TRACKING OF PATIENTS IN THE HOSPITAL USING RFID

By

AZREE FAIZAL BIN AZENAN

FINAL REPORT

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas

Bandar Seri Iskandar

31750 Tronoh

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

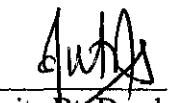
TRACKING OF PATIENTS IN THE HOSPITAL USING RFID

by

Azree Faizal Bin Azenan

A project dissertation submitted to
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
In partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:



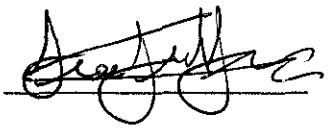
Pn Hanita Bt. Daud
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

Sept 2011

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and the original work contained herein have not been undertaken nor done unspecified sources or persons.



Azree Faizal Bin Azenan

ABSTRACT

The movements of patients in hospital especially Alzheimer's and mental disorder patients are monitored manually and these type of patients will have tendency to forget their exact location. If they were to go missing, it is very difficult to track and locate them. It is crucial to locate them as fast as possible because their condition is very fragile and there have been case that patients died when they were missing. The main objective of this research is to develop RFID based system that will be able to track the movement of the patients in hospital. This technology offers numerous advantages over the traditional method. The implementation of this project is each patient especially the Alzheimer's and mental disorder patients will be tagged with RFID tag and readers will be placed at strategic locations which will be monitored by nurses and other medical staffs. The nurses can monitor the patients by using software developed by combinations of programmes. With this technology, it can easily track the patients' location on real time and can prevent the patients from entering the unauthorized zone or section. As for current practice, the tracking methods are either visually or depend on surveillance cameras.

ACKNOWLEDGEMENTS

I would like to praise Allah the Almighty for helping and guiding me into the completion of my Final Year Project (FYP) in Universiti Teknologi PETRONAS. I would also express my deepest appreciation to my family for their encouragement and support for me to reach for this level.

My gratitude also goes to my project supervisor, Ms. Hanita Daud for her supervision and support throughout the process of the project. Not to forget Mr. Elisha Tadiwa Nyamasvisva for giving me the technical expertise in order to complete the project successfully.

Last but not least, my thanks also go to my fellow friends and individual who were supportive throughout the period of the project.

TABLE OF CONTENTS

ABSTRACT iv

ACKNOWLEDGEMENTS v

LIST OF FIGURES viii

LIST OF TABLE ix

CHAPTER 1: INTRODUCTION 1

1.1 Background of Study 1

1.2 Problem Statements 2

1.2.1 Problem Identification 2

1.2.2 Significance of the Project 2

1.3 Objectives of the Project and Scope of Study 3

1.4 The Relevancy of the Project 3

1.5 Feasibility of the Project 3

CHAPTER 2: LITERATURE REVIEW 5

2.1 RFID Technology Overview 5

2.2 RFID Tag 6

2.3 RFID Readers 10

2.4 Host Computer 11

2.5 Operating Frequency 12

2.6 Principle of RFID Operation 14

2.7 Application of RFID in Hospital Environment 15

2.7.1 Pharmaceutical Applications 16

2.7.2 Hospital and Medical Device Application 17

2.8 Tracking of Patients in Hospital using RFID 18

2.9 Software Based Application 20

CHAPTER 3: METHODOLOGY 21

3.1 Research Methodology 21

3.1.1 Phase One (Project Initialization) 21

3.1.2 Phase Two (Project Designing) 21

3.1.3 Phase Three (Prototyping) 21
3.2 Project Flow Chart 22
3.3 Tools Required 24
3.3.1 MySQL 24
3.3.2 Microsoft Visual Studio 2010 24
3.3.3 RFID Hardware and Specification 25
3.4 Project Duration 27
CHAPTER 4: RESULT AND DISCUSSION 28
4.1 Result and Discussion 28
4.1.1 System Overview 28
4.1.2 Graphical User Interface (GUI) 29
30
CHAPTER 5: CONCLUSION AND RECOMMENDATION 36
5.1 Conclusion 36
5.2 Recommendation 36
REFERENCES 37
APPENDICES 38

LIST OF FIGURES

Figure 1 : 13.56-MHz RFID tags 7
Figure 2 : Active Tag with the scale of coin 7
Figure 3 : Optimal and non-optimal tag and reader position 8
Figure 4 : Basic components of an RFID system 12
Figure 5 : Inductive and propagation coupling RFID systems 14
Figure 6 : Backscatter modulation 15
Figure 7: Healthcare positioning tag with reusable wristband with buckle 18
Figure 8 : Healthcare positioning tag with reusable wristband with lock 18
Figure 9 : Elpas RF BUS Reader-Ceiling Mount 19
Figure 10 : Elpas RF BUS Reader- Wall Mount 19
Figure 11 : Project Flow Chart 23
Figure 12: Microsoft Visual Studio 2010 25
Figure 13 : System Flow for Active Wave Hardware Used 26
Figure 14: Active Wave RFID Kit 27
Figure 15: The System's Flow 29
Figure 16: Patient Tracking System Login 29
Figure 17: Main Menu of the System 30
Figure 18 : Patient Detail 31
Figure 19 : Reader IP Configuration 31
Figure 20 : Add Staff Profile 32
Figure 21: Add Patient 32
Figure 22: Database for the Staff Registration 33
Figure 23: Database for the Patient Registration 33
Figure 24 : Alert Alarm Triggered for Entering Dangerous Zone 33
Figure 25 : Records of Alert History 34
Figure 26 : List of All Registered Patients 34
Figure 27 : List of All Registered Staff 35
Figure 28 : Patient Archive 35

LIST OF TABLE

Table 1: RFID Operating Frequencies and Associated Characteristics 13

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Radio-frequency identification (RFID) is a promising technology and one of the most rapidly growing segments of today's automatic identification data collection (AIDC) industry. However, this emerging technology is not new; in fact it is currently being applied in various applications throughout the world. It was originally implemented during World War II to identify and authenticate allied planes, in an identification system known as Identification, Friend or Foe, and is still being used today for the same purposes [1].

RFID reader obtains the information of objects and surroundings through communication with tag antennas. Readers and tags can communicate in a wireless way without being need to connect with any wire and cable. RFID can also identify mobile objects of high speed and it can identify a certain amount of tags simultaneously by its anti-collision mechanism.

An RFID system has three basic components which are tags, readers and host computers. RFID tags consist of tiny semiconductor chips and miniaturized antennas inside some form of packaging. Some RFID tags look like paper labels and are applied to boxes and packaging. Others are incorporated into the walls of injection-molded plastic containers [3].

Each tag is programmed with a unique identifier (ID) that allows wireless tracking of the object or person the tag is embedded to. The chip can include many information and data such as serial numbers, time stamps, configuration instructions, technical data,

medical records, and travel history. This is because the chip used in RFID tags can hold large amount of data.

One of the greatest obstacles to the wide adoption of any new technology is a standardization process. The purpose of standardization is to define the most efficient platform on which an industry can operate and advance. Several organization are involved in drafting standards for RFID technology but in looking at the present status, it seems it will be some time before all of the details are agreed on [1].

1.2 Problem Statements

1.2.1 Problem Identification

In recent years, the tracking for patients at hospitals has given the medical staff so many burdens and it has effect the workflow and the efficiency of the healthcare system. The Alzheimer's and mental disorder patient are the one who has tendency to be missing in the hospital facilities. In a study by Bowers et al (1999), 4% of people harmed themselves or other in some way following their absence from the ward, while a more recent report (Man Univ, 2006) found that no less than 27% of inpatient suicide take place off the ward, often after the person has gone missing [8]. Such incident creates a time-consuming work for nurses, medical staffs and other professionals (eg. police) (Bower et al 1999.c). Moreover, due to conventional method of tracking the patients, it will take some time before the patients can be found by the nurses. During that time, patients who need extra and full time care may be in serious condition and it may lead to serious casualty. In the current market, there is no way portable way to monitor the heartbeat of the patients if they are on the move.

1.2.2 Significance of the Project

The project system is an innovative tool for tracking the Alzheimer's and mental disorder patients in hospital. The system helps and guides the nurse and medical staff to track these types of patients who are have tendency to be missing in the hospitals. The system continuously track and scan the hospital area in zoning. Thus, makes the system

as real-time locating system (RTLS). The system helps to eliminate the event of missing patients and prevent the patients from entering the restricted zone.

1.3 Objectives of the Project and Scope of Study

The main objective of this project is to track the Alzheimer's and mental disorder patient in the hospital using RFID system.

The sub-objectives of this project are;

a) Provide zonal tracking of patients
b) Alert signal if the patient entered the dangerous room.

1.4 The Relevancy of the Project

In order to achieve these objectives, a few tasks and research need to be performed by developing the database using Microsoft Visual 2010, MySQL program and hardware of RFID. A recommendation is to be made based on the research and finding of the subject regarding the applicability of the RFID based patient tracking in Malaysian context.

1.5 Feasibility of the Project

This project comprise two semesters project such that more research and study will take part in Semester 1 (Part 1) while more modeling, prototyping, and testing part will take part in Semester 2 (Part 2). However, some part of modeling and developing the database and user interface application work can be started during Part 1.

There are two important things that need to be fully understood before starting with the project. The two important things are; the understanding of RFID technology and understanding of development platform. Therefore, this project is feasible to be carried out within the provided time and scope.

In term of time frame, the project is feasible as the complete integration between RFID device, VB application, and SQL Server Database is done. Thus it fulfills the

objectives of the project. The project is also feasible in term of cost because no cost implied for the Active RFID hardware since it can be requested from Universiti Teknologi Petronas (UTP).

CHAPTER 2

LITERATURE REVIEW

2.1 RFID Technology Overview

Generally, RFID represents a way of identifying objects or people using radio waves. Identification is made likely by means of unique numbers that identify object, people, and information, stored on microchips which can be read automatically. With the recent development and advancement in RFID technology, the automatic identification data capture industry is accelerating its effort to identify new applications to take advantage of RFID [1].

Similar to broadcast television and radio, RFID systems operates on four major frequency bands: low frequency (LF), high frequency (HF), ultrahigh frequency (UHF), or microwave bands. Why RFID do operated at different frequency bands? This is because the different frequencies have different propagation characteristics [1]. UHF band is the systems that are coming on the market today and the microwave band is the hot prospect in the future application. The most common method use for RFID is relying on storing and remotely retrieving data which involves the usage of tags, readers that collect data and manage it in a portable, changeable database [2].

Basically, RFID systems are consist of a few principle components and fundamental technologies that can make them work. The first main components of RFID system is RFID tag (transponder) which is located on the object to be identified and is the data carrier of the system. The second one is reader (transceiver) which may able to both read data from and write data to a transponder. The third one is data processing

subsystem which utilizes the data obtained from the transceiver in some useful manner [1]. However, it is must be noted that the operating frequency play a huge role on the functionality of RFID system.

RFID technology allows the data and information to be stored and read wirelessly without requiring the line of sight (LOS). It can be read inside a case, box, carton and etc. Furthermore, the tags can be read hundreds at a time by the reader which means it can do bulk reading of objects. There are two types of RFID systems, passive tags and active tags. A passive tag does not require an on-board power source as does an active one.

When within range of the reader, RFID tags are "interrogated" by and RFID reader. The reader generates a radio frequency "interrogation" signal that communicated with the tags. The reader also has a receiver that captures a reply signal from the tags and decodes the signal. More detailed description of the system will be followed.

2.2 RFID Tag

The RFID tags can be categorized into three different types: passive, active and semi passive. These kinds of tags have operating characteristics respectively and the means by which they receive power for transmission determines their type [2].

RFID tags may be powered by a tiny battery inside the tag (active tags) by an RFID reader that "wakes up" the tag to request a reply when the tag comes within range (passive tags).

RFID passive tags just rely on the operating power from current induces in their antenna through the signal waves generated by the readers to be functional. Therefore, the passive tags only operate at maximum length of 3 meters or less. The simplest of passive tags is capable of holding something in the range of 64-bits of factory-written unique data.

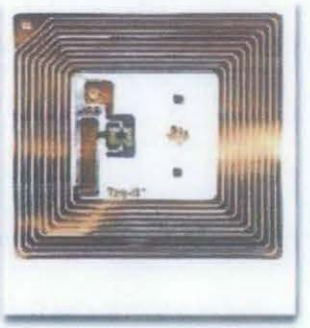


Figure 1 : 13.56-MHz RFID tags

Semi passive tags use battery to run the chip's circuitry, but communicate by drawing power from the reader [3]. Once triggered by the reader's signal, these tags utilize their own powers drained from the battery to execute their tasks.

Active RFID tags with their own power source can actively and intensively transmit and processing data. Active tags can communicate with reader 100 meters or more away. This kind of tag need less signal from the reader compare to passive tags. Therefore, active tags can contain sensors and data loggers because they are continually powered. Active tags are suitable to be data loggers because they can support a clock and can contain significant amounts of memory.

