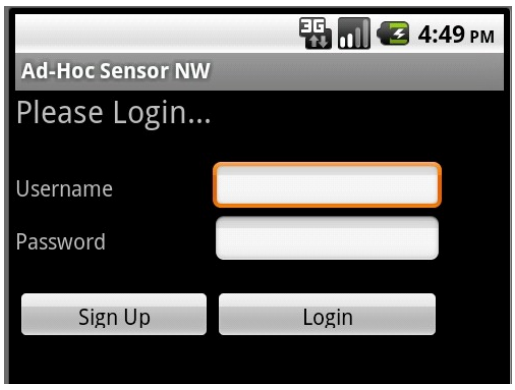

Fig 2.5: Login screen wireframe

2.2.1.2 Add Job

This will enable user to add job into queue. This will invoke the back end web service to add this job into the server database. The UI screen will consist of following fields to be added to the DB.

- Sensor Type – Which sensor data need to be recorded
- Latitude – Base location latitude of the data to be recorded
- Longitude – Base location longitude of the data to be recorded
- Location range – Radius of the area that use to collect data
- Start time – When to start recording data
- End time – When this job going to expire
- Frequency – In what frequency data need to be recorded in minutes
- Time period – How many hours does it need to record data
- Nodes – Number of nodes to be participating in collecting data
- Description – Additional information

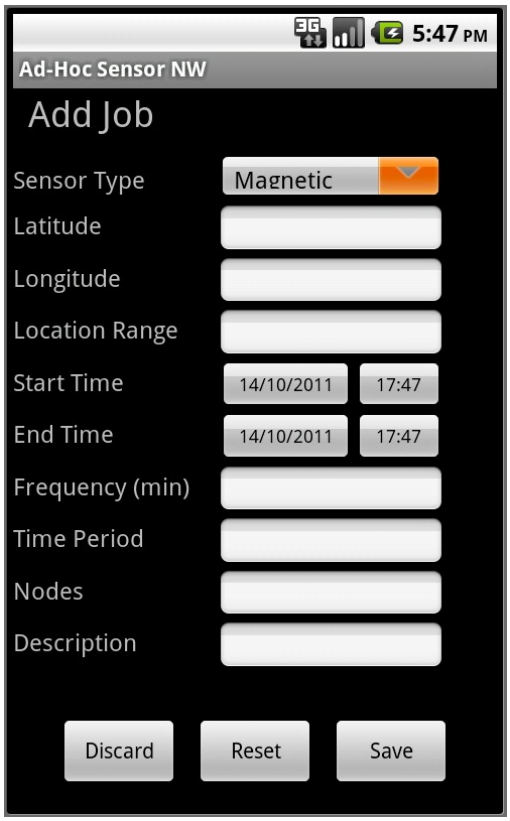

Fig 2.6: Add Job wireframe

2.2.1.3 View/Edit Job

This enable users to view and edit the job that he added. First user can view the jobs. Then he is able to edit them before they get expired. User can edit all information except the sensor type. Because a job totally depend on the sensor type and changing the sensor type is more or like adding new job into the queue. For both view and edit operation it should invoke the server implementation to get those data.

2.2.1.4 Synchronize

This is one of the most important functionality on this system. Because this will synchronize the local system, that is mobile application with the centralized server which is the back end web service. User should be explicitly can invoke this function to synchronization his device with server. And also other operations may be also invoking this function to keep device updated. This operation need to do following tasks in order to complete synchronization.

i. Upload data

First it should check whether there are any records in the local 'data' table. If there are any, those data should be uploaded into the server and once server updated successfully all the record on the local 'data' table should delete. Because otherwise local database will gradually fill up the mobile devices memory space.

ii. Delete job records

If there are any invalid jobs (that is expired or on hold jobs) in the local 'job' table delete them all. Because since those are invalid jobs, they are not going to execute in the mobile device.

iii. Get new jobs

Here first it check whether there is a valid job on the local 'job' table. And if there is, it should continue working on that job and persist data on that job. And if there isn't any jobs on the local 'job' table then it should invoke the server to get new jobs from the server. And continue work on them.

2.2.1.5 Record Service

This is the main functionality of this application. That is to read and persist the sensor data according to the job. This a quite complex task and following points will illustrate the functionality.

i. Once user touch to start recorder service this should start separate new worker thread to start work on reading and uploading data. It should run independently and record data according to the specified job.

ii. First thing it should do once user command to start recorder service is to synchronize the mobile device with the server. Because may be the device will be stale state.

iii. Now there should be new valid jobs on the local 'job' table. Therefore it will grab that job and start working on it. It should persist the sensor reading and other relevant information in to the local 'data' table with the specified job frequency. Each time, before it persist data into the local database it must make sure that this job is valid with in specified time range and specified location range. If it is not satisfy that criteria it will make the job as expired or on-hold according to the

situation. And continue get some other jobs available on server.

iv. There must be specified upload timeout, and once it reach upload timeout it will upload all the records on the local 'data' table into the server and clear the local table to avoid unnecessary space problem on mobile device.

v. If there is valid job, but it suppose to start somewhere in future then it should go into sleep mode until it reach the time to start that job.

vi. Also if there isn't any job found according to the location and the capabilities then the service will not need to be alive and it should go into sleep mode for specified time period and check back again after it wakes up.

