A Framework for Patient Monitoring

A. L. Praveen Aroul, William Walker, Dinesh Bhatia
Department of Electrical Engineering
University of Texas at Dallas
Richardson, Texas 75080
{alp052000, wpw021000, dinesh}@utdallas.edu

Abstract—**A comprehensive health and patient monitoring and data recording system has been developed using wireless sensor networks and supporting device and web infrastructure. This paper describes the components of the overall system. This includes the network architecture, node architecture, software infrastructure for recording data locally on servers as well as on Microsoft HealthVault data repository.**

Keywords-Wireless Sensor Network; ZigBee; remote monitoring; tele-medicine; data repository

I. INTRODUCTION

Rapid rise in the size of aging population combined with the rise in the healthcare costs is demanding cost effective and reliable patient monitoring systems. Monitoring patients and maintaining their health records is important for effective disease and health management. Technologies that support effective patient monitoring will not only reduce burden on the existing healthcare system but also mitigate healthcare cost impact on the overall economy of the nation. Several solutions that make use of various modalities of communication have been proposed. In this paper we describe our ongoing efforts towards building cost effective, low power, and ubiquitous patient monitoring systems. We describe our vision, supporting architectures and, software infrastructure for the patient monitoring system.

II. OVERALL SYSTEM DESCRIPTION

Our patient monitoring system is organized as follows. Section III explains the node architecture in detail. This section also describes the Activity Monitoring framework that was built on the node architecture. A discussion on the network architecture is provided in Section IV. Section V explains the different bio-potential sensors such as %SpO_2, ECG, Blood Pressure that have been interfaced to our network. The final section describes our progress towards interfacing our network with Microsoft® HealthVault™.

III. NODE ARCHITECTURE

The node architecture of our patient monitoring system is depicted in fig. 1. We have developed sensor node hardware "Biote", with a considerably smaller form factor. The node architecture is realized using state of the art sensors and wireless transceivers. One of the main hardware features of the work is the timely and efficient usage of the power and energy for acquiring, processing and RF transmission of the sensed event. Latest developments in design and fabrication of transceivers and sensors have created devices with ultra-low power capabilities thereby facilitating system implementations that can very well endure extended periods of on time on a small button sized 3V battery.

A. Hardware Architecture – wrist module

The Biote is built into a wrist module which houses an accelerometer, different bio-potential sensors interfaces, a microcontroller, and an RF communication transceiver. The various subsystems in the wrist module are explained in the following sections. A prototype hardware platform that is used for collecting data is illustrated in Fig. 2.

1) Sensing Subsystem

The sensing subsystem module includes a tri-axial low-g accelerometer, BMA150 from Bosch Sensortec [1] with digital output allowing both static and dynamic acceleration measurements. This sensing device was primarily used for the Activity Monitoring (AM) framework. The sensor can provide g measurement ranging from $\pm 2g$ to $\pm 8g$. For the accelerometer sensor employed in the AM Framework system, we have used a range of $\pm 2g$. The accelerometer sensor is interfaced to the MSP430 microcontroller using the serial peripheral interface (SPI).

The sensing subsystem also includes the various bio-potential sensors that are interfaced to the Biote through Universal Asynchronous Receiver and Transmitter (UART) interface of the MSP430 microcontroller. The bio-potential sensor interfaces are explained in detail in section V.

2) Processing Subsystem

The main processing unit is a TI MSP430F2619 microcontroller [2]. This device hosts a 16-bit RISC CPU, a wide range of integrated intelligent peripherals which enables a reliable sensor interface, 120 KB ISP Flash and 4KB RAM space for intermediate buffer during the processing or transmission. The processor also provides five low power modes which enable us to achieve an extended

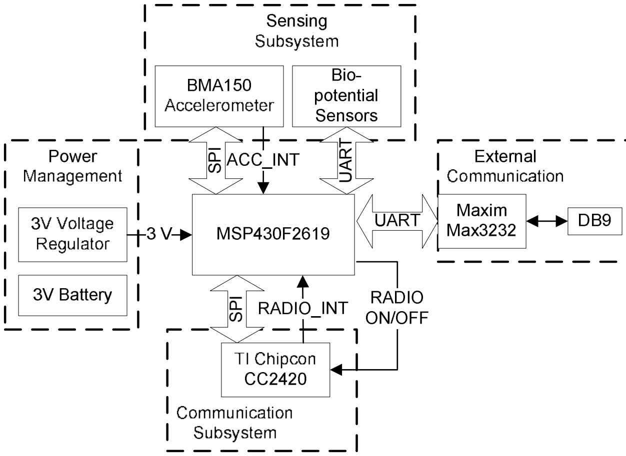

Figure 1 Node Architecture

Figure 2 Prototype Hardware Platform

battery life which is required for our system. The on-chip digital controlled oscillator (DCO) allows the state transition to wake-up the device from one of the low power modes to active mode within a short interval of time of less than 1 us.