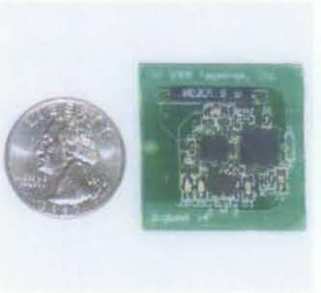


Figure 2 : Active Tag with the scale of coin

There are general RFID tag requirements whose quite importance depend on the application. These requirements majorly determine the criteria for selecting an RFID tag antenna [1]:

a) Frequency band

Desired frequency band of operation depends on the regulations and standard of the country where tag will be used.

b) Size and form

Tag form and size must be such that it can be implanted or attached to required objects.

c) Read range

Minimum required read range is usually specified.

d) Object

Tag performance varies when it is placed on different objects or when other objects are present in the vicinity of the tagged object. An antenna can be designed for optimal performance on a particular objet.

e) Orientation

The read range depends on antenna orientation. The way the tag is attached to the object with respect to the polarization of the reader's field can give a significant effect on the communication distance for HF and UHF tags [1]. The optimal orientation for tags is for two antenna coils (reader and tag) to be parallel to each other. Refer to figure 3 below.

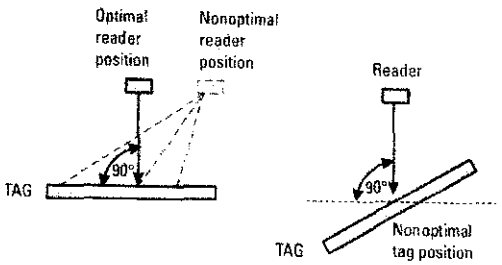


Figure 3 : Optimal and non-optimal tag and reader position

f) Application with mobility

RFID tags can be mobile during operation and it does not affect at all the RFID operation. However, the tag spend less time in the read field will demand a high-read-rate capability for the reader.

g) Cost

In one RFID system, there will be thousands of RFID tag being used. Therefore, the RFID tag must be a low-cost device thus dictating limitations both on antenna structure and on the choice of materials for its construction.

h) Reliability

The RFID tag must be a reliable device that can sustain variations in temperature, humidity, and stress and survive such processes as label insertion, printing, and lamination [1].

i) Power of the tag

An active tag carries its own battery and has greater range than passive tags. Passive tags rely on the reader for power to be functional, and semi passive tags rely on the reader for powering transmission but the battery for powering their own circuitry.

Basically, there are four different characteristics pertains data storage capability in RFID tag which are:

a) Data Capacity

As for example, a library tags typically have space for 256 bits of information which is more than adequate for current system requirements. Apart from that, some tags have room for up to 2,084 bits of information. Therefore, it can be said that this is one of the advantages in RFID system that overcomes the lack of information that can be stored in traditional barcode system.

b) Read/Write Characteristics

This characteristic allows information to be stored in the tag and then it can be updated and modified when it is needed. However, not all type of tags has this feature.

c) Password and Encryption

Data in the RFID tags are encrypted to provide extra and additional security to the user and system. Only the RFID readers that contain the encryption code can decipher and read the data from the encrypted tag.

d) RTF versus TTF

All RFID readers transmit constant signal that can power up the RFID tags that are within the range. In a system where the 'reader talks first' (RTF), the reader will transmit a second command signal that request the data from the tags. As a result, the tags response to the second signal from the reader and start sending identifier and pertinent data it stores from the reader [3]. While in the 'tag talk first' (TTF) event, the tags straight away reply to the reader's signal without requiring the reader to transmit second command signal as in RTF.

2.3 RFID Readers

The second component in a basic RFID system is the reader of interrogator. The term "reader" is a misnomer [4]. Technically, reader units are transceivers (a combination of transmitter and receiver). However, because of their role to query a tag and received data and information from it, they are seen as "reading the tag" and hence the term reader. Reader may be portable handheld terminals or fixed devices positioned at strategic points, such as entrance, assembly line, or toll booth. The reader is equipped with antennas for sending and receiving signals and a processor to decode data [1].

The reader extracts the information and data from the RFID tag and it may be self-contained and record the information internally. However, it may also be part of a localized system such as Local Area Network (LAN) or Wide Area Network (WAN). For this function, the reader includes an RF transmission for receiving and data decoding sections. In addition, the reader often includes a serial communication (RS-232, USB and so on) capability to communicate with a host computer [1].

RFID readers are consists of an antenna that communicates with the RFID tags and an electronics module that is networked to the host computer. The module relays messages between the host computer and all the tags within the antenna's read range, enabling one reader to communicate with hundreds of tags simultaneously.

Basically, depending on mobility, the RFID readers are classified into two types. The two types of the readers are:

a) Fixed RFID reader

   Reader read the tags in a stationary position and it is setup in specific interrogation zones and creates a "bubble" of RF energy that can be tightly controlled.

b) Mobile RFID reader

   Reader is mobile when reading the tags. Mobile reader includes handhelds reader and vehicle mounted RFID readers.

2.4 Host Computer

The host computer is very essential in any RFID system. It can take the shape of a personal computer (PC), laptop, server or a workstation as long as they run database and control software. The host computer is the brain of the RFID system and every information's collected from the tags are processed by the host [5]. The RFID reader will cross-reference the tag's data within its self-contained database. After the reader receives new data, it will send the data to the host. The readers and the host communicate through a secure wireless link.

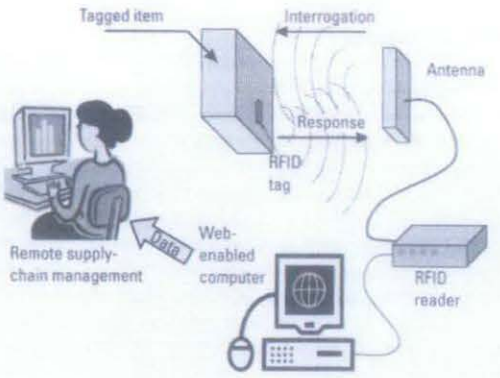


Figure 4 : Basic components of an RFID system

2.5 Operating Frequency

Different RFID systems operate at a variety of radio frequencies. Each range of frequencies offers its own operating range, power requirements, and performance. Different range may be subject to different regulations or restrictions that limit what application they can be used for.

Operating frequency is also important in determining the physical dimensions of the RFID tag. Different sizes and shapes of antennas will operate at different frequencies. Other than that, the operating frequencies also determine how tags physically communicate with each other. For instance, stacking flat foil inlay tags on top of each other may interfere or prevent tags from reading properly [6].

RFID operates in unlicensed spectrum space, sometimes referred to as ISM (Industrial, Scientific, and Medical) but the exact frequencies that constitute ISM may vary depending on the rules and regulations in different countries. These operating frequencies are generally considered to be organized into four main frequency bands and the table shows these different radio waves bands and the common frequencies uses for RFID systems (IEEE, 2005).

Table 1: RFID Operating Frequencies and Associated Characteristics

Band	**Low Frequency (LF)**	**High Frequency (HF)**	**Ultra High Frequency (UHF)**	**Microwave**
Frequency	30-300kHz	3-30MHz	300 MHz-3GHz	2-30 GHz
Typical RFID Frequencies	125-134kHz	13.56 MHz	433 MHz or 865-956 MHz	2.45 GHz
Approximate read range	Less than 0.5 meter	Up to 1.5 meters	433 MHz = up to 100 meters 865-956 MHz = up to 3 meters	Up to 10 meters
Typical data transfer rate	Less than 1 kilobit per second (kbps)	Approximately 25 kbps	433 - 956 MHz = 30 kbps	Up to 100 kbps
Characteristics	Short range, low data transfer rate, penetrates water but not metal	Higher ranges, reasonable data rate (similar to ISM phone), penetrates water but not metal	Long ranges, high data transfer rate, concurrent read of <100 items, cannot penetrate water or wall.	Long range, high data transfer rate, cannot penetrate water or wall.
Typical Use	Animal ID Car Immobilizer	Smart Labels Contact-less travel cards Access and Security	Specialist animal tracking Logistic	Moving vehicle toll

2.6 Principle of RFID Operation

The coupling between tag and reader is achieved in one of two ways depending on the carrier frequency used and the system antenna design. In these systems the field is effectively tied to its source and the field that couples with the tag is modulated by means of the tag circuitry, such that the data-related changes can be sensed by the reader [1]. Basically, the principle of RFID operation has two basic ways of exchanging information which are:

a) Inductive Coupling and Load Modulation

Inductive coupling means that the tag and the reader's antenna are coupled by the magnetic flux through both coils. When a tag is places within the alternating magnetic field created by the reader, it draws energy from the magnetic field. In load modulation the carrier signal is modulated by switching impedance from a matched condition to an unmatched condition to alter the reflection coefficient [1]. The systems are based on a transformer-type coupling between the primary coil in the reader and the secondary coil in the transponder.

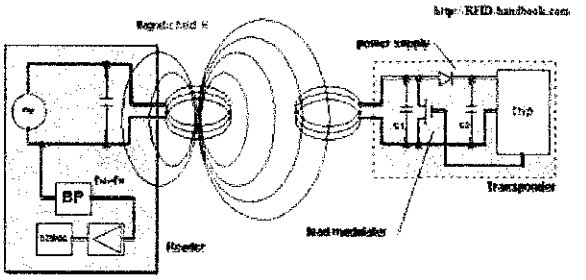


Figure 5 : Inductive and propagation coupling RFID systems

b) Propagation Coupling and Backscatter Modulation

The term backscatter modulation refers to the communication method used by RFID tag to send data back to the reader [1]. The power reflected

from the transponder is radiated into free space. A small proportion of this free space attenuation is picked up by the reader's antenna. The reflected signal travels into antenna connection of the reader in the backwards direction and can be decoupled using a directional coupler.

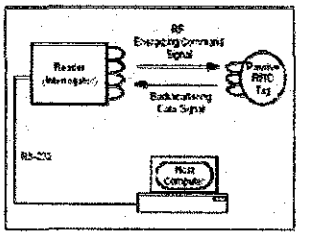


Figure 6 : Backscatter modulation

2.7 Application of RFID in Hospital Environment

This section aims to describe how the RFID technology can help in medical facilities and hospitals. To start with, many assets and actors of the facilities have to be tagged [9]:

a) The medical equipment must embed RFID tags.
b) The doctors, nurses, caregivers and other staff members wear a smart badge storing their employee ID number.
c) On arrival, each patient receives a wristband with an embedded RFID tag storing a unique identifier and some information about him
d) All the patients' medical histories and other important documents are tagged with self-adhesive RFID labels containing a unique number
e) The blister packs and other drugs' packages all contain RFID labels
f) The bags of blood are attached with a self-adhesive RFID label holding a unique identifier, the hospital tracking number and some important information about the contained type of blood

With all the assets and the actors of the facilities has been tagged. There are so many application can be done inside the hospital facilities [10]. The applications are;

2.7.1 Pharmaceutical Applications

a) Drug Counterfeiting

Pharmaceutical companies, distributors, and hospitals need technology to deter drug counterfeiting. The World Health Organization estimates that between 5 percent to 8 percent of global pharmaceuticals are counterfeit. Thus, the pharmaceutical industry reports that it loses $2 billion per year due to counterfeit drugs [11].

With RFID tags, they can help to detect products that are

- Counterfeit
- Tampered with adulterated or substituted
- Unacceptable (i.e, expired, discarded, returned, recalled, etc. [12])

b) Clinical Trials

RFID technology can improve the tracking of drug usage throughout the clinical-phase testing protocols. Improved tracking and accountability can improve the reliability and speed of the United States Food and Drug Administration (FDA) drug approval process

c) Inventory Management

Manufacturers and distributors need improved visibility throughout the supply chain to gain an accurate account of inventory. Lack of visibility of customer orders results in increased inventory because healthcare practitioners often keep buffer stocks to avoid stock outs. Increased inventory visibility could reduce buffer stocks by substituting knowledge for inventory, thereby reducing total inventory costs

2.7.2 Hospital and Medical Device Application

a) Medical Devices and Asset Tracking

RFID has strong application potential with medical device companies. The FDA requires medical device companies to be able to identify each unit by serial number. Medical device companies need better control of implants on consignment with hospitals because returns can occur more than 50 percent of the time. RFID technology that improves visibility into returns could enable faster redeployment since the company would know sooner when an unused product could be returned.

Surgical instruments and other devices must be properly cleaned and packaged between uses. Tags on the instruments and readers on the sterilization chambers and storage cabinets can validate proper cleaning and help locate needed instruments. Since medical devices are often mounted on portable carts, smart tags placed on the devices and readers installed in the doorways can enable personnel to quickly locate a crucial piece of equipment and immediately determine its fitness for use.

b) Patient Tracking

Patient identification and location assistance are often needed to ensure patient safety when urgent medical attention is needed. Patient tags with RFID chips will meet this need.

c) Product Tracking

Hospitals currently have to track radioactive isotopes throughout the facility from storage to transport and then from administration to disposal. RFID tags and readers can automate these tasks thereby saving time and resources.