2.2.1.6 View Data

This will enable user to view the data recorded by the different users for that particular job. But since there are small space available at mobile device screen, its should not displayed the information on this view data screen.

2.2.1.7 Email Data

Usually mobile devices have small screens compared to computers. Therefore its little difficult to display large amount data on those small screen. Therefore user and request the system to send his data to the email address specified by him. Then it becomes convenient for the user to do what ever he want with those data. Email send operation should be done at the server side

2.2.1.8 User Ranking Process

There are many users are adding jobs and contributing to other to collect data. Therefore it is necessary to have some kind of process to identify the users who contributes a lot and make their jobs priority when they need to collect some data. This will encourage other to contribute and grow this community forward. This can be done as and every time user uploads data into the server it should increment integer associated with the user. This integer can be called as rank of the user. This will maintain the stability and fairness of the system.

2.2.1.9 Server Side Implementation

There should be a server side back end to cater user requests. All the data should be finally uploaded into that server. Then users can look into their data from the server or request it to email. Users should submit their data requirement, that is a job into the system. This server should manage all those jobs from different users and distribute those jobs to the most suitable contributors to start work on them. And contributors have to upload readings back to the server. All the communication with node should be happen through that back end service.

2.2.1.10 Exception Handling

There should be proper exception handling mechanism. Because this mobile application interacts with a separate server and therefore it need to properly propagate exception from server side to mobile application which is to front end of the system. Otherwise front end will not know what happened at the back end and it will freeze the application till get response from the back end.

2.2.1.11 Service Level Authentication

All the mobile devices are accessing a single server. And there should be way to authenticate the application that invoke the services. Otherwise anyone from outside can invoke those service and they can misuse the system very easily.

2.2.2 Non-Functional Requirements

i. Application need to use the sensor and the battery optimum way. Otherwise it will leads to drain the devices battery very quickly. Which make it really hard time for user to charge his device frequently.

ii. Because of the mobile application is running on mobile device, it consist of limited battery, limited processing power and limited memory. Therefore it should be really careful when implementation to make it 100% optimized. Otherwise it will eat up all the resource of the device and it will ruin the device.

iii. Since this is running on the background this should not effect the normal usage of the mobile phone. Simply this should not disturb any of the users activities.

iv. Sensor network integration need to be done seamlessly that the mobile device user should not feel that there is running some other application on his device.

v. The UI layer should follow the proper standards. Which will make it more familiar to the users. Therefore users must be able to adopt to the system very easily.

vi. Server side need to be alive all the time. Because without it, mobile application can not do anything. Therefore it is very important to make sure that service is up and running all the time.

vii. When hundreds of uses are connecting the service, there may be high capacity transferring from server to devices, which will make the service slow down at the mobile end. Therefore server side must have enough bandwidth to cater large number of users connecting at once.

2.3 Review and Compare

You also can compare and contrast the system against other similar systems to get pros and cons of the system.

3 Chapter : Methodology

Ad-hoc sensor network are very much important to the research who are involved in different kind of researches. It does not restricted to particular research interests, but it is used and still using in total different subject areas. Therefore it will be important to create some system out of those requirements.

3.1 Proposed Solution

The main objective of this is to build a sensor network using mobile devices by considering above requirements. In this context a mobile phone will be just a sensor mote which can play the traditional sensor motes role. Therefore once the prototype application implemented with the above specified requirements, it can be used to collect relevant sensor readings as for users criteria. Once user specified the location and the time frequency of the data needed, application will upload this job to the server. Then server will try to find relevant devices to do that job. For this it will use the devices in build GPS sensor. When it finds a device it will send that job and device will take over the job and start recording data. First it will persist data into its local database. Then it will upload that data into the centralized server in batches. This will make use of devices data connection or the wifi network to upload data. But since its not using throughout while time, it saves most of the devices battery life.

Therefore this effort here is to use public mobile devices as sensor network. That is todays most of the mobile devices like phones, tablets, game consoles can collect data, persist to databases, find location information and communicate over Internet. But it can not do this task using factory mobile it self. It needs some sort of application to manage device communication, data transfers user jobs and everything. Thats why it needs an application and that is where android becomes the best platform to implement the application on.

3.2 Design Overview

As for the requirements this system need to be an client server system, which allow all the client device to connect into the server to get user requests ans upload data into the system. The client will be a mobile device application and the service will be a web service. Mobile application needs to be deployed on mobile device and the web service needs to be deployed

on high available web server. This web service is the heart of the system. It manages everything and mobile application execute the task assigned by the service. Then all mobile devices frequently communicate with the server and feed data and relevant information.