3) Communication Subsystem

The communication module has a TI Chipcon CC2420 RF transceiver [3]. This module is an IEEE 802.15.4 and ZigBee-2006 compliant RF transceiver specifically designed for low power applications. Operating on the 2.4 GHz unlicensed industrial, scientific and medical (ISM) bands, the CC2420 provides extensive hardware support for packet handling, data buffering, burst transmissions, data encryption and authentication, clear channel assessment, link quality information and packet timing information thereby off-loading the work-load from the MSP430 microcontroller. The radio transceiver is also interfaced to the MSP430 microcontroller using the serial peripheral interface.

B. Software Infrastructure

1) Operating System and Communication Protocol Stack

The TI-Z-Stack has been used as our communication protocol stack. It is fully compatible with the ZigBee 2006 specifications [4]. The reason behind selecting TI Z-Stack is that it enables low cost, low power, reliable device monitoring and control, ensures efficient usage of the available bandwidth in devices, provides a platform and implementation for wirelessly networked devices and enables “out of the box” interoperable devices. The TI Z-stack utilizes a light weight operating system abstracted as a layer in the Z-stack device architecture. Operating System Abstraction Layer (OSAL) of the TI Z-stack provides scheduling, memory management and messaging features.

2) Step Recognition Algorithm

For the AM framework, to count the number of steps taken by the elderly / patient inside the experimental sensor network, we have utilized a step recognition algorithm. This algorithm exploits the tri-axial accelerometer BMA150 from Bosch Sensortec [1] to collect the acceleration values and calculates the step count on the end device. The end device could be worn as a wrist watch or attached to the leg of the patient / elderly.

The algorithm has the flexibility to estimate the step size taken during the day-to-day patient / elderly activity. This includes the lazy-walk, walk, and jog features. These values could be calibrated either using training performed on an individual patient or using approximate case values taken from a large-scale database. This database is created collectively using numerous samples of patients / elderly.

The step count and the type of activity of the elderly / patient are then transmitted to the coordinator over the RF interface. The coordinator receives this data and transmits to the health server or the mobile gateway based on the processing requirements.

IV. NETWORK ARCHITECTURE

The network topology contains a coordinator node, multiple router nodes and mobile sensor nodes. As the entire network needs to be covered with the availability of redundant paths, multiple nodes that have routing capability are deployed into the system. The coordinator node is connected to the base station / gateway or the mobile gateway, which is linked to the central health server for storage and processing. The coordinator node initiates the network, allowing other nodes to join the network by issuing the PAN address and the device address and also performs the routing operations similar to a router node. The coordinates of the router nodes (FFD) are considered to be fixed and mains powered in the network scenario, in other words, a static routing infrastructure is employed. Since the sensor nodes (RFD) are mobile, they are battery powered and as a thumb rule to maximize the battery period, the sensor nodes should be in sleep mode (in ultra low power mode) for most of the time. It is imperative that ultra low power operations are guaranteed at all times for obtaining a prolonged lifetime of the batteries on these mobile sensor nodes. The topology of this network is as shown in fig. 3.

The router nodes' locations are pre-configured and fixed and these nodes are powered using the traditional means. These fixed router nodes form a mesh network with the availability of redundant paths to increase the fault tolerance in the system and they connect the mobile nodes to a base station or a mobile gateway for relaying the information gathered from the mobile sensor nodes to the central health server.

For the purpose of activity monitoring that requires patient / elderly movement around the network, the sensor nodes are designed to be mobile. The mobile sensor nodes that can be incorporated as a wrist watch, can acquire, process, transmit the sensor data to the base station.