2.8 Tracking of Patients in Hospital using RFID

Based on the products of Visonic Technologies (VT) which is Elpas Wandering Patients Protection Solutions, it is designed to reduce the cost of supervising ambulatory at-risk patients without excessively restricting their mobile independence. Elpas Wandering Patient Protection Solutions enables complete freedom of movement of care givers, guests, other patients or family members while preventing high-risk patients from leaving monitored areas without proper un-supervision [7].

Based on the products of Visonic Technologies the three major components that are essential for tracking of patients systems are;

a) Healthcare Positioning Tag

The Elpas Healthcare Positioning Tag is a wrist worn Active RFID that is an effective cost containment and risk management tool for acute care hospitals, geriatric care clinics and long-term nursing care facilities.

The tag's onboard RF/R dual technology transmitter emits low power RF (433 MHz) messages. This data enables the real-time visibility of medical personnel or patients down to bed-level precision without Electromagnetic Interference (EMI) to medical equipment.

Figure 7: Healthcare positioning tag with reusable wristband with buckle

Figure 8 : Healthcare positioning tag with reusable wristband with lock

b) Receiving Antenna (Reader)

The Elpas RF Bus Reader is supervised 433MHz fixed indoor RTLS receiver. The reader is designed to detect and relay real-time 'Location' and 'State' data from Elpas Active RFID Tags to Elpas Safety and Security software platforms or other hospital management systems.

The reader supports standard IT network communication and is easily integrated onto wired and wireless Ethernet Wi-Fi networks for relaying data to and from host computer. The reader supports large tag populations at read-distances up to 20m/65ft (360° coverage area) in open room environments.

The antenna or readers are placed strategically array throughout the facility to create blanket RF coverage. The readers will capture a passing patient's positioning tag ID number and transmit that number to the back-end system. Nurses and doctors can access the information regarding the patient's status on their PDAs.

Figure 9 : Elpas RF BUS Reader-Ceiling Mount

Figure 10 : Elpas RF BUS Reader- Wall Mount

c) Network of host computers

System will provide real-time tracking and can be monitored by a network of computers employing system software and database. If the patient enters the dangerous zone, the alarm will be triggered and the staffs will receive the alert through the network of host computers.

2.9 Software Based Application

The software based applications contain user interface (UI) and database used on the specifications of a particular project that uses RFID. RF readers are connected to computers that run the application to provide input data to be interpreted by the software. With the application of software, RFID makes a complete system

CHAPTER 3

METHODOLOGY

3.1 Research Methodology

3.1.1 Phase One (Project Initialization)

Phase One is the early stage of the project development. Thorough planning and understanding of the project is essential for the project to be done smoothly. The understanding of the project is including the research on the technology that will be used which is RFID. The data and information of the similar project also need to be done in order to have the clearer view of the project and to make sure the project is unique compared to the previous projects.

3.1.2 Phase Two (Project Designing)

Phase Two is where the designing the layout of the process of the system will be done. The Graphical User Interface (GUI) of the system is developed in this phase as well as the database development. The author also needs to determine the suitable hardware for the project. The hardware needed for the project is RFID Active Tag and RFID Reader

3.1.3 Phase Three (Prototyping)

Phase Three is where the hardware and the software integration begin. The integration of the system will be focusing on the interfacing between the devices, hardware and system of the project. After several testing have been done, the author need to improve the GUI of the software in order to make the system user-friendly.

3.2 Project Flow Chart

The rough idea on how the project flow and development phase is shown in the flow chart below.

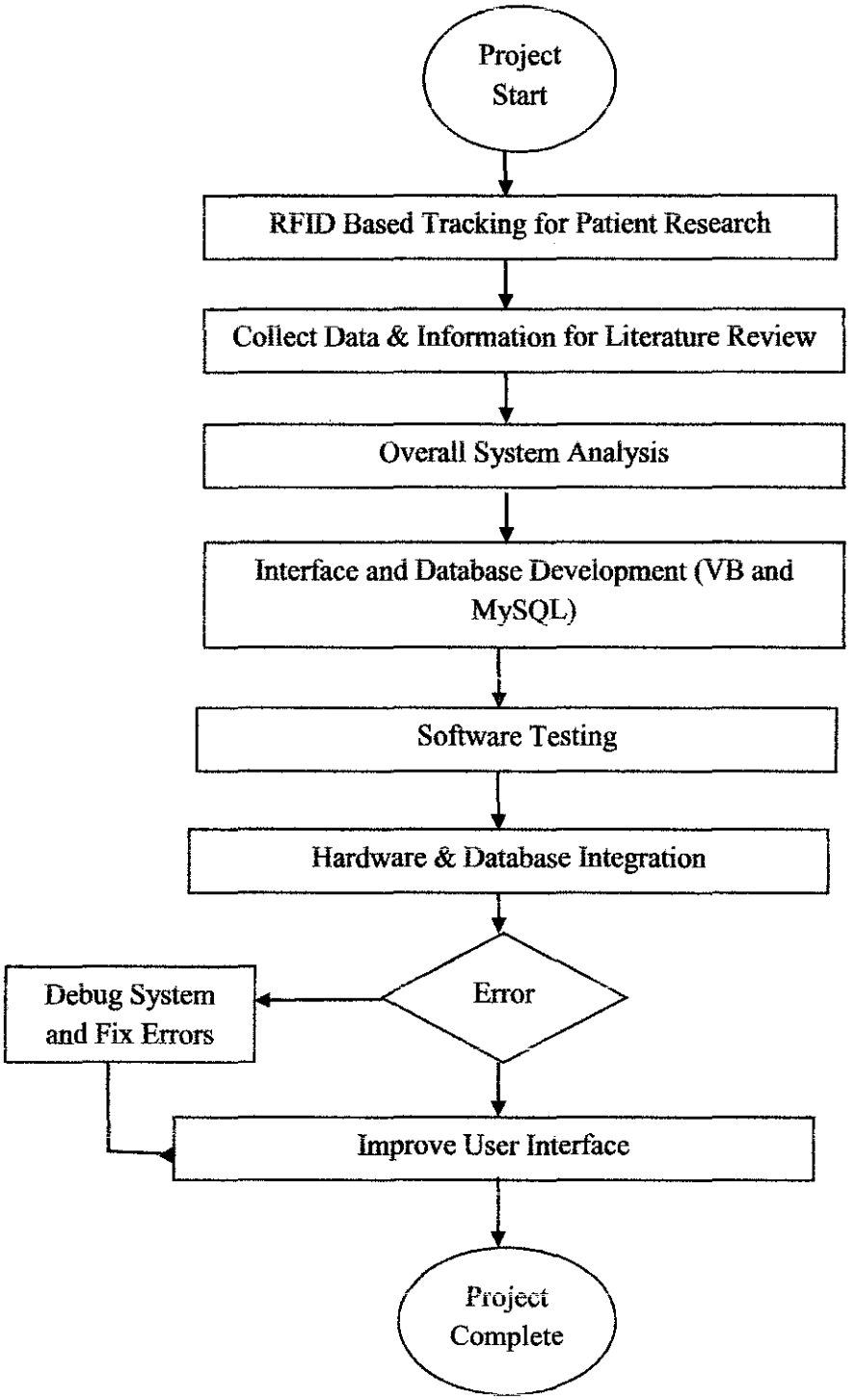


Figure 11 : Project Flow Chart

3.3 Tools Required

For the development of the RFID system, the tools required are Microsoft Visual Studio 2010, MySQL program and hardware of RFID. MySQL is the perfect platform to develop the database because MySQL is open source which means it is free, robust and fast. It is also easy to install on pretty much any operating system. Meanwhile, Microsoft Visual 2010 will provide user-friendly environment for the user to use the system.

3.3.1 MySQL

MySQL is a relational database management system (RDBMS that runs as a server providing multi-user access to a number of databases. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

3.3.2 Microsoft Visual Studio 2010

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio 2010 IDE has been redesigned which, according to Microsoft, clears the UI organization and "reduces clutter and complexity". The new IDE better supports multiple document windows and floating tool windows, while presenting better multi-monitor support. The IDE shell has been rewritten using the Windows Presentation Foundation (WPF), whereas the internals have been redesigned using Managed Extensibility Framework (MEF) that offers more extensibility points than previous versions of the IDE that enabled add-ins to modify the behavior of the IDE.

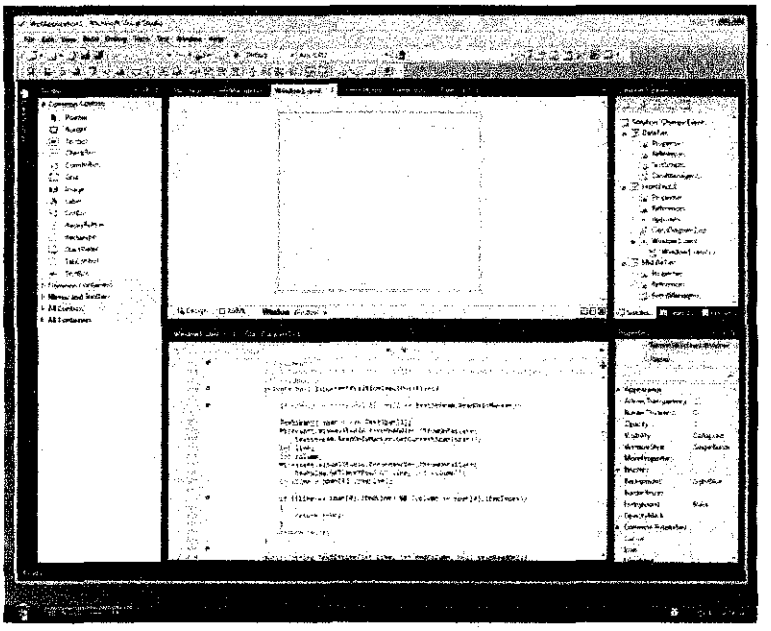


Figure 12: Microsoft Visual Studio 2010

3.3.3 RFID Hardware and Specification

For RFID hardware the two most important things are RFID active tag Sensor and RFID reader.

The active RFID tag can store much bigger data and information. It also can broadcast response signal to the reader and it is typically more reliable as there are only fewer errors compared to passive tags. The active tag sensors used for this project has characteristics as below;

i. Long communication range
ii. Having capability of initiating communications (communicate with the reader each 30 seconds for the accuracy of patient's location)
iii. Capability to perform independent monitoring
iv. The tag can stand a harsh development

Readers communicate to the host computer via an RJ-45 cable or via an Ethernet network connection. Readers are used to read the tags and transmit the received data to

the host computer. Reader has an antenna that emits radio waves while the tag responds by sending back its data. For Active Wave RFID reader, it will use two frequencies that are 916.5 MHz and 433 MHz for transmitting and receiving signal. This is to prevent clashing of transmitting and receiving data from the tags. RFID reader for this project will be in beacon mode which makes the reader let the tag to continuously send the data to them in certain period of times and place it into the database.

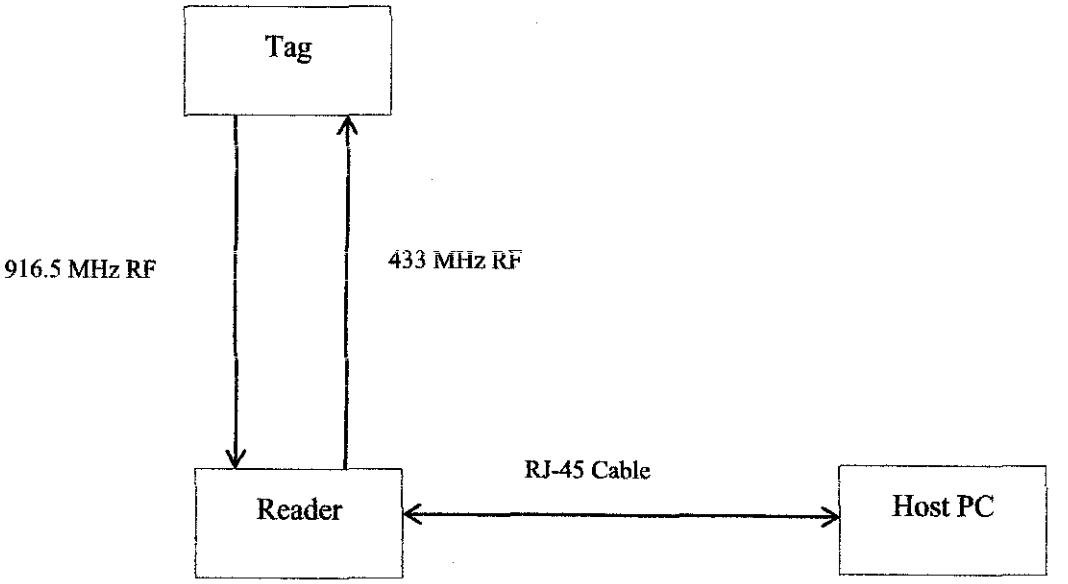


Figure 13 : System Flow for Active Wave Hardware Used

RFID Kit Components:

i. One Active Wave Reader and Power Supply
ii. One Active Wave RJ-45 Cable
iii. 6 Active Wave Wristbands
iv. One programming Station Software Application

Figure 14: Active Wave RFID Kit

3.4 Project Duration

In order to effectively monitor the progress of this project, the Gantt Chart consists of Part 2 of this project has been constructed. Please see the Appendix A.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Result and Discussion

As described in project methodology, the system has been developed and the Graphical User Interface (GUI) has been built. Basically, what will be discussed in this chapter is the patient tracking system in the hospital, how it works and how to use it. The author has developed the design of the user interface and the database of the system

4.1.1 System Overview

In order for the system to work perfectly, the right sequence of process need to be followed.