Since all the mobile devices are independently communicating with the server it will avoid the issue of single point of failure. That is one device failure did not effect to fail the whole system. And mobile device can use it own data connection to connect into Internet and the communicate with the server. It does not need any infrastructure. As well as mobile device can user its GPS modules and other sensor when it needs them.

Following diagram illustrates overall layout and the architecture of the system.

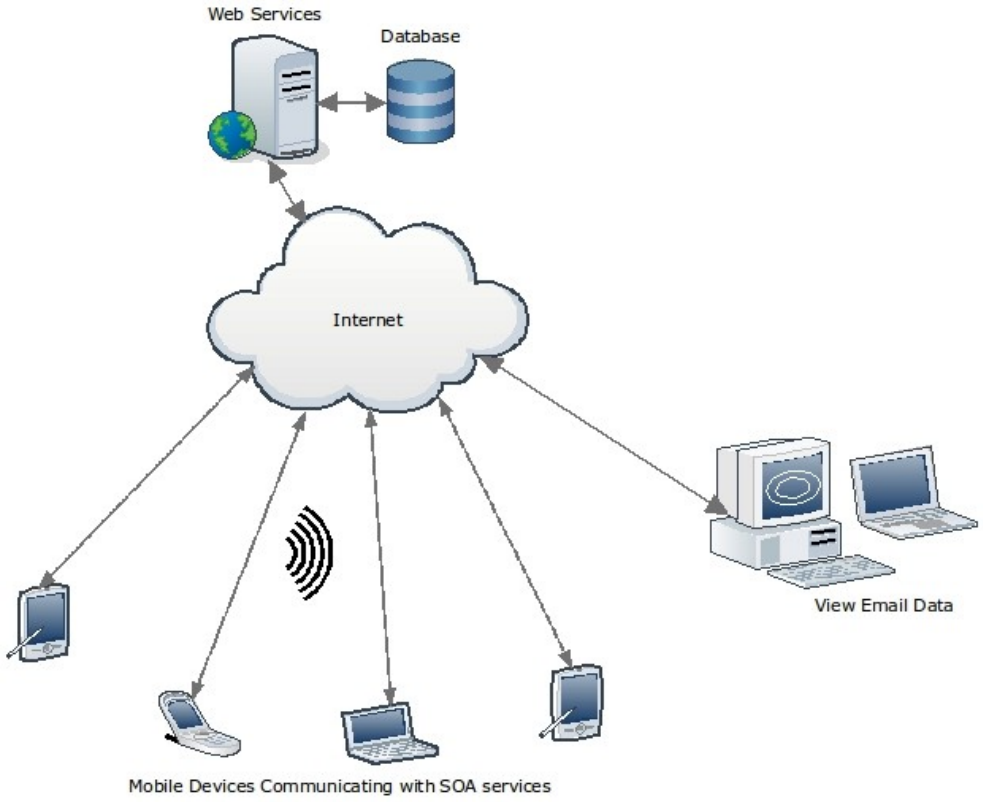

Fig 3.1: System Overview

3.3 Front End Technologies

Front end of this system is the mobile application which runs of the user devices. This application executes user jobs. And this is totally depends on the centralized server and it always interacting with the server.

3.3.1 Suitable mobile platform

Today there are many popular and sophisticated mobile device platforms based on different hardware vendors. Therefore finding the best suited platform for this purpose has become important to the success of the project. Following are the most popular mobile device platforms today,

- Apple iOS
- RIM BlackBerry OS
- HP WebOS
- Google Android
- Samsung Bada
- Microsoft Windows Mobile
- Nokia Symbian Belle

After considering most of the mobile device platforms, Android has become the best suitable platform for this purpose. There were many reasons behind to choose android over others. They are,

i. Free and open source platform by Google.

ii. Android comes with most of the mobile phone manufactures and service provider. Which means it will not make any restrictions.

iii. Now Android holds close to 50% of the smartphone market share all around the world. Therefore it will help to reach most of the smartphone users in the world.

iv. Wide range of devices from low end mobile phone to high end tablet. It will helps to find different audience.

v. It does not need any specific hardware or software to work on the platform or to develop application on top of it.

vi. Android basically uses java for application development. Therefore it makes easy to adopt to work on it.

vii. Eclipse integrated SDK is freely available for application developers on android. It makes the implementation a quite simple.

viii. There is a strong development community for Android who are willing to help beginners. And also there are lot of on-line resources and book written for android development.

Therefore based on the above mentioned reasons Android became the most suitable platform to implement the prototype application to covert mobile phone into sensor node. And it is decided to use Android 2.2 Froyo as the targeted version for the application.

3.3.2 Development Environment

Eclipse is the best development environment for the android. Its free and open source IDE and one of the most widely used IDE in Java community. There is fully featured SDK for the Google android and it also includes emulator which can be used to test most of the android applications. Also there is android plug-in to eclipse which is call Android Development Tools (ADT). With those tools, android SDK, ADT and eclipse make it really comfortable to start writing application on android platform.