For a wireless sensor node, the majority of the power is consumed during the transmission and reception of data on the wireless radio. Hence to conserve power, it is necessary that the radio be switched to sleep mode when it is not performing any data transfer. At the back end of the mobile sensor node, we have the low power IEEE 802.15.4 compliant radio transceiver. It operates in the sleep mode for the majority of the time slice and only turns on the radio in case of an unexpected event, joins the network and then starts transmitting the sensor data to the base station. For applications which require constant monitoring using the wireless sensors, the life of the battery is also a critical issue. Power being a vital resource in this case, for a successful operation, the patient / elderly cannot monitor the health of the batteries and replace them once they fail. So it is crucial that the battery lasts for long periods of time before being replaced.

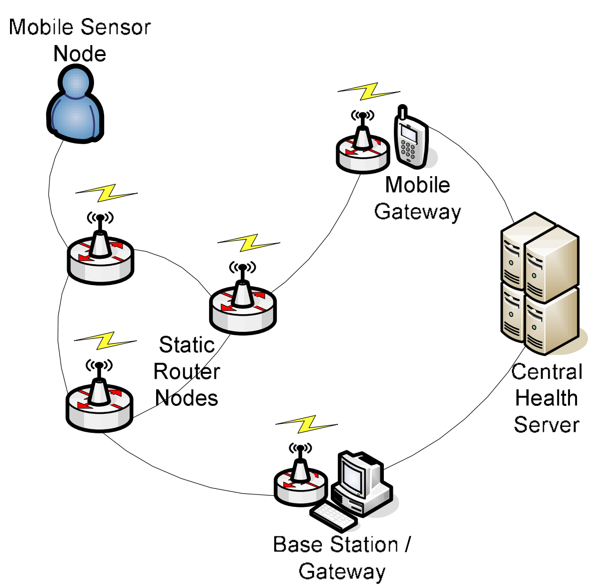

Figure 3 Network Topology

V. BIO-POTENTIAL SENSOR INTERFACES

A. %SpO_2 Data Acquisition

A commercially available Smiths Medical PM 31392B1 Micro Power Oximeter Board [5] was used to collect patient oxygen saturation (%SpO_2) readings for the system. The oximeter takes continuous %SpO_2 and pulse rate measurements. It includes a serial port connection that facilitates communication at 4800 bps. A Crossbow wireless sensor node communicates with the pulse oximeter on this serial link and receives the patient's %SpO_2 and heart rate readings. Once the readings are received, the sensor node communicates with the network and transmits them to the base station. The prototype hardware is shown in fig. 4, where the micro pulse oximeter, custom interface board and the Biote wireless sensor node can be seen.

Unlike the blood pressure system described in [6] which required commands to initiate and record a reading, the pulse oximeter communication is unidirectional, with no required commands. It continuously sends reading data through the serial link at a rate of 60 packets per second, which is much higher than the data rate for the blood pressure system.

It was determined that transferring all of the data points to the central monitoring station was not necessary to provide a valid graph of the patient's %SpO_2 and pulse rate. A data rate of once every 6 seconds was chosen as the initial rate, which should provide an accurate picture of the patient's status. All of the data is received by the wireless node attached to the pulse oximeter; however it is filtered and forwarded at the rate chosen to the central monitoring station. The wireless node also periodically checks its battery level, and will forward a low battery warning to the monitoring station if applicable.

B. ECG Measurement

A commercially available MedLab GmBH EG1000 one channel ECG module [7] is used to collect patient ECG data. The ECG module continuously monitors the cardiac rhythm

Figure 4 %SpO_2 Data Acquisition Unit

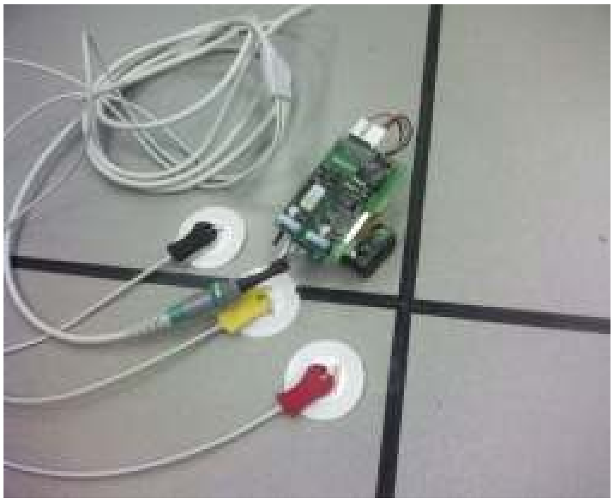

Figure 5 ECG Measurement Unit

and makes periodic measurements of the patient's pulse rate. The ECG and pulse data are sent out of the ECG module using an on-board serial port. One Biote sensor node is connected to the ECG module via the serial port. The prototype hardware is shown in fig. 5, where the ECG module, custom interface board and the Biote sensor node can be seen.