1. When the patient has been admitted to the hospital, he will be registered and right away tagged.
2. He will send to the respective ward. The patient will be monitored by a network of readers and linked to the control room.
3. If the patient wandering around and entered the dangerous zone, the control room will be alerted and the alarm will be triggered.
4. The medical staff along with the security will take the action immediately.

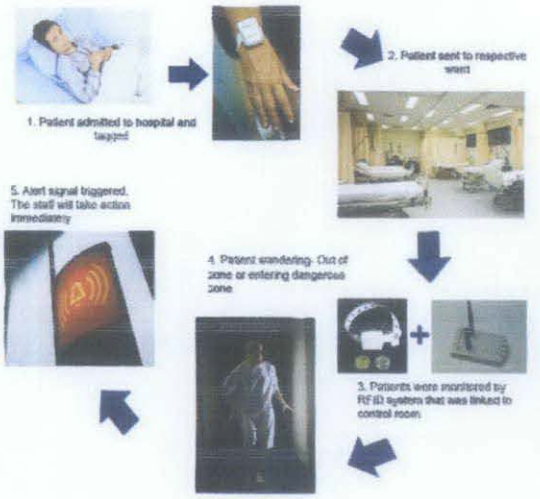


Figure 15: The System's Flow

4.1.2 Graphical User Interface (GUI)

Figure 16: Patient Tracking System Login

The operator of the Patient Tracking System will need to go through the login interface first before they can able to monitor and get access to the software (refer to Figure 16). After the operator fill in the staff ID and password, the main system window will appear.

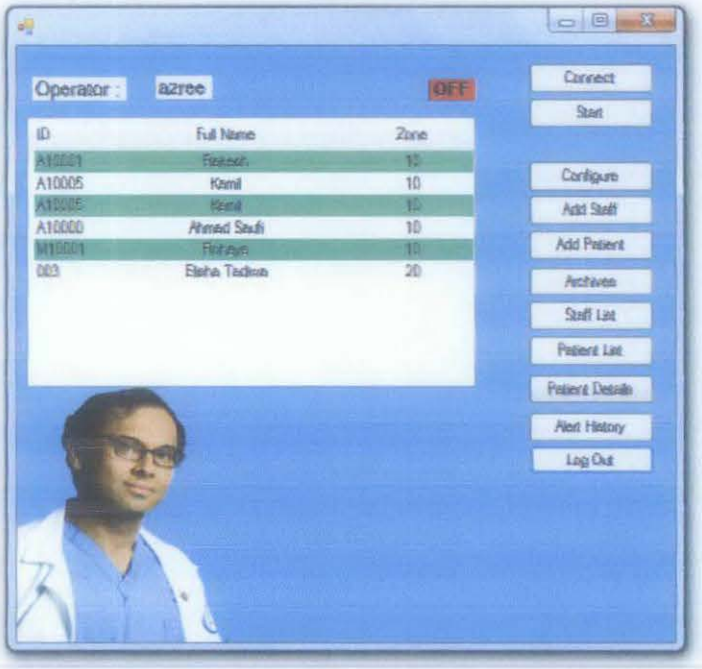


Figure 17: Main Menu of the System

On the main system window, the operator is able to access to the system and monitor the patients (refer to Figure 17). The window will help the operator to see the current position of all the patients inside the hospital. Moreover, the operator can also know the total number of patients. The operator need to turn ON the RFID reader first by clicking the connect button.

The operator can see the detail of the patients by clicking the 'Patient Detail Button'. After he clicked it, the new window will pop up and the operator needs to insert the patient's Tag ID and the detail of the patient will appear (refer to Figure 18).

Figure 18 : Patient Detail

When there is situation where the hospital management would like to add tracking coverage area and connected with more RFID reader, they can configure it by clicking the 'Configure' button. The window will pop up and the operator can add a new and remove IP address on particular readers (refer to Figure 19).

Figure 19 : Reader IP Configuration

The operator of the system could be able to add patients and staff profile by clicking the 'Add Staff' and 'Add Patient' button (refer to Figure 20 and 21 respectively). After he fills in the details, he must click 'Confirm' button and the data will be stored in MySQL database.

Figure 20 : Add Staff Profile

Figure 21: Add Patient

The information that has been created will be stored at the database of the systems as shown on Figure 22 and Figure 23.

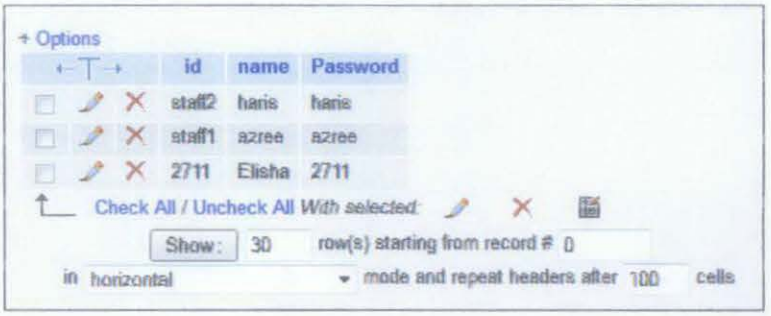


Figure 22: Database for the Staff Registration

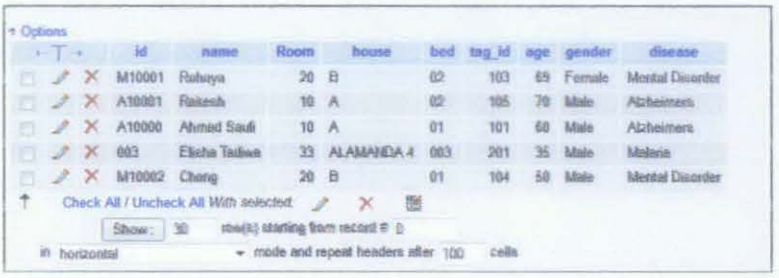


Figure 23: Database for the Patient Registration

If there is a conflict happen for example patients went into the dangerous zone, an alert signal will pop up and states the name of particular person. The alarm sound also will be triggered in order to make sure the operator is alert with the situation.

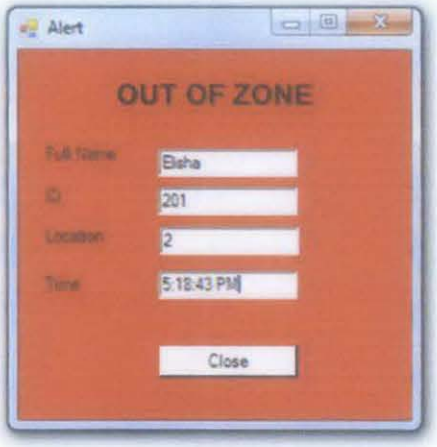


Figure 24 : Alert Alarm Triggered for Entering Dangerous Zone

All the conflict that has happened can be viewed by clicking the 'Alert History' button as refer to Figure 25.

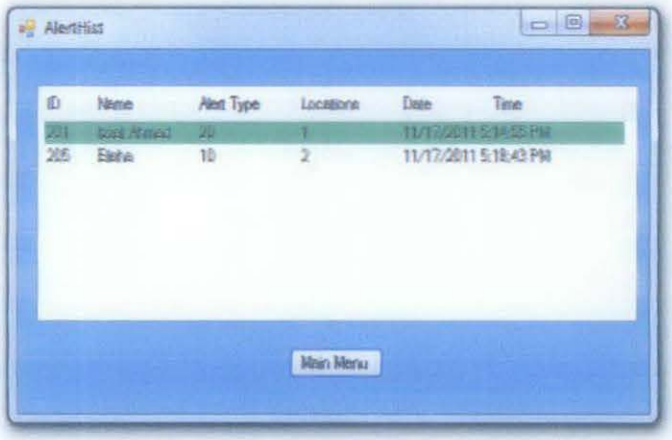


Figure 25 : Records of Alert History

The operator can view the list of all the registered patients by clicking the 'Patient List' button (refer to Figure 26). He can also view the list of all registered staff by clicking the 'Staff List' button (refer to Figure 27).

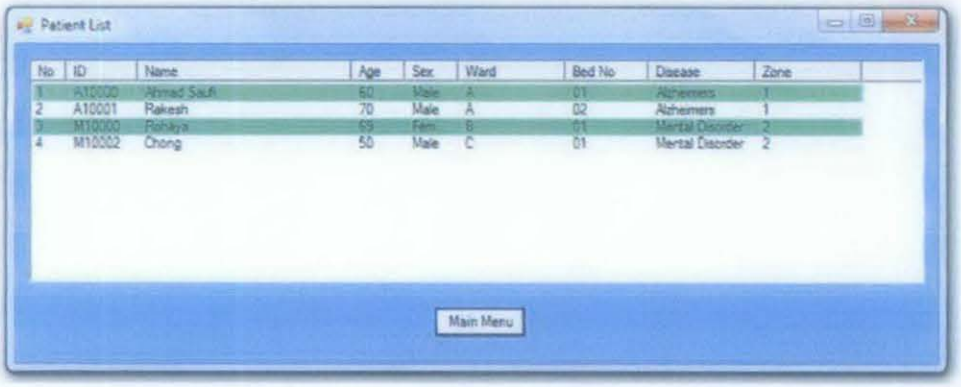


Figure 26 : List of All Registered Patients

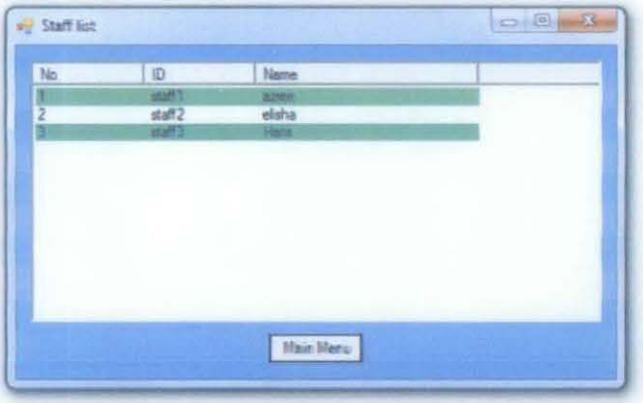


Figure 27 : List of All Registered Staff

The operator can delete the list of registered patients and it will automatically be stored in Archive database. The operator can view the archive of previous patients who has been registered by clicking the 'Archive' button (refer to Figure 28).

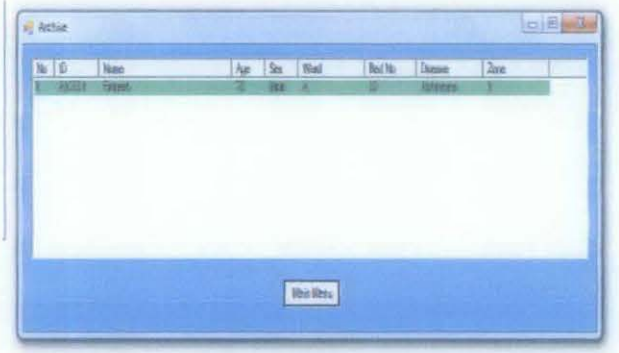


Figure 28 : Patient Archive

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As a conclusion, tracking patients in hospital using RFID system is developed with the objective to able to track the movements of the patients especially the Alzheimer's and mental disorder patients. The operator of the system would be able to monitor the location of the patients and would be alerted if the patient has entered the dangerous zone. This system will help the medical staff in hospital to track them and ease their work load of monitoring them every minute. It is an innovative system for tracking the patients and will be practical to be implemented inside the hospital facilities in Malaysia.

5.2 Recommendation

There are a few recommendations that can be done to improve the system better. The development of the system database needs to be optimized. The user interface also needs to be modified in order to create user friendly software. After the software has been fully developed, the integration with the RFID hardware need to done and several testing need to be performed. The testing will be done in selected hospital to prove that this system is applicable for the implementation in hospital.

The system can be also implemented to sense the temperature of the patients. This will help the medical staff to monitor the body temperature of patients remotely. Other than that, the RFID sensor can also detect the necessary information such as blood pressure.

REFERENCES

[1] Harvey Lehpamer. 2008. RFID Design Principles, Artech House

[2] RFID Journal Staff Writer. "RFID: Frequently Asked Questions",2005 . RFID Journal.

[3] Harold G. Clampitt, M., PE .2007. "RFID Certification Textbook"

[4] Frank Thornton, B.H., Anand M. Das, Hersh Bhargava, Anita Campbell, John Kleinschmidt. 2006. "RFID Security"

[5] Steven Shepard, Radio Frequency Identification, USA : McGraw-Hill

[6] Stephen A. Weis, RFID (Radio Frequency Identification): Principles and Applications, MIT CSAIL

[7] Visonic Technologies, http://visonictech.com

[8] David Bartholomew, David Duffy, Nigel Figgins. 2009. Strategies to Reduce Missing Patients: A Practical Workbook.