Android comes with light weight database called 'SQLite'. Therefore this can be used to data persisting operation to keep data on mobile device. But this can not be used to store data for permanent usage, since mobile device have limited storage capacity it is necessary to upload data into the centralized server frequently and clear the local SQLite database to save space.

It is not enough to test android application only on emulator. Since this application involved on hardware sensors its hard to completely test it on emulator. It is necessary to have android mobile device to test sensor reading, location services etc. Not only that, it is very important to test an application on real mobile device to test aspects like performance and battery life. Therefore android mobile phone called 'Samsung T959 Galaxy S' is used as the primary test device to test this application. And some other random devices were used to test the compatibility of the application.

As the back end of this application is a Axis2 web service there need to be way to invoke web services from the mobile application. There is library to support in that case, which is called KSOAP2. The ksoap2-android project provides a lightweight and efficient SOAP library for the Android platform. Therefore it avoids all unnecessary overhead of invoking back end.

3.4 Back End Technologies

In this system server side back end is the heart of it. All the processing and decision making happens here. And android application or the front end just executing the orders from the server. Therefore server side implementation also need to be more efficient and flexible. Following technology stack has becomes more appropriate after considering the requirement which it suppose to cater.

i. Servlet Container : Apache Tomcat 6.0

   Apache Tomcat is the most popular and widely used open source servlet container. And it is a simple & light weight server which can be use with Axis web service engine.

ii. Web Service Engine : Apache Axis2 1.4

   Axis2 is also an open source and light wight web service engine. This is a complete re-design and re-write of the widely used Apache Axis SOAP stack. Apache Axis2 supports SOAP 1.1 and SOAP 1.2, and it also has integrated support for the widely popular REST style of Web services.

iii. Database : MySQL 5.0

   As a free and open source project, MySQL is the first choice as the database of this system. Even though it has few paid versions for commercial use, it provides full-featured database management system which is more than enough for this kind of system. And there is really nice front end for MySQL called 'MySQL administrator' and 'MySQL query browser'. This make its even easy to manage this database.

iv. Eclipse with Web Tools plug-in (WTP)

   Eclipse IDE is the obvious choice for Java developers. And with WTP it has full support to developing web service and other j2ee application. It is a hassle free environment to work with Java application.

3.5 Implementation Details

Further implementation details on important components of the system are described below. Those diagrams and pseudo codes will illustrate the logic behind the functionalities.

3.5.1 Recorder Service

This service is running on the mobile device. It is a background service which will run by itself upon user command to start service. This is the basically do record all the sensor reading into the local sqlite database with all other information, according to job specified.