The ECG sampling rate is programmable. EG1000 supports three sampling rates of 50, 100 and 300 samples per second. The sampling rate is controlled by sending a few bytes of control words via the serial port to the ECG module. The data is transmitted at 9600 baud, 8 bits, 1 stop bit, and no parity. While capturing patient data, each time a pulse is detected by the module's algorithm, a block with a new, averaged pulse rate is transmitted. The ECG sample points are transmitted continuously at a rate which was last programmed by the user. As the ECG is continuously monitored, both the module and the sensor node attached to it are continuously on. Every time a certain number of ECG data bytes are received by the node, a transmission packet is created and the node transmits the packet over the network. Hence, the rate at which the node transmits each packet over the network depends on the sampling rate of the ECG module. Higher the sampling rate of the module, higher is the packet transmission rate.

C. BPM Interface

The UA-767PC BPM [8] provided patient BP and heart-rate readings for the system. It includes a serial port that facilitates bi-directional communication at 9600 bps. The Biote node was programmed as the sensor node to communicate with the BPM on this serial link to start the reading process and receive the patient's BP and heart-rate readings. Once the readings were received, the sensor node communicates with the network and transmits them to the base station. The prototype hardware is shown in fig. 6, where the BPM module, custom interface board and the Biote sensor node can be seen.

Figure 6 Blood Pressure Interface Unit

To start the communication process with the BPM, the sensor node sends a start signal to the BPM to switch it into communication mode and opens the BPM communication port. The BPM is now ready to receive commands from the sensor node. Next, the sensor node issues a command to take a measurement. This causes the BPM to inflate the arm cuff, and acquire the BP and heart-rate measurements. When the reading process has completed, the readings are sent to the sensor node. Limited processing is performed by the sensor node on the data before transmitting it through the network to the base station.

Fig. 8 shows the communication sequence between the sensor node and BPM to acquire a reading. The communication format is in ASCII format and is described in [9]. Initially a “Turn on” message (0x55) is sent from the sensor node to the BPM. This byte orders the BPM to wake up and be prepared to receive commands from the sensor node. Next, the “Open Comm Port” message is sent to open the serial port of the BPM. Here, the first byte (0x02) indicates that the message is a command from the sensor node. Bytes 3 and 4 (0x50 and 0x43) inform the BPM details of the device sending the message. In this case, it is the sensor node and is defined as “PC” in ASCII. The following two bytes (0x30 and 0x35) are the command to open the BPM communications port.

The BPM responds to the sensor node's “Open Comm Port” command by transmitting an “Acknowledgement” message that begins with 0x01 which indicates that the message is a control message. Once again, the next two bytes describe the device sending the message. Since this message is being sent by the BPM, it is represented as “70” in ASCII. The following two bytes detail who receives the message which is the sensor node represented as “PC” in ASCII. The final byte (0x06) indicates that the control message is an acknowledgement.

Figure 7 BPM serial port example messages

After receiving an acknowledgement from the BPM for its “Open Comm Port” command, the sensor node instructs the BPM to take a measurement. The “Take Reading” message follows the same format as the “Open Comm Port” message. ASCII “10” is sent as the command bytes. The BPM responds to this message by outputting a “Data” message. The first two bytes of the “Data” message are fixed and represent “80” in ASCII. The next two bytes represent the hex value of the systolic BP reading minus the diastolic BP reading in ASCII. From fig. 7, it can be seen that this value equals “3C”. The following two bytes are the ASCII representation of the hex value for the diastolic BP reading. This value also equals “3C” in ASCII. Therefore, in this case, the systolic BP reading of the patient is 120 and the diastolic BP reading is 60. The following two bytes are the ASCII representation of the hex value for the heart-rate. From fig. 7, it can be seen that the patient heart-rate is 60 beats per minute. In the present system, the measurement process is started when the sensor node is turned on.