[9] P. Fuhrer and D. Guinard. Building a smart hospital using RFID technologies. In Proceedings of the 1st International Workshop on eHealth (ECEH06), volume P-91 of LNI, pages 131–142, October 2006.

[10] J. Reiner and M. Sullivan. 2005, *RFID in Healthcare – A panacea for the regulations and issues affecting the industry?* UPS Supply Chain Solutions.

[11] "Pharmaceutical Product Tampering News Media Factsheet," HDMA, April 2004.

[12] Koh R., E.M. Schuster, I Chackrabarti, and A. Beliman, "Securing the Pharmaceutical Supply Chain," Auto-ID Center, June 1, 2003

Final Year Project 1 Gantt Chart

Details/Week	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Title Selection/Proposal														
Journal/Conference paper studies														
Literature Review														
Extended Proposal														
Develop Interface and Database														
Proposal Defense														
Preparation of Draft Report														
Preparation of Interim Report														

Final Year Project 2 Gantt Chart

Details/Week	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
Writing Progress Report														
Integration of the Hardware and Software														
Pre-EDX														
Preparation of Draft Report														
Preparation of Final Report														
Viva														

i. Login

```
Imports System.IO
Imports MySql.Data.MySqlClient
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET

Public Class login

    Public ofName As String

    Private Function checkPswd(ByVal name As String, ByVal pswd As String) As
Boolean

        Dim found As Boolean = False
        Dim rows As Integer = 0
        Dim myConnString As String = "Database=inmate_db;Data
Source=localhost;User Id=root;Password="
        Dim cn As New MySqlConnection(myConnString)
        Dim cmdGH As New MySqlCommand("SELECT * FROM officer WHERE id = '" & name
& "'AND password = '" & pswd & "'", cn)
        Dim daGH As New MySqlDataAdapter
        Dim dsGH As New DataSet
        Dim dtGH As New DataTable
        Dim offName As String = ""

        cn.Open()

        Try
            daGH.SelectCommand = cmdGH
            daGH.Fill(dsGH, "employees")
            dtGH = dsGH.Tables("employees")
            rows = dtGH.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
        End Try

        If ConnectionState.Open Then
            cn.Close()
        End If

        If (rows > 0) Then
            found = True
            Dim namCmd As New MySqlCommand("SELECT name FROM officer WHERE id = '"
& name & "'", cn)
            Dim rdr As MySqlDataReader
            cn.Open()
            rdr = namCmd.ExecuteReader
            While rdr.Read
```

```
            offName = rdr.Item("name").ToString
        End While
        rdr.Close()
        cn.Close()
        ofName = offName
    Else
        found = False
    End If

    Return found

End Function

    Private Sub btnLogin_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdlogin.Click
    If (checkPswd(Me.txtsid.Text, Me.txtpswd.Text) = True) Then
        main.Show()
        Me.Hide()
    Else
        MessageBox.Show("Incorrect User name or Password")
        txtsid.Text = ""
        txtpswd.Text = ""
    End If
End Sub



    Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdexit.Click
    Me.Close()
End Sub


End Class
```

ii. Main Menu

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
Imports MySql.Data.MySqlClient

Public Class main

    Dim Hconn As IntPtr
    Dim readerIP(20) As Byte
    Dim readerPort As UInt16
    Dim commPort As UInt32
    Dim commBaud As UInt32
    Dim myPKTID As Integer
    Dim registered As Boolean
```

```
    Dim strhttp As String

    Dim ActiveWaveAPI As AW_API_NET.APINetClass = New AW_API_NET.APINetClass
    Dim ReaderEventHandler As AW_API_NET.fReaderEvent
    Dim TagEventHandler As AW_API_NET.fTagEvent
    Dim ipIdx As Integer = 0
    Dim pubReaderID As Integer
    Dim c As Integer
    Dim txtSender As String

    Dim tagID As ListViewItem
    Private Strt As System.Threading.Thread
    Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="
    Dim selID As String


    Private Sub main_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        lblOpName.Text = login.ofName

        readerPort = Convert.ToUInt16(10001)
        commPort = Convert.ToUInt32(1)
        commBaud = Convert.ToUInt32(115200)

        ReaderEventHandler = New AW_API_NET.fReaderEvent(AddressOf
Me.OnReaderEvent)  'AddressOf Me.OnReaderEvent
        TagEventHandler = New AW_API_NET.fTagEvent(AddressOf Me.OnTagEvent)

        Call readConfig()

        registered = False

        Strt = New System.Threading.Thread(AddressOf Threadi)
        Strt.Start()

    End Sub

    Private Function OnReaderEvent(ByVal readerEvent As
AW_API_NET.rfReaderEvent_t) As Integer

        Dim ipStr As String = ""

        'Beep()
        If
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SCAN_NETWORK
)) Then
            For i As Integer = 0 To readerEvent.ip.Length - 1
                ipStr += Convert.ToChar(readerEvent.ip(i))
            Next i
            IPListBox.Items.Add(ipStr)
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_OPEN_SOCKET)
) Then
            For i As Integer = 0 To readerEvent.ip.Length - 1
                ipStr += Convert.ToChar(readerEvent.ip(i))
            Next i
```

```
            'AddMsg("Socket Opened IP = " + ipStr)
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_CLOSE_SOCKET
)) Then
            For i As Integer = 0 To readerEvent.ip.Length - 1
                ipStr += Convert.ToChar(readerEvent.ip(i))
            Next i
            'AddMsg("Socket Closed IP = " + ipStr)

            Dim index As Integer
            If ipStr.Length > 0 Then
                index = IPListBox.FindStringExact(ipStr)
                If index >= 0 Then
                    IPListBox.Items.RemoveAt(index)
                End If
            End If
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_STD_FGEN_POW
ERUP)) Then
            'FGenIDTextBox.Text = readerEvent.fGenerator.ToString()
            'AddMsg("STD FGen Powered UP")
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_READER_POWER
UP)) Then
            'ReaderIDTextBox.Text = readerEvent.reader.ToString()
            'tText(readerEvent.reader.ToString())
            'AddMsg("Reader Powered UP")
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_QUERY_STD_FG
EN)) Then
            Dim str As String

            'AddMsg(readerEvent.eventType.ToString)
            'AddMsg(AW_API_NET.APIConsts.RF_READER_POWERUP.ToString)

            'AddMsg("STD FGEN Query _____")
            str = readerEvent.smartFgen.fsValue
            'AddMsg("FS Value = " + str)
            str = readerEvent.smartFgen.txTime
            'AddMsg("TX Time = " + str)
            str = readerEvent.smartFgen.waitTime
            'AddMsg("Wait Time = " + str)
            str = readerEvent.smartFgen.assignRdr
            'AddMsg("Assigned Rdr = " + str)
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_GET_RDR_FS))
Then
            'FSTextBox.Text = readerEvent.smartFgen.fsValue.ToString()
            'AddMsg("Reader ID:" + readerEvent.reader.ToString() + "  FS:" +
readerEvent.smartFgen.fsValue.ToString())
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SET_RDR_FS))
Then
            'AddMsg("Reader FS was set successfuly")
        ElseIf
readerEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_SCAN_IP))
Then
            ipStr = GetStringIP(readerEvent.ip)
```

```
            If ipStr.Length > 0 Then
                If IPListBox.FindStringExact(ipStr) = -1 Then
                    IPListBox.Items.Add(ipStr)
                End If
                'AddMsg("ScanIP IP=" + ipStr)
            End If
        End If

        ReportReaderEvent(readerEvent)

        Return 0
    End Function

    Private Sub ReportReaderEvent(ByRef readerEvent As AW_API_NET.rfReaderEvent_t)

        If readerEvent.errorStatus.ToString() = 0 Then
            commStatus.Text = "ON"
            pubReaderID = readerEvent.reader.ToString()
        End If

        ''AddMsg(msg)
    End Sub

    Private Function OnTagEvent(ByVal tagEvent As AW_API_NET.rfTagEvent_t) As
Integer

        Dim ipStr As String = ""

        'Beep()
        If
tagEvent.eventType.Equals(AW_API_NET.APIConsts.RF_CFG_TAG_REPORT_TAMPER) Then

            MessageBox.Show("ID = " & tagEvent.tag.id.ToString & " Tempered")
        ElseIf
tagEvent.eventType.Equals(Convert.ToUInt16(AW_API_NET.APIConsts.RF_TAG_READ)) Then
            Dim str As String
            Dim n As Integer
            n = CInt(Convert.ToInt16(tagEvent.tag.dataLen))
            For i As Integer = 0 To n - 1
                str = tagEvent.tag.data(i)
                ipStr += str + ""
            Next
        End If



        ReportTagEvent(tagEvent)

        Return 0
    End Function

    Private Sub ReportTagEvent(ByRef tagEvent As AW_API_NET.rfTagEvent_t)
        Dim tpr As String = ""
        If tagEvent.tag.id <> 0 Then

            tpr = tagEvent.tag.status.tamperSwitch.ToString
            InsertRow(tagEvent.tag.id.ToString(), tagEvent.errorStatus.ToString(),
tagEvent.reader.ToString(), tpr)
```

```
            'MessageBox.Show(tpr)
            Strt = New System.Threading.Thread(AddressOf Thread1)
            Strt.Start()

            'MessageBox.Show("Tag ID = " + tagEvent.tag.id.ToString + " " +
tagEvent.tag.status.tamperSwitch.ToString)
        End If
        ''AddMsg(msg)
    End Sub


    Private Sub InsertRow(ByVal tgid As Integer, ByVal st As Integer, ByVal rdr As
Integer, ByVal tpr As String)

        Dim name As String = ""
        Dim zone As Integer = 0
        Dim location As Integer = 0
        Dim rowsAlarm As Integer
        Dim dat As DateTime = Nothing
        Dim id As String = ""

        Dim cn As New MySqlConnection(strCn)
        ' If the connection string is null, use a default.
        'MsgBox("Start")
        'Beep()

        If st = 0 Then


            cn.Open()

            Dim cmdName As New MySqlCommand("SELECT * FROM inmate WHERE tag_id = "
& tgid, cn)
            Dim readerInmate As MySqlDataReader
            readerInmate = cmdName.ExecuteReader()
            While readerInmate.Read
                id = readerInmate.Item("id")
                name = readerInmate.Item("name")
                zone = readerInmate.Item("Room")
            End While
            readerInmate.Close()
            cn.Close()


            cn.Open()
            Dim cmdLoc As New MySqlCommand("SELECT RoomNo FROM readers WHERE
ReaderID=" & rdr, cn)
            Dim readerLoc As MySqlDataReader
            readerLoc = cmdLoc.ExecuteReader()
            While readerLoc.Read
                location = readerLoc.GetString(0)
            End While
            readerLoc.Close()
            cn.Close()


            Dim cmdAlarm As New MySqlCommand("SELECT * FROM alarm WHERE id = '" &
id & "'", cn)
```

```
            Dim daAlarm As New MySqlDataAdapter
            Dim dsAlarm As New DataSet
            Dim dtAlarm As New DataTable

            cn.Open()

            Try
                daAlarm.SelectCommand = cmdAlarm
                daAlarm.Fill(dsAlarm, "searchresult")
                dtAlarm = dsAlarm.Tables("searchresult")
                rowsAlarm = dtAlarm.Rows.Count()
            Catch ex As Exception
                MsgBox("Error: " & ex.Source & ": " & ex.Message,
MsgBoxStyle.OkOnly, "Connection Error !!")
            End Try

            If ConnectionState.Open Then
                cn.Close()
            End If



            If tpr = "False" Then

                If rowsAlarm = 0 Then

                    If location = zone Then
                        cn.Open()
                        Dim cmdIns1 As New MySqlCommand("INSERT INTO alarm (ack,
id, Name, Zone, ProgressTime) VALUES ('OK' " & ",'" & id & "','" & name & "'," &
location & ",'" & Format(DateTime.Now(), "yyyy-MM-dd hh:mm:ss") & "')", cn)
                        cmdIns1.ExecuteNonQuery()
                        cn.Close()
                    Else
                        cn.Open()
                        Dim cmdIns1 As New MySqlCommand("INSERT INTO alarm (ack,
id, Name, alertType, Zone, ProgressTime) VALUES ('Alert' " & ",'" & id & "','" &
name & "', 'OUT OF ZONE', " & location & ",'" & Format(DateTime.Now(), "yyyy-MM-dd
hh:mm:ss") & "')", cn)
                        cmdIns1.ExecuteNonQuery()
                        cn.Close()
                        MessageBox.Show("Alert!!!! (ID:" + tgid.ToString + ")" +
name + " moved from Zone" + zone.ToString + " To Zone" + location.ToString)
                    End If


                Else

                    If location = zone Then
                        cn.Open()
                        Dim cmdUpdate2 As New MySqlCommand("UPDATE alarm SET ack =
'OK', alertType = '', ProgressTime = '" & Format(DateTime.Now, "yyyy-MM-dd
hh:mm:ss") & "', Zone = " & location & " WHERE id = '" & id & "'", cn)
                        cmdUpdate2.ExecuteNonQuery()
                        If ConnectionState.Open Then
                            cn.Close()
                        End If
```