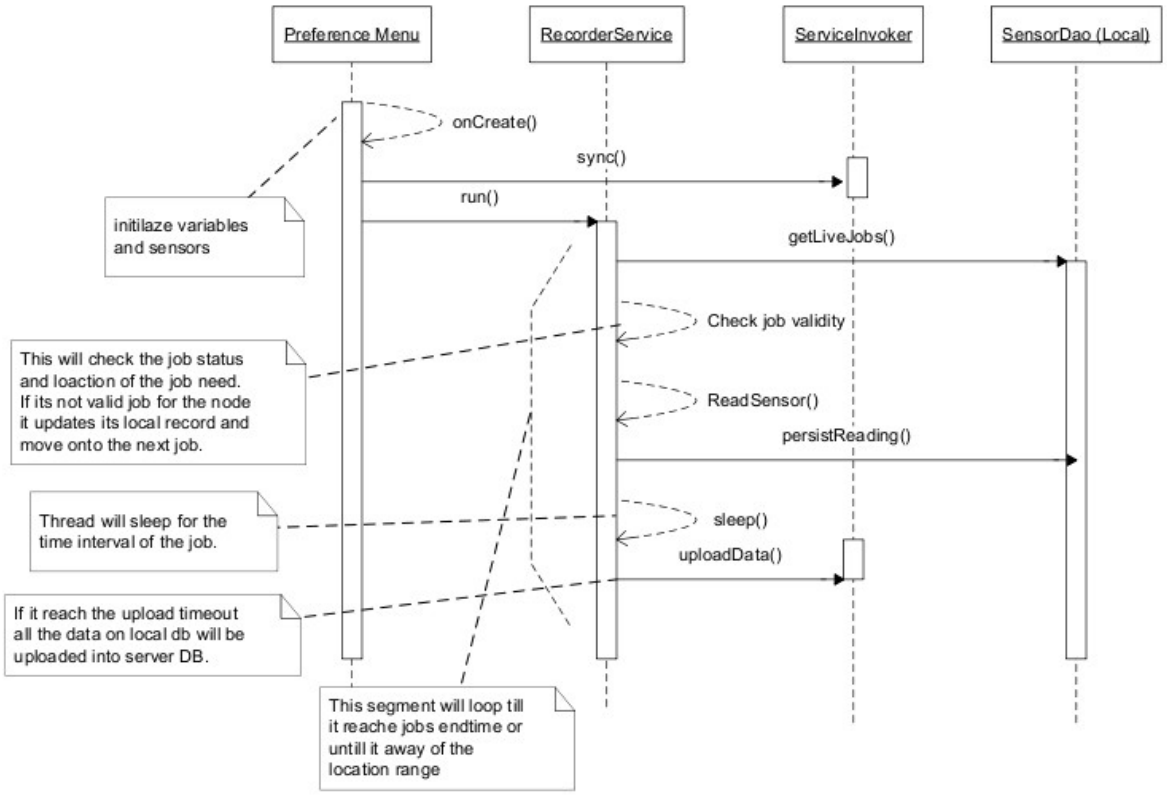

Fig 3.2: Recoder Service Sequence diagram

Pseudo code

```
Initialize sensors and variables
Initialize available location service (GPS provider or network provider)
Sync local database with back end
While service is active
Get live jobs from local database
If job found choose first record
      While job status is alive
            If upload timeout reached upload data
            If job is in valid location range and time range
```

```
                persist sensor reading data into local database
                thread sleep for the job frequency till next reading needed
        else
                if  job expired or out of location range
                        update the correct job status
                        sync local db with back end
                        get live jobs from local db
                        if there is any live job select the first job
                else if job is yet to start
                        sleep the thread till it start it
                        update the job status
else
        sleep thread for constant amount of time
        sync local db with back end
```

3.5.2 Add Job

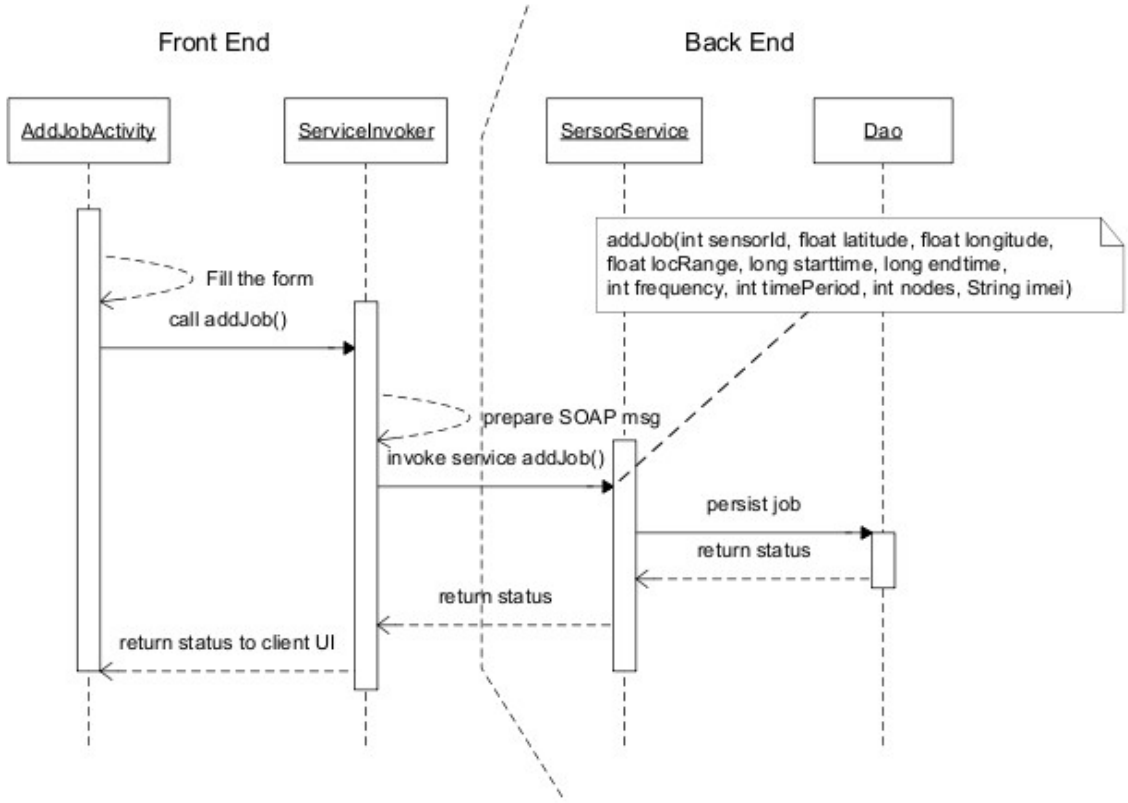

Fig 3.3: Add Job Sequence diagram

This will add an entry to the job table as users' data requirement. Mobile application directly invokes the back end web service and persist the job record. It will not persist anything to local database.