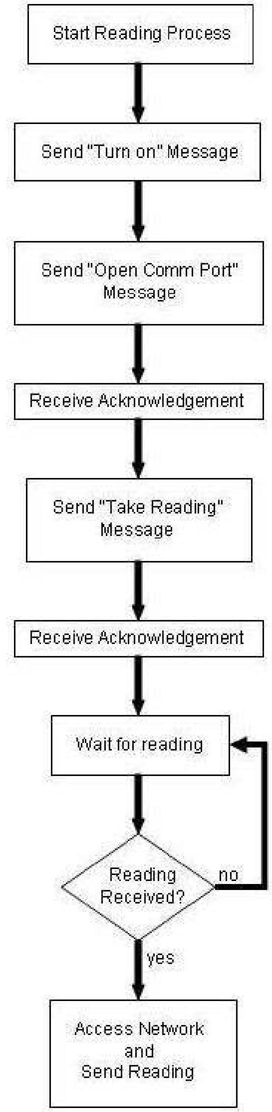

Figure 8 BPM Communication Flowchart

VI. MICROSOFT® HEALTHVAULT™

The purpose of this section is to describe the progress made towards interfacing our network with Microsoft® HealthVault™ [10]. A brief discussion on Microsoft® HealthVault™ is given, followed by a description of our system. Included in the system description is a discussion of the problem that our system attempts to solve and a basic requirement that our system must satisfy. Next, a discussion on the interfaces created for the system and details on the interface operation is provided. This section is concluded with a description of the system operation and what future steps are planned for its implementation.

A. Microsoft® HealthVault™ Basics

Microsoft® HealthVault™ is a free, online database used to store and share health related information. Currently, all HealthVault interaction is web-based, offline access is not supported. In a web-based application, a certificate identifies the intended application only, whereas in an offline client application, a certificate could be used by any program running on the client machine. The intention is that when a user logs onto HealthVault from an application, access is granted to only the registered application.

In HealthVault, all information concerning a single person is stored in a Record. A Record can contain personal information such as name, nickname, address, ethnicity, etc. and health information such as blood pressure readings, glucose level readings, known allergies, etc. From a Record, it is possible to view and change other records depending on the permissions granted. For example, a parent can create a Record for self and Records for children. In this case, the parent would have access to view and change/add values to the children’s Records. Records can be shared as well. An example of a shared Record is when siblings share their Records with each other.

B. System Implementation

In our system, we are interested in the monitoring of persons from the standpoint of a Master Record. The term "Master Record" refers to a person who is interested in monitoring one or more persons. We are examining two types of relationships in this system, ***(a)*** a doctor/hospital to patient relationship and ***(b)*** a family member to another family member relationship, where the family member being monitored is in need of constant monitoring. This system can be depicted as a tree with the Master Record at the root and the patients/family members as the leaves. In both of these cases, information in the Master Record is minimal and is used as a means to retrieve/add information from/to the monitored person's Record.

The setup for our initial testing of this system is the doctor patient relationship. An example problem that our system is designed to solve is the following. How can a doctor monitor a patient's blood pressure through the patient's normal daily activities at the patient's home? Note that the choice for monitoring blood pressure is arbitrary as it could have been blood glucose levels, temperature, etc. Such a system must provide ease of access, both by the patient sending the blood pressure readings and by the doctor viewing the readings. Interfaces are designed to hide the complexity of the system from the patient and doctor.

C. Basic System Interfaces

There are five basic types of interfaces used in our system ***(1)*** sensor to wireless sensor network, ***(2)*** wireless sensor network to an internet accessible device, ***(3)*** internet accessible device to the website, ***(4)*** human to website, and ***(5)*** website to the HealthVault database. This section covers interfaces *(3)*, *(4)*, and *(5)*. Interface *(1)* is discussed briefly in the network architecture section concerning the connection between the blood pressure monitor, pulse oximeter, and ECG module and our wireless sensor network. Interface *(2)* provides a means for the data received from these sensing devices to be sent through the internet. At some point in the wireless sensor network is a device that is capable of communicating with the wireless sensor network and the internet. There are multiple ways for this interface to be implemented. One way is through the use of a handheld device such as a PDA or cell phone with IEEE 802.11 connection capability. Another way is to connect one of the wireless motes, called the base station, to a PC through the serial port on the PC and connecting the PC to the internet via an Ethernet or wireless internet connection.