```
                    Else
                        cn.Open()
                        Dim cmdUpdate2 As New MySqlCommand("UPDATE alarm SET ack =
'Alert', alertType = 'OUT OF ZONE', ProgressTime = '" & Format(DateTime.Now,
"yyyy-MM-dd hh:mm:ss") & "', Zone = " & location & " WHERE id = '" & id & "'", cn)
                        cmdUpdate2.ExecuteNonQuery()
                        cn.Close()
                        MessageBox.Show("Alert!!!! (ID:" + tgid.ToString + ")" +
name + " moved from Zone" + zone.ToString + " To Zone" + location.ToString)

                    End If

                End If

            ElseIf tpr = "True" Then
                'alert.lblType.Text = "TEMPERED!!!!"
                'alert.txtTime.Text = Format(DateTime.Now, "hh:MM:ss dd-mm-ss")
                'alert.txtName.Text = name
                'alert.txtLocation.Text = location
                'alert.txtID.Text = id
                If rowsAlarm = 0 Then

                    cn.Open()
                    Dim cmdIns1 As New MySqlCommand("INSERT INTO alarm (ack, id,
Name, alertType, Zone, ProgressTime) VALUES ('ALERT' " & ",'" & id & "','" & name
& "', 'TEMPERED'," & location & ",'" & Format(DateTime.Now(), "yyyy-MM-dd
hh:mm:ss") & "')", cn)
                    cmdIns1.ExecuteNonQuery()
                    cn.Close()

                Else

                    cn.Open()
                    Dim cmdUpdate2 As New MySqlCommand("UPDATE alarm SET ack =
'ALERT', alertType = 'TAMPERED', ProgressTime = '" & Format(DateTime.Now, "yyyy-
MM-dd hh:mm:ss") & "', Zone = " & location & " WHERE id = '" & id & "'", cn)
                    cmdUpdate2.ExecuteNonQuery()
                    cn.Close()

                End If
                MessageBox.Show("Alert!!!! (Tag ID:" + tgid.ToString + ")" + name
+ " is Tempered")

            End If
        End If
        Strt = New System.Threading.Thread(AddressOf Thread1)
        Strt.Start()

    End Sub


    Public Function GetStringIP(ByVal ip As Byte()) As String

        Dim p As Integer
        Dim s As String
        Dim ct As Integer

        ct = 0
```

```
        p = 0
        s = ""
        While (Convert.ToBoolean((ct <= 3)) AndAlso Convert.ToBoolean((p < 20))
AndAlso Convert.ToBoolean((ip(p) <> 0)))
            If ip(p) <> 46 Then
                s += Convert.ToString(ip(p) - 48) '- 48
                p += 1
            Else
                ct += 1
                p += 1
                s += "."
            End If

        End While
        Return s
    End Function

    Private Sub readConfig()
        Try
            ' Create an instance of StreamReader to read from a file.
            Using sr As StreamReader = New StreamReader("config.txt")
                Dim line As String
                ' Read and display the lines from the file until the end
                ' of the file is reached.
                frmConfig.lbIp.Items.Clear()
                line = sr.ReadLine()
                If Mid(line, 1, 3) = "ip:" Then
                    frmConfig.lbIp.Items.Add(Mid(line, 4))
                End If
                If Mid(line, 1, 7) = "http://" Then
                    strhttp = line
                End If
                'lbIp.Items.Clear()
                While Not line Is Nothing
                    line = sr.ReadLine()
                    If Mid(line, 1, 3) = "ip:" Then
                        frmConfig.lbIp.Items.Add(Mid(line, 4))
                    End If
                    If Mid(line, 1, 7) = "http://" Then
                        strhttp = line
                    End If
                End While
                sr.Close()
            End Using
        Catch E As Exception
            ' Let the user know what went wrong.
            Console.WriteLine("The file could not be read:")
            Console.WriteLine(E.Message)
        End Try

    End Sub


    Private Sub BtnConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnConnect.Click
        scanIP()
        commStatus.BackColor = Color.DarkSeaGreen
        'activeReader()
```

```
        BtnStart.Enabled = True
    End Sub

    Private Sub scanIP()
        Dim iRet As Integer
        Dim ip(20) As Byte
        Dim stripx As String

        readIp()

        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        If registered = False Then

            ' Register reader callback handler
            ActiveWaveAPI.rfRegisterReaderEvent(ReaderEventHandler)
            'AddMsg("ReaderEvent registered")

            ' Register tag callback handler
            ActiveWaveAPI.rfRegisterTagEvent(TagEventHandler)
            'AddMsg("TagEvent registered")

            registered = True
        End If

        For c As Integer = 0 To frmConfig.lbIp.Items.Count - 1
            stripx = frmConfig.lbIp.Items.Item(c).ToString
            'MsgBox(stripx)

            For i As Integer = 0 To stripx.ToString.Length - 1
                ip(i) = Convert.ToByte(stripx.ToString.Chars(i))
            Next i
            iRet = ActiveWaveAPI.rfScanIP(ip, Convert.ToUInt16(myPKTID))

        Next c
        OpenConn()
    End Sub

    Private Sub OpenConn()
        Dim iRet As Integer
        Dim ip(20) As Byte
        Dim cIP(20) As Char
        Dim stripx As String = "192.168.10.26"

        myPKTID = myPKTID + 1
        'If AllIPRadioButton.Checked Then
        iRet = ActiveWaveAPI.rfOpenSocket(readerIP, readerPort, False,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS), Convert.ToUInt16(myPKTID))
        ' Else
        'Dim ipStr As String = IPTextBox.Text
        'cIP = ipStr.ToCharArray(0, ipStr.Length)

        'For i As Integer = 0 To IPTextBox.Text.Length - 1
        'ip(i) = Convert.ToByte(IPTextBox.Text.Chars(i))
```

```
        'Next i

        'iRet = ActiveWaveAPI.rfOpenSocket(ip, readerPort, False,
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_IP), Convert.ToUInt16(myPKTID))
        'End If
        'For i As Integer = 0 To stripx.ToString.Length - 1
        '    ip(i) = Convert.ToByte(stripx.ToString.Chars(i))
        'Next i

        'iRet = ActiveWaveAPI.rfOpenSocket(ip, readerPort, False,
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_IP), Convert.ToUInt16(myPKTID))

        'If (iRet = 0) Then
        'AddMsg("rfOpenSocket Successful. Return Code = " + iRet.ToString())
        'Else
        'AddMsg("rfOpenSocket Failed Return Code = " + iRet.ToString())
        'End If


    End Sub

    Private Sub readIp()
        Try
            ' Create an instance of StreamReader to read from a file.
            Using sr As StreamReader = New StreamReader("config.txt")
                Dim line As String
                ' Read and display the lines from the file until the end
                ' of the file is reached.
                'lbIp.Items.Clear()
                line = sr.ReadLine()
                If Mid(line, 1, 3) = "ip:" Then
                    frmConfig.lbIp.Items.Add(Mid(line, 4))
                End If
                While Not line Is Nothing
                    line = sr.ReadLine()
                    If Mid(line, 1, 3) = "ip:" Then
                        frmConfig.lbIp.Items.Add(Mid(line, 4))
                    End If
                End While
                sr.Close()
            End Using
        Catch E As Exception
            ' Let the user know what went wrong.
            Console.WriteLine("The file could not be read:")
            Console.WriteLine(E.Message)
        End Try
    End Sub

    Private Sub CloseConn()
        Dim iRet As Integer
        Dim ip(20) As Byte
        Dim cIP(20) As Char

        myPKTID = myPKTID + 1
        'If AllIPRadioButton.Checked Then
        iRet = ActiveWaveAPI.rfCloseSocket(readerIP,
Convert.ToUInt16(AW_API_NET.APIConsts.ALL_IPS))
        IPListBox.Items.Clear()
```

```
        'Else
        'Dim ipStr As String = IPTextBox.Text
        'cIP = ipStr.ToCharArray(0, ipStr.Length)

        'For i As Integer = 0 To IPTextBox.Text.Length - 1
        'ip(i) = Convert.ToByte(IPTextBox.Text.Chars(i))
        'Next i

        'iRet = ActiveWaveAPI.rfCloseSocket(ip,
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_IP))
        'End If

        If (iRet = 0) Then
            'AddMsg("rfCloseSocket Successful. Return Code = " + iRet.ToString())
        Else
            'AddMsg("rfCoseSocket Failed Return Code = " + iRet.ToString())
        End If

    End Sub


    Private Sub BtnStart_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnStart.Click
        ListView1.Items.Clear()
        activeReader()
        TmrCall.Enabled = True
    End Sub

    Private Sub activeReader()
        Dim iRet As Integer

        'If (STDFGenCheckBox.Checked = True) Then
        'MsgBox("Change baudrate to 115200")
        ' Return
        'End If

        myPKTID = myPKTID + 1
        'If (RdrCmdTypeComboBox.Text.Equals("ALL_READERS")) Then
        iRet = ActiveWaveAPI.rfResetReader(UInt16.Parse(1), UInt16.Parse(0),
UInt16.Parse(0), Convert.ToUInt16(AW_API_NET.APIConsts.ALL_READERS),
Convert.ToUInt16(myPKTID))
        'Else
        'If (ReaderIDTextBox.Text.Equals("")) Then
        ' MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error Msg")
        'Return
        'End If
        'iRet = ActiveWaveAPI.rfResetReader(UInt16.Parse(1),
Convert.ToUInt16(ReaderIDTextBox.Text), UInt16.Parse(0),
Convert.ToUInt16(AW_API_NET.APIConsts.SPECIFIC_READER), Convert.ToUInt16(myPKTID))
        'End If
        'AddMsg("ResetReader: " + iRet.ToString())
    End Sub



    Private Sub TmrCall_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TmrCall.Tick
        callTagID()
```

```
    End Sub

    Private Sub callTagID()
        Dim iRet As Integer
        'Dim limit As UInt32
        Dim tagSelect As AW_API_NET.rfTagSelect_t
        Dim tagList(50) As UInt32
        Dim rdrID As UInt16
        Dim longInterval As Boolean
        Dim RdrCmdType As Integer
        'tagList(0) = 22
        'tagList(1) = 23
        'If (STDFGenCheckBox.Checked = True) Then
        'MsgBox("Change baudrate to 115200")
        'Return
        'End If

        'If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
        'If (ReaderIDTextBox.Text.Equals("")) Then
        'MsgBox("No Reader ID", MsgBoxStyle.OkOnly, "Error Msg")
        'Return
        'Else
        'rdrID = Convert.ToUInt16(ReaderIDTextBox.Text)
        'End If
        'Else
        rdrID = UInt16.Parse(0)
        'End If

        'If (RdrCmdTypeComboBox.SelectedIndex = 0) Then
        'RdrCmdType = ActiveWaveAPI.SPECIFIC_READER
        'RdrCmdType = APIConsts.SPECIFIC_READER
        'Else
        'RdrCmdType = ActiveWaveAPI.ALL_READERS
        RdrCmdType = APIConsts.ALL_READERS
        'End If

        tagSelect.tagList = tagList

        'If (TagBox.Text.Equals("")) Then
        'If ((TagCmdTypeComboBox.SelectedIndex = 0) Or
(TagCmdTypeComboBox.SelectedIndex = 3)) Then
        'MsgBox("No Tag ID", MsgBoxStyle.OkOnly, "Error Msg")
        'Return
        'Else
        tagSelect.tagList(0) = UInt32.Parse(0)
        tagSelect.numTags = Convert.ToUInt32(50)
        'End If
        'Else
        'tagSelect.tagList(0) = Convert.ToUInt32(TagBox.Text)
        'tagSelect.numTags = Convert.ToUInt32(1)
        'End If

        'If (TagCmdTypeComboBox.SelectedIndex = 0) Then
        'tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_ID)
        'ElseIf (TagCmdTypeComboBox.SelectedIndex = 1) Then
        tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_FIELD)
```

```
        'ElseIf (TagCmdTypeComboBox.SelectedIndex = 2) Then
        'tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_TYPE)
        'Else
        'tagSelect.selectType =
Convert.ToUInt32(AW_API_NET.APIConsts.RF_SELECT_TAG_RANGE)
        'End If

        'If (ACCRadioButton.Checked) Then
        '    tagSelect.tagType = ActiveWaveAPI.ACCESS_TAG
        'ElseIf (ASTRadioButton.Checked) Then
        '    tagSelect.tagType = ActiveWaveAPI.ASSET_TAG
        'Else
        '    tagSelect.tagType = ActiveWaveAPI.INVENTORY_TAG
        'End If

        'If (ACCRadioButton.Checked) Then
        tagSelect.tagType = APIConsts.ACCESS_TAG
        ' ElseIf (ASTRadioButton.Checked) Then
        'tagSelect.tagType = APIConsts.ASSET_TAG
        ' Else
        'tagSelect.tagType = APIConsts.INVENTORY_TAG
        'End If
        If (myPKTID >= 223) Then
            myPKTID = 1
        Else
            myPKTID = myPKTID + 1
        End If