3.5.3 Select Jobs

Get jobs is one of the important operations on the back end. Because it is the one who select the most appropriate mobile node for a particular job. Following sequence diagram, pseudo code illustrate the basic logic behind this functionality.

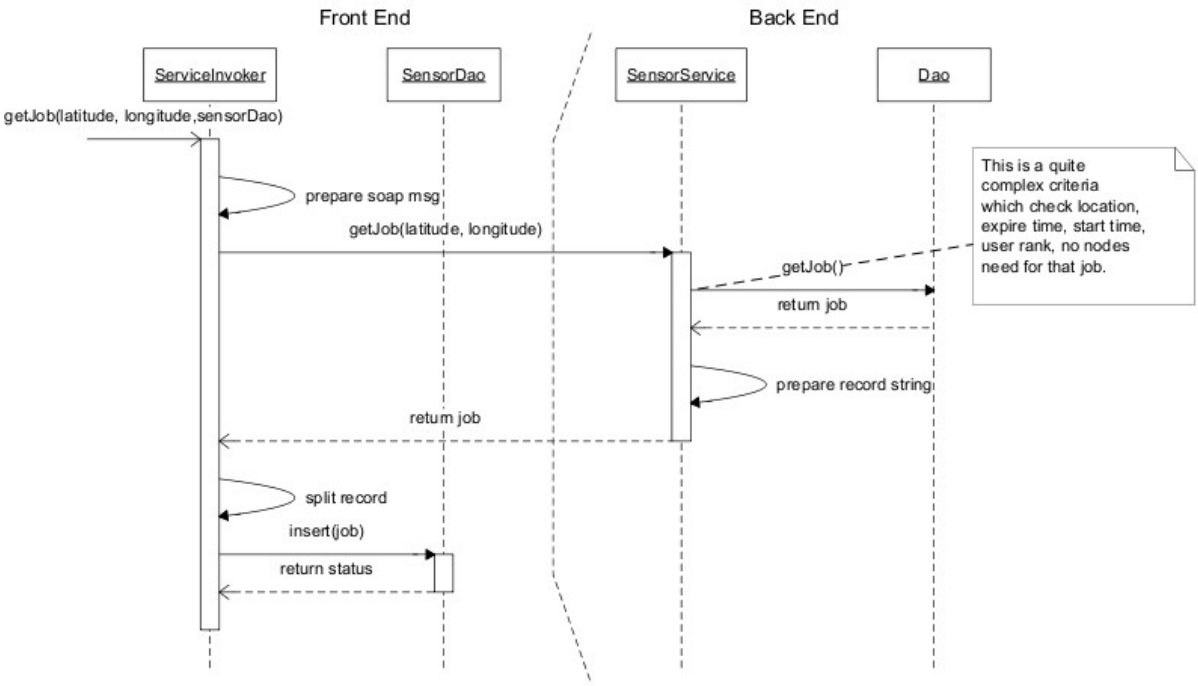

Fig 3.4: Get Job Sequence diagram

i. When selecting the appropriate job it is very important to check the location range of the mobile node with job location. Therefore to find distance between two GPS location is calculated using following formula. All these longitude, latitude values need to be from radians. R is the radius of the planet.

   location1 = (lat1, lon1)

   location2 = (lat2, lon2)

   dist = arccos(sin(lat1) * sin(lat2) + cos(lat1) * cos(lat2) * cos(lon1 - lon2)) * R

ii. Number of nodes that needed to complete the job, should be greater than the node_count which is the number nodes currently do that job. That is this job still needs more nodes to do its work.

iii. Job expire date must be satisfied. It should fall in future.

iv. Job start time should satisfied following rules

```
if job start_time is null
      start_time = now
if job start_time is not null and its already in past
      start_time = now
if still start_time is in future
      it will be the start_time
```

v. Job expire time should satisfied following rules

```
If expire_time is null
      expire_time will be (start_time + time_period)
if expire_time is not null and (start_time + time_period) < expire_time
      expire_time = start_time + time_period
else
      expire_time will be itself
```

vi. First order the record set by start time ascending order to find the closest jobs to the current time. And then order records by user rank descending, to get the highest rank users job with priority within the closest jobs.

vii. DateTime on sql can not assign as null it will always be "0000-00-00 00:00:00" therefore we need to check for "UNIX_TIMESTAMP(field)=0". Because UNIX timestamp of a time will 0 when it has 0 values.