While interface *(4)* allows the user to input data to the website and to HealthVault via interface *(5)*, interface *(3)* is a key to satisfying the requirement of making the system easy to use by the patient. Interface *(3)* is created using a TCP Socket connection between the internet accessible device and the web server hosting the system website. This interface allows data to be sent directly from the wireless sensor network without requiring the patient to perform any action. Currently, this is a simple Socket connection which includes no security measures other than the default security of the TCP connection. Future iterations of this system will include added security measures and a means of identifying the location of the sender (as it stands, any device that has the proper server address and port number can send information to the web server). A specific port is assigned to each HealthVault user upon logging into the website. Any data received along the port is processed into a HealthVault data type for uploading via interface *(5)*.

Interface *(5)* is created through the use of the HealthVault SDK provided by Microsoft. In order to send or receive any HealthVault information, the user must physically log into their HealthVault account. The HealthVault login page is displayed in Fig. 9 which appears when a user initiates the system website. Once logged in, the website has the ability to send or receive any HealthVault information accessible by the person who is logged in. Also, upon logging in, a TCP Socket is assigned such that any data sent along the assigned port can be processed and sent to HealthVault. Fig. 10 displays the page upon login. The user can select a patient from the patient list provided. Once a patient is selected, the user navigates to the patient view page, where readings can be viewed and added manually. Fig. 11 displays the patient readings page for blood pressure readings.

As it stands, our system is intended to work in the following manner. A care provider logs onto our website. Our wireless sensor network is set up at the patient's home complete with blood pressure monitor and a device with internet access. The patient can then take a blood pressure reading from anywhere inside the area of the wireless sensor

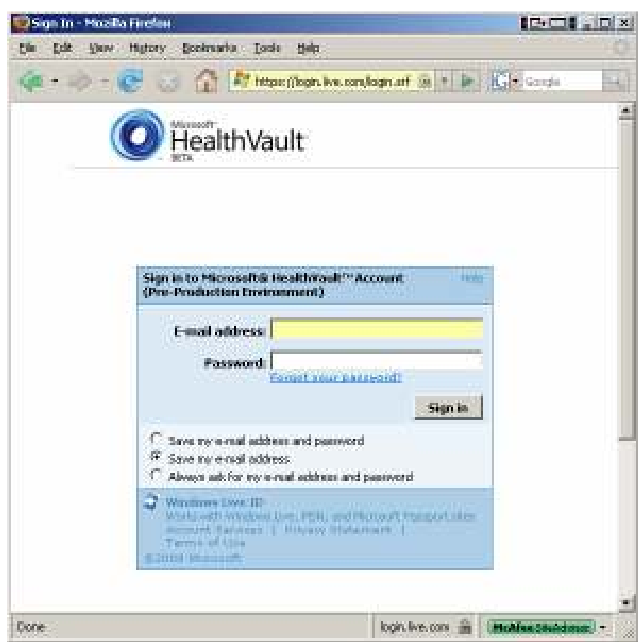

Figure 9 HealthVault Login Page

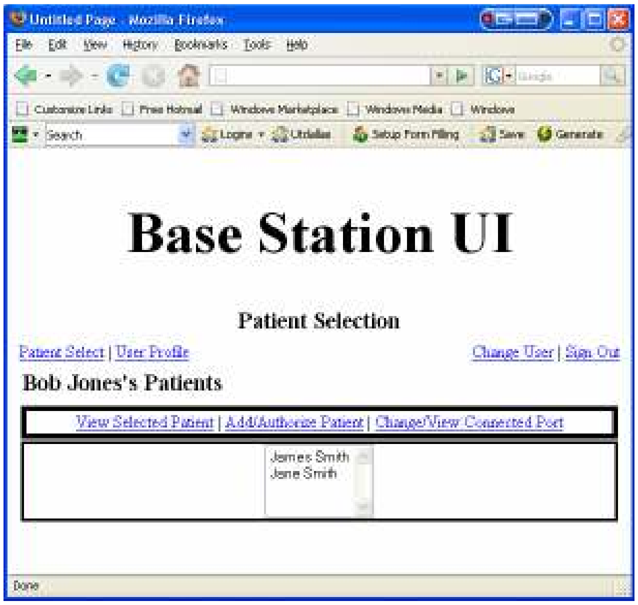

Figure 10 Doctor's Patients' List page

network. The blood pressure readings are transmitted across the wireless sensor network to the device with internet access. The device opens a communication port with our web server and transmits the blood pressure data. Upon receiving the blood pressure data, the web server processes the data and uploads it to HealthVault. The care provider is now able to view this new information by navigating to the desired patient's readings page.