        'If (LongIntervalCheckBox.Checked) Then
        'longInterval = True
        'Else
        longInterval = False
        'End If

        iRet = ActiveWaveAPI.rfCallTags(UInt16.Parse(1), rdrID, UInt16.Parse(0),
UInt16.Parse(0), tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))
        'iRet = ActiveWaveAPI.rfQueryTags(UInt16.Parse(1), rdrID, UInt16.Parse(0),
tagSelect, True, longInterval, Convert.ToUInt16(RdrCmdType),
Convert.ToUInt16(myPKTID))

        'AddMsg("CallTags: " + iRet.ToString())


    End Sub

    Private Sub BtnAddOfficer_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnAddOfficer.Click
        addofficer.Show()
    End Sub

    Private Sub BtnAddInmate_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnAddInmate.Click
        addinm.Show()
    End Sub
```

```
    Private Sub BtnLogOut_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnLogOut.Click
        login.Show()
        Me.Hide()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0
            Dim str As String = ""

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM alarm", cn)
            Dim da As New MySqlDataAdapter(cmd)
            Dim ds As New DataSet()
            da.Fill(ds, "alarm")
            Dim myReader As MySqlDataReader

            ListView1.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If

            'Display raw in listview (attendlist)
            myReader = cmd.ExecuteReader()
            While myReader.Read
                'str = myReader.getString(0)
                No = No + 1
                tagID = ListView1.Items.Add(myReader.Item("id").ToString)
                'tagID = ListView1.Items.Add(Format(myReader.Item("tagID"),
"000"))
                tagID.SubItems.Add(myReader.Item("Name").ToString)
                tagID.SubItems.Add(myReader.Item("Zone").ToString)
                'tagID.SubItems.Add(myReader.getString(3))
                'tagID.SubItems.Add(myReader.Item("Zone").ToString)
            End While

            cmd.Connection.Close()

            'coloring backgroung
            While i <= ListView1.Items.Count - 1
                If i Mod 2 = 0 Then
                    ListView1.Items(i).BackColor = Color.Aquamarine
                Else
                    ListView1.Items(i).BackColor = Color.White
                End If
                i = i + 1
            End While

        End If

    End Sub
```

```
    Sub Thread1()
        list()
    End Sub


    Private Sub BtnConfig_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnConfig.Click
        frmConfig.Show()
    End Sub

    Private Sub ListView1_DoubleClick(ByVal sender As Object, ByVal e As
System.EventArgs) Handles ListView1.DoubleClick
        Dim I As Integer
        Dim ID As String = ""
        Dim name As String = ""
        Dim house As String = ""
        Dim bed As String = ""
        Dim tID As String = ""
        Dim EnrDate As String = ""

        For I = 0 To ListView1.SelectedItems.Count - 1
            'MsgBox(ListView1.SelectedItems(I).Text)
            ID = ListView1.SelectedItems(I).Text
        Next

        Dim cn As New MySqlConnection(strCn)
        Dim cmd As New MySqlCommand("SELECT * FROM inmate WHERE id = '" & ID &
"'", cn)
        Dim rdr As MySqlDataReader

        cn.Open()
        rdr = cmd.ExecuteReader
        While rdr.Read
            name = rdr.Item("name").ToString
            house = rdr.Item("house").ToString
            bed = rdr.Item("bed").ToString
            tID = rdr.Item("tag_id").ToString
            'EnrDate = Format(rdr.Item("enroll"), "yyyy-MM-dd")
        End While

        details.lblName.Text = name
        details.lblHouse.Text = house
        details.lblID.Text = ID
        details.lblBed.Text = bed
        details.lblTagID.Text = tID
        details.lblDate.Text = EnrDate

        Details.Show()
    End Sub

    Private Sub BtnAlert_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnAlert.Click
        AlertHist.Show()
    End Sub

    Private Sub BtnInmLst_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnInmLst.Click
```

```
        InmateList.Show()
    End Sub

    Private Sub BtnOpList_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnOpList.Click
        OfficerList.Show()
    End Sub


    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        PatientDetails.Show()
    End Sub


End Class
```

iii. Configure

```
Imports System
Imports System.IO

Public Class frmConfig

    ''Private Sub btnAdd_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
    ''    lbIp.Items.Add(txtIp.Text.ToString)
    ''End Sub

    ''Private Sub btnRemove_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnRemove.Click
    ''    If lbIp.SelectedItem.ToString <> "" Then
    ''        lbIp.Items.Remove(lbIp.Items(lbIp.SelectedIndex))
    ''    End If
    ''End Sub

    ''Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    ''    writeIp()
    ''    Me.Close()
    ''End Sub

    ''Private Sub writeIp()
    ''    ' Create an instance of StreamWriter to write text to a file.
    ''    Dim c As Integer
    ''    Using sw As StreamWriter = New StreamWriter("ips.txt")
    ''        ' Add some text to the file.
    ''        For c = 0 To lbIp.Items.Count - 1
    ''            sw.WriteLine(lbIp.Items.Item(c).ToString)
    ''        Next c
    ''        sw.Close()
    ''    End Using

    ''End Sub

    ''Private Sub readIp()
```

```
    Private Sub frmConfig_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        readIp()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Me.Close()
    End Sub

End Class
```

iv. Add Staff

```
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.Windows.Forms
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports System.IO
Imports MySql.Data.MySqlClient

Public Class addofficer

    Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="

    Private Sub addofficer_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load


    End Sub


    Private Sub btn_confirm_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_confirm.Click
        Dim id As String = TextBox1.Text.ToString
        Dim rowsCheck As Integer = 0
        Dim name As String = TextBox2.Text.ToString
        Dim pswd As String = TextBox3.Text.ToString

        Dim cn As New MySqlConnection(strCn)

        Dim cmdCheck As New MySqlCommand("SELECT * FROM officer WHERE id = '" & id
& "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable

        cn.Open()

        'Check from table (search)
        Try
            With daCheck
```

```
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
        End Try

        'If already connected, close connection
        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then
            'Open connection
            cn.Open()

            'Check for name and if no existing inmates available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO officer (id, name,
Password) VALUES ('" & id & "','" & name & "','" & pswd & "')", cn)
            cmdIns.ExecuteNonQuery()
            If ConnectionState.Open Then
                cn.Close()
            End If

            'If registration successful, show ID and name
            MsgBox(id + "/" + name + " has been registered successfully")

        Else
            'If already existing officer, show error
            MsgBox("Error!!! " + id + "is already registered in the database")

        End If

    End Sub


    Private Sub btn_Oclose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_Oclose.Click
        Me.Close()
    End Sub

End Class
```

v. Add Patient

```
Imports MySql.Data.MySqlClient
Public Class addinm

    Private Sub Submit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Submit.Click
```

```
        Dim id As String = TextBox1.Text.ToString
        Dim rowsCheck As Integer = 0
        Dim fname As String = TextBox2.Text.ToString
        Dim zones As String = TextBox7.Text.ToString
        Dim houses As String = TextBox3.Text.ToString
        Dim bedno As String = TextBox4.Text.ToString
        Dim tagid As String = TextBox5.Text.ToString
        Dim dateenrol As String = System.DateTime.Now


        Dim age As String = TextBox9.Text.ToString
        Dim gender As String = TextBox10.Text.ToString
        Dim disease As String = TextBox8.Text.ToString




        Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="


        'Connect to database
        Dim cn As New MySqlConnection(strCn)

        'Open database
        Dim cmdCheck As New MySqlCommand("SELECT * FROM inmate WHERE id = '" & id
& "'", cn)
        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable

        cn.Open()

        'Check from table (search)
        Try
            With daCheck
                .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
        End Try

        'If already connected, close connection
        If ConnectionState.Open Then
            cn.Close()
        End If


        If rowsCheck = 0 Then
            'Open connection
            cn.Open()
```

```
            'Check for name and if no existing inmates available, add them
            Dim cmdIns As New MySqlCommand("INSERT INTO inmate (id, name, Room,
house, bed, tag_id, age, gender, disease) VALUES ('" & id & "','" & fname & "','"
& zones & "','" & houses & "','" & bedno & "','" & tagid & "','" & age & "','" &
gender & "','" & disease & "')", cn)
            cmdIns.ExecuteNonQuery()

            If ConnectionState.Open Then
                cn.Close()
            End If

            'If registration successful, show ID and name
            MsgBox(id + "/" + fname + " has been registered successfully")

        Else
            'If already existing officer, show error
            MsgBox("Error!!! " + id + "is already registered in the database")

        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Me.Close()
    End Sub


End Class
```

vi. Staff List

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class OfficerList

    Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem

    Private Sub OfficerList_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        OffList.Columns.Clear()
        OffList.Columns.Add("No", 90, HorizontalAlignment.Left)
        OffList.Columns.Add("ID", 90, HorizontalAlignment.Left)
        OffList.Columns.Add("Name", 180, HorizontalAlignment.Left)
        Strt = New Thread(AddressOf Thread1)
        Strt.Start()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
```

```
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM officer", cn)
            Dim myReader As MySqlDataReader


            OffList.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If

            myReader = cmd.ExecuteReader()
            While myReader.Read
                No = No + 1
                ID = OffList.Items.Add(No.ToString)
                ID.SubItems.Add(myReader.Item("id").ToString)
                ID.SubItems.Add(myReader.Item("Name").ToString)
            End While

            cmd.Connection.Close()

            'coloring backgroung
            While i <= OffList.Items.Count - 1
                If i Mod 2 = 0 Then
                    OffList.Items(i).BackColor = Color.Aquamarine
                Else
                    OffList.Items(i).BackColor = Color.White
                End If
                i = i + 1
            End While

        End If

    End Sub

    Sub Thread1()
        list()
    End Sub

End Class
```

vii. Patient List

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class InmateList

    Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem
```

```
    Private Sub InmateList_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        InmList.Columns.Clear()
        InmList.Columns.Add("No", 30, HorizontalAlignment.Left)
        InmList.Columns.Add("ID", 90, HorizontalAlignment.Left)
        InmList.Columns.Add("Name", 180, HorizontalAlignment.Left)
        InmList.Columns.Add("Room", 45, HorizontalAlignment.Left)
        InmList.Columns.Add("House", 45, HorizontalAlignment.Left)
        InmList.Columns.Add("Bed", 45, HorizontalAlignment.Left)
        InmList.Columns.Add("Tag ID", 45, HorizontalAlignment.Left)
        'InmList.Columns.Add("Enrolled", 140, HorizontalAlignment.Left)
        Strt = New Thread(AddressOf Thread1)
        Strt.Start()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM inmate", cn)
            Dim myReader As MySqlDataReader


            InmList.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If

            myReader = cmd.ExecuteReader()
            While myReader.Read
                No = No + 1
                ID = InmList.Items.Add(No.ToString)
                ID.SubItems.Add(myReader.Item("id").ToString)
                ID.SubItems.Add(myReader.Item("name").ToString)
                ID.SubItems.Add(myReader.Item("Room").ToString)
                ID.SubItems.Add(myReader.Item("house").ToString)
                ID.SubItems.Add(myReader.Item("bed").ToString)
                ID.SubItems.Add(myReader.Item("tag_id").ToString)
                'ID.SubItems.Add(myReader.Item("enroll").ToString)
            End While

            cmd.Connection.Close()

            'coloring backgroung
            While i <= InmList.Items.Count - 1
                If i Mod 2 = 0 Then
                    InmList.Items(i).BackColor = Color.Aquamarine
                Else
                    InmList.Items(i).BackColor = Color.White
                End If
```

```
            i = i + 1
        End While

    End If

End Sub

Sub Thread1()
    list()
End Sub

Private Sub InmList_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles InmList.SelectedIndexChanged

End Sub
End Class
```

viii. Patient Details

```
Imports MySql.Data.MySqlClient
Imports System.Data
Imports System
Imports System.Runtime.InteropServices
Imports Microsoft.VisualBasic
Imports AW_API_NET
Imports System.IO
'Imports MySql.Data.MySqlClient
Imports System.Threading
Public Class PatientDetails



    Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="
    Dim temper As ListViewItem
    Dim tgID As String
    Private str2 As System.Threading.Thread
    Private str1 As System.Threading.Thread
    Private Sub Patient_Records_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs)

        ' System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = False

        '  lvwTemp.Columns.Clear()
        '  lvwTemp.Columns.Add("Temperature", 100, HorizontalAlignment.Left)
        '  lvwTemp.Columns.Add("Date                              Time", 225,
HorizontalAlignment.Right)

        ' lvwTemp.Hide()

    End Sub
    Private Sub btndelete_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)


    End Sub
```

```
    Private Sub listDetails()

        Dim uname As String = ""
        Dim sex As String = ""
        Dim bedno As String = ""
        Dim age As Integer = 0
        Dim disease As String = ""
        Dim zone As String = ""
        Dim ward As String = ""

        Dim tgID As String = txtTagID.Text

        If (checkID(tgID) = True) Then

            If Me.InvokeRequired Then
                Me.Invoke(New MethodInvoker(AddressOf listDetails))
            Else

                'Establish connection
                Dim temp As String = ""
                ' Dim dat As DateTime = Nothing
                Dim cn2 As New MySqlConnection(strCn)

                'Fill in Patient details
                Dim cmdDetails As New MySqlCommand("SELECT * FROM inmate WHERE
tag_id = '" & tgID & "'", cn2)
                Dim rdrDetails As MySqlDataReader

                cn2.Open()
                rdrDetails = cmdDetails.ExecuteReader
                While rdrDetails.Read
                    Name = rdrDetails.Item("name")
                    sex = rdrDetails.Item("gender")
                    bedno = rdrDetails.Item("bed")
                    disease = rdrDetails.Item("disease")
                    age = rdrDetails.Item("age")
                    zone = rdrDetails.Item("room")
                    ward = rdrDetails.Item("house")
                End While

                rdrDetails.Close()
                cn2.Close()

                txtfn.Text = Name
                txtage.Text = age.ToString
                txtsex.Text = sex
                txtward.Text = ward
                txtbedno.Text = bedno
                'txtdw.Text = dateofadmin
                txtdisease.Text = disease
                txtzone.Text = zone
                'txtdiagnosis.Text = diagn
                'txtdoa.Text = Format(admisD, "yyyy-MM-dd HH:mm:ss")
                'txtdob.Text = Format(birthD, "yyyy-MM-dd HH:mm:ss")
                'txticnum.Text = ic

                txtfn.Show()
                txtage.Show()
```

```
            txtsex.Show()
            txtward.Show()
            txtbedno.Show()
            txtdisease.Show()
            txtzone.Show()
            'txtdisease.Show()
            ' Label1.Show()
            Label2.Show()
            Label3.Show()
            Label4.Show()
            Label5.Show()
            Label6.Show()
            'Label7.Show()
            'Label8.Show()
            Label9.Show()
            Label10.Show()




            'mainmenu.Show()

        End If

    Else

        MessageBox.Show("There is no record corresponding to this ID")

        '          txtfn.Text = " "
        '         txtage.Text = ""
        '        'txtic.Text = ""
        '       txtsex.Text = ""
        '      txtward.Text = ""
        '     txtbedno.Text = ""
        '    txtdw.Text = ""
        '   txtdisease.Text = ""
        '  txtzone.Text = ""
        '
        '          lvwTemp.Items.Clear()
        'lvwTemp.Hide()
        '         btnconfirm.Hide()

    End If

End Sub
Sub Thread1()
    listDetails()
End Sub
Private Function checkID(ByVal id As String) As Boolean

    Dim found As Boolean = False
    Dim rows As Integer = 0
    Dim myConnString As String = "Database=inmate_db;Data
Source=localhost;User Id=root;Password="
    Dim cn As New MySqlConnection(myConnString)
    Dim cmdGH As New MySqlCommand("SELECT * FROM inmate WHERE tag_id = '" & id
& "'", cn)
```

```
        Dim daGH As New MySqlDataAdapter
        Dim dsGH As New DataSet
        Dim dtGH As New DataTable

        cn.Open()

        Try
            daGH.SelectCommand = cmdGH
            daGH.Fill(dsGH, "details")
            dtGH = dsGH.Tables("details")
            rows = dtGH.Rows.Count()
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
        End Try

        If ConnectionState.Open Then
            cn.Close()
        End If

        If (rows > 0) Then
            found = True
        Else
            found = False
        End If

        Return found

    End Function

    Private Sub btnconfirm_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnconfirm.Click
        str2 = New Thread(AddressOf Thread2)
        str2.Start()
    End Sub
    Private Sub listvw()

        '        Dim i As Integer = 0
        '
        '        tgID = txtTagID.Text
        '        Dim cn As New MySqlConnection(strCn)
        '        If Me.InvokeRequired Then
        ' Me.Invoke(New MethodInvoker(AddressOf listvw))
        ' Else

        'Populate the ListView
        '        Dim cmd As New MySqlCommand("SELECT * FROM temptrack WHERE tagID
= '" & tgID & "'", cn)
        '        Dim rdrTemp As MySqlDataReader

        'lvwTemp.Items.Clear()

        '        If cn.State = ConnectionState.Closed Then
        ' cn.Open()
        ' End If

        'Display raw in listview (attendlist)
        'rdrTemp = cmd.ExecuteReader()
```

```
        '        While rdrTemp.Read
        'temper = lvwTemp.Items.Add(rdrTemp.Item("temperature"))
        'temper.SubItems.Add(Format(rdrTemp.Item("time"), "yyyy-MM-dd
HH:mm:ss"))
        'End While

        ' rdrTemp.Close()
        ' cn.Close()

        'coloring backgroung
        ' While i <= lvwTemp.Items.Count - 1
        'If i Mod 2 = 0 Then
        'lvwTemp.Items(i).BackColor = Color.Aquamarine
        '  Else
        '  lvwTemp.Items(i).BackColor = Color.White
        '  End If
        ' i = i + 1
        '  End While

        'lvwTemp.Show()

        ' End If

    End Sub
    Private Sub btnSearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSearch.Click
        str1 = New Thread(AddressOf Thread1)
        str1.Start()
    End Sub
    Sub Thread2()
        listvw()
    End Sub



    Private Sub mmenu_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mmenu.Click
        Me.Hide()
        main.Show()

    End Sub



    Private Sub btndel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndel.Click
        MsgBox("You are about to delete this Patient")
        btndel.Visible = False
        btndelete.Visible = True

    End Sub

    Private Sub btndelete_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btndelete.Click
        ' Dim name As String = txtName.Text.ToString
        Dim rowsCheck As Integer = 0
        Dim ID As String = txtTagID.Text.ToString
        ' Dim Post As String = cboPost.Text.ToString
```

```
        ' Dim user_name As String = txtUname.Text.ToString
        ' Dim Password As String = txtpswd.Text.ToString
        'Dim strCn As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="
        'Dim strCn2 As String = "Database=hospital_db;Data Source=localhost;User
Id=root;Password="


        'Connect to database
        Dim cn1 As New MySqlConnection(strCn)
        'Dim cn2 As New MySqlConnection(strCn)
        'Open database
        'Dim cmdCheck As New MySqlCommand("SELECT * FROM temptrack WHERE tagID =
'" & ID & "'", cn1)
        Dim cmdCheck2 As New MySqlCommand("SELECT * FROM inmate WHERE tag_id = '"
& ID & "'", cn1)

        Dim daCheck As New MySqlDataAdapter
        Dim dsCheck As New DataSet
        Dim dtCheck As New DataTable

        cn1.Open()

        'Check from table (search)
        Try
            With daCheck
                ' .SelectCommand = cmdCheck
                .Fill(dsCheck, "searchresult")
            End With
            dtCheck = dsCheck.Tables("searchresult")
            rowsCheck = dtCheck.Rows.Count()

            'If unable to connect, show error!
        Catch ex As Exception
            MsgBox("Error: " & ex.Source & ": " & ex.Message, MsgBoxStyle.OkOnly,
"Connection Error !!")
        End Try

        'If already connected, close connection
        If CBool(ConnectionState.Open) Then
            cn1.Close()
            'cn2.Close()
        End If


        If rowsCheck <> 0 Then
            'Open connection
            cn1.Open()
            'cn2.Open()

            'Check for name and if no existing inmates available, add them
            ' Dim cmdIns As New MySqlCommand("DELETE FROM temptrack WHERE(tagID =
'" & ID & "' )", cn1)
            Dim cmdIns2 As New MySqlCommand("DELETE FROM patientdetails
WHERE(tagid = '" & ID & "' )", cn1)

            'Dim myInsertQuery As String = "Delete From YourTable Where ( field1 =
'" & field1 & "' );"
```

```
            ' cmdIns.ExecuteNonQuery()
            cmdIns2.ExecuteNonQuery()

            If CBool(ConnectionState.Open) Then
                cn1.Close()
                'cn2.Close()
            End If

            'If registration successful, show ID and name
            MsgBox(ID + " has been Deleted Succesfully")

        Else
            'If already existing officer, show error
            MsgBox("ERROR!!!  " + ID + "  does not exist or has already been
deleted")

        End If
        txtfn.Text = ""
        txtage.Text = ""
        'txtic.Text = ""
        txtsex.Text = ""
        txtward.Text = ""
        txtbedno.Text = ""
        txtdw.Text = ""
        txtdisease.Text = ""
        txtzone.Text = ""

        lvwTemp.Items.Clear()
        'lvwTemp.Hide()
        btnconfirm.Hide()

    End Sub


End Class
```

ix. Alert History

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class AlertHist

    Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="
    Dim strCn2 As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem

    Private Sub AlertHist_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        AlertList.Columns.Clear()
        AlertList.Columns.Add("ID", 90, HorizontalAlignment.Left)
        AlertList.Columns.Add("Name", 180, HorizontalAlignment.Left)
```

```
        AlertList.Columns.Add("Alert Type", 180, HorizontalAlignment.Left)
        AlertList.Columns.Add("Locations", 180, HorizontalAlignment.Left)
        AlertList.Columns.Add("Date        Time", 180, HorizontalAlignment.Left)
        Strt = New Thread(AddressOf Thread1)
        Strt.Start()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0
            Dim str As String = ""

            Dim cn As New MySqlConnection(strCn)
            Dim cn2 As MySqlConnection
            Dim cmd As New MySqlCommand("SELECT * FROM alarm WHERE ack = 'Alert'",
cn)
            Dim da As New MySqlDataAdapter(cmd)
            Dim ds As New DataSet()
            da.Fill(ds, "alarm")
            Dim myReader As MySqlDataReader
            Dim myReader2 As MySqlDataReader
            Dim cmd2 As MySqlCommand
            Dim zone As String = ""

            AlertList.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If

            'Display raw in listview (attendlist)
            myReader = cmd.ExecuteReader()
            While myReader.Read
                If (myReader.Item("alertType").ToString = "OutOfZone") Then
                    cn2 = New MySqlConnection(strCn2)
                    cmd2 = New MySqlCommand("SELECT * FROM inmate WHERE id = '" &
myReader.Item("id") & "'", cn2)
                    cn2.Open()
                    myReader2 = cmd2.ExecuteReader
                    zone = myReader2.Item("Room")
                    myReader2.Close()
                    cn2.Close()
                    ID = AlertList.Items.Add(myReader.Item("id").ToString)
                    ID.SubItems.Add(myReader.Item("Name").ToString)
                    ID.SubItems.Add(myReader.Item("alertType").ToString)
                    ID.SubItems.Add(zone + "/" + myReader.Item("Zone").ToString)
                    ID.SubItems.Add(myReader.Item("ProgressTime").ToString)
                Else 'If (myReader.Item("alertType").ToString = "Tamper") Then
                    ID = AlertList.Items.Add(myReader.Item("id").ToString)
                    ID.SubItems.Add(myReader.Item("Name").ToString)
                    ID.SubItems.Add(myReader.Item("alertType").ToString)
                    ID.SubItems.Add(myReader.Item("Zone").ToString)
                    ID.SubItems.Add(myReader.Item("ProgressTime").ToString)
```

```
            End If
        End While

        cmd.Connection.Close()

        'coloring backgroung
        While i <= AlertList.Items.Count - 1
            If i Mod 2 = 0 Then
                AlertList.Items(i).BackColor = Color.Aquamarine
            Else
                AlertList.Items(i).BackColor = Color.White
            End If
            i = i + 1
        End While

    End If

End Sub

Sub Thread1()
    list()
End Sub



End Class
```

x. Alert

```
Public Class alert

    Private Sub alert_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Dim UserQueryResult As DataTable = officer.DoSQL("SELECT * FROM `inmate`
WHERE `tag_id` = '" & Me.Tag & "' LIMIT 0,1")

        'Dim buffer(3) As String
        'buffer(1) = UserQueryResult.Rows(0)(1)
        'buffer(2) = UserQueryResult.Rows(0)(0)
        'buffer(3) = UserQueryResult.Rows(0)(2)

        'lbl_nm.Text = buffer(1)
        'lbl_d.Text = buffer(2)
        'lbl_loca.Text = buffer(3)
        'lbl_tm.Text = Now.ToShortTimeString
    End Sub

    Private Sub btn_Submit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_Submit.Click
        Me.Close()
    End Sub
End Class
```

xi. Archive

```
Imports MySql.Data.MySqlClient
Imports System.Threading

Public Class archive

    Dim strCn As String = "Database=inmate_db;Data Source=localhost;User
Id=root;Password="
    Private Strt As System.Threading.Thread
    Dim ID As ListViewItem

    Private Sub InmateList_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        arclist.Columns.Clear()
        arclist.Columns.Add("No", 30, HorizontalAlignment.Left)
        arclist.Columns.Add("ID", 45, HorizontalAlignment.Left)
        arclist.Columns.Add("Name", 150, HorizontalAlignment.Left)
        arclist.Columns.Add("Zone", 90, HorizontalAlignment.Left)
        arclist.Columns.Add("Ward", 90, HorizontalAlignment.Left)
        arclist.Columns.Add("Bed", 90, HorizontalAlignment.Left)
        arclist.Columns