3.5.4 Synchronize

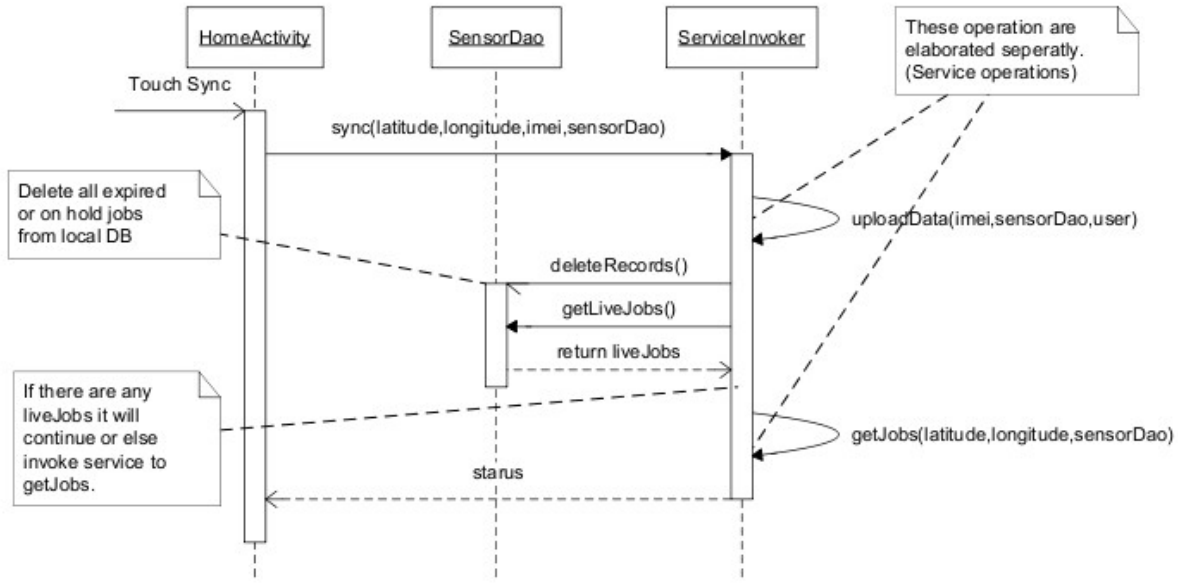

Fig 3.5: Synchronize Sequence diagram

This is used to synchronize the local database with the centralized database. It uploads all records on the data table and delete the invalid jobs from the local database. Then get valid job from the server and insert into local database.

3.5.5 Upload Data

This will upload the data from local device to the centralized database. Each time application uploads data into the server it will increment the user rank to mark him as a valued contributor. Following sequence diagram explains its activities from mobile device to the back end server.

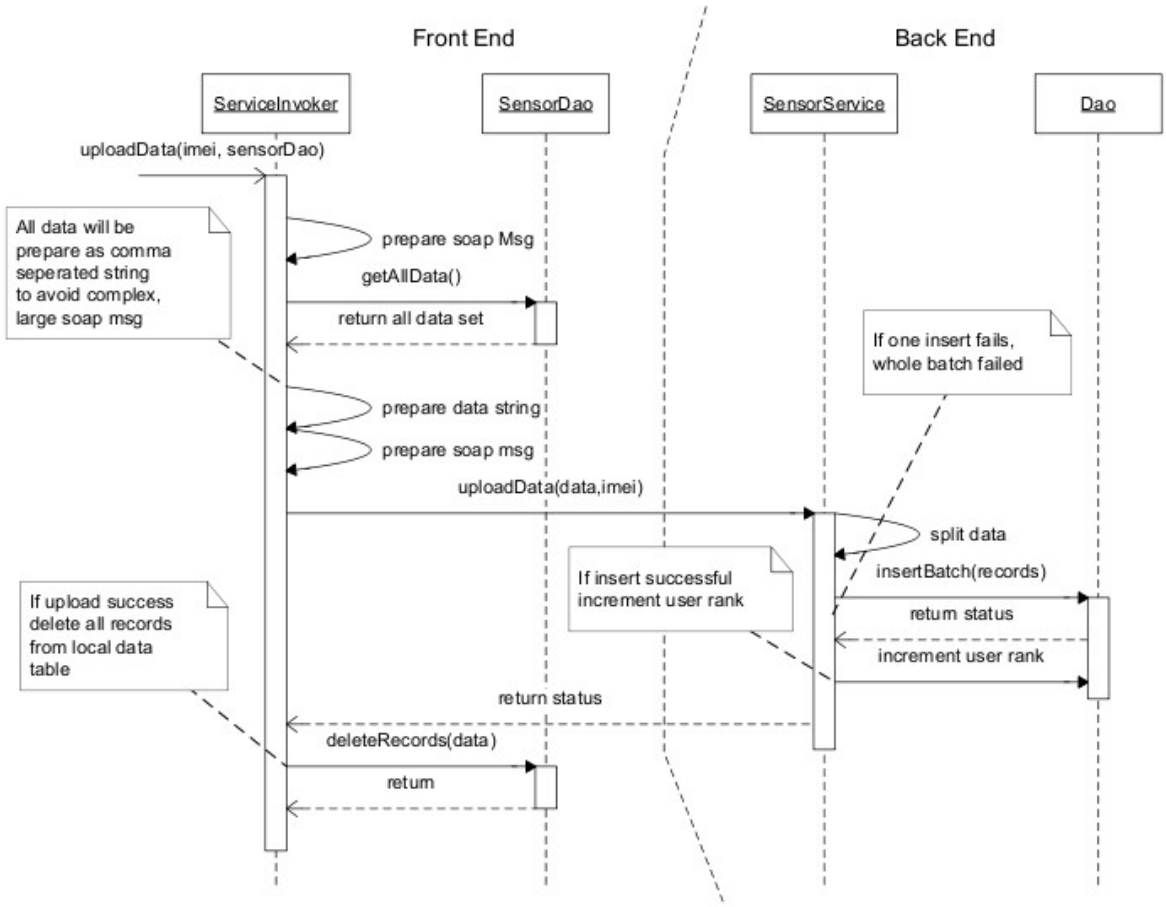

Fig 3.6: Upload Data Sequence diagram

3.5.6 Database Design

This is the database table structure for the system. Both front end and back end duplicate this database schema. But local application is not using all the tables and the field of the schema. And it is important to have database on local device, because otherwise it will have to connect to server all the time. This will lead to eat up memory, CPU time and battery of the mobile device.

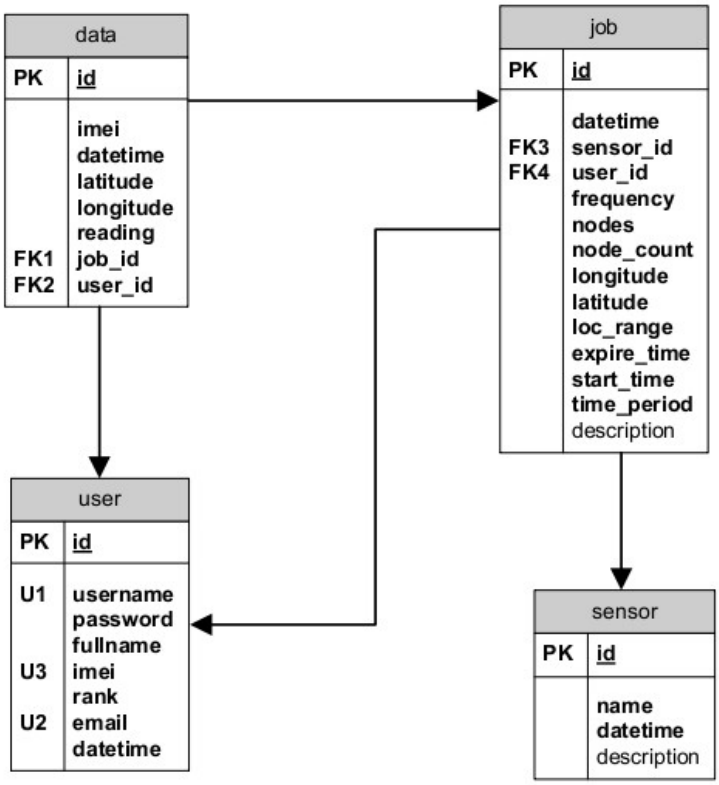

Fig 3.7: Database structure

4 Chapter : Evaluation

This is the chapter 4 and the final chapter of the dissertation.

4.1 Project Plan

Time duration for this project is about one year or 300 hours of time. Therefore we need to manage time very well to achieve expected target on this time line. In order to manage project to achieve the target it is necessary to break down tasks into individual components and assign time for those and monitor whether they are achieved on the expected time. If not necessary action needed to be taken to mitigate the rick of project being delayed.

5 References

Xiang-Yang Li (2008). *Wireless Ad Hoc and Sensor Networks: Theory and Applications*, First Edition, Cambridge University Press, ISBN 0521865239.

Cordeiro, C.D.M. & Agrawal, D.P. (2006). *Ad Hoc & Sensor Networks: Theory and Applications*, First Edition, World Scientific Publishing Company, ISBN 9812566821.

Ableson, F., Collins, C. & Sen, R. (2009). *Unlocking Android: A Developer's Guide*, First Edition, Manning Publications, ISBN 1933988673.

Burnette, E. (2009). *Hello, Android: Introducing Google's Mobile Development Platform,* Second Edition, Pragmatic Bookshelf, ISBN 1934356492.

Hashimi, S.Y., Komatineni, S. & MacLean, D. (2010). *Pro Android 2*, First Edition, Apress, ISBN1430226595.

6 Web References

[WWW 01] http://en.wikipedia.org/wiki/Smartphone [12/02/2011]

[WWW 02] http://en.wikipedia.org/wiki/Wireless_sensor_network [12/02/2011]

[WWW 03] http://www.computingunplugged.com/issues/issue200410/00001398001.html [03/03/2011]

[WWW 04] http://www.sensor-networks.org/ [03/03/2011]

[WWW 05] http://en.wikipedia.org/wiki/Sensor_node [03/03/2011]

http://www.sqlite.org/

http://developer.android.com/sdk/eclipse-adt.html

http://www.eclipse.org/webtools/