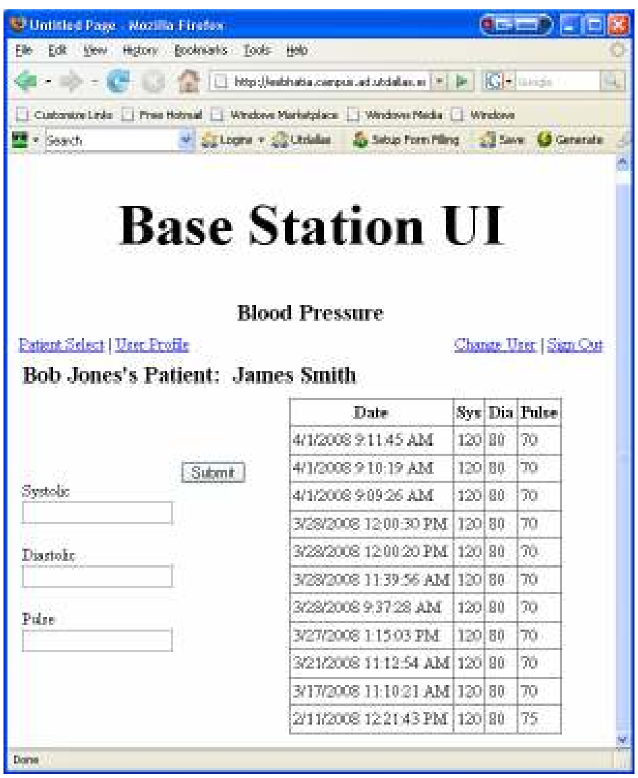

Figure 11 Patient's Blood Pressure Readings

VII. CONCLUSION

Aging is becoming the major and most impacting social issue. It is starting to put economic burden in the form of healthcare costs. Monitoring of health and cost effective disease management is the only way to ensure economic viability of the healthcare system. We have presented a comprehensive health and patient monitoring and data recording system that has been developed using wireless sensor networks and supporting device and web infrastructure. We have also described our vision, supporting architectures and, software infrastructure for the patient monitoring system.

ACKNOWLEDGMENT

The authors at UTD's Embedded and Adaptive Computing Lab would like to thank Texas Instruments Inc. for their support in carrying out this research.

REFERENCES

[1] Bosch Sensortec, MEMS Acceleration Sensors, http://www.bosch-sensortec.com.

[2] Texas Instruments Inc., 16-bit ultra low power mixed signal microcontroller, http://www.ti.com/msp430

[3] Texas Instruments Inc, 2.4 GHz IEEE 802.15.4 RF Transceiver, http://www.chipcon.com/files/CC2420_Data_Sheet_1_4.pdf

[4] ZigBee® Alliance, http://www.zigbee.org/en/index.asp

[5] Smiths Medical PM, Inc., "Micro Power Oximeter Board", http://www.smithsoem.com/pdf/Micro%20Power%20Oximeter.pdf.

[6] W. Walker, T. Polk, A. Hande, and D. Bhatia, "Remote blood pressure monitoring using a wireless sensor network", Proceedings of the IEEE Sixth Annual Emerging Information Technology Conference, Dallas, Texas, August 2006.

[7] Medlab GmbH, "EG1000 One Channel EKG Module Technical Manual", 1996, http://www.medlab-gmbh.de/english/downloads/eg01000.pdf.

[8] A & D Medical, Inc., "UA-767BT Wireless Blood Pressure Monitor", 2005, http://www.lifesourceonline.com/products/telemonitoring.cfm.

[9] A & D Medical, Inc., "UA-767PC RS-232 Command Sets and Data Format Specification v2.1", March 2004.

[10] Microsoft® HealthVault™, http://www.healthvault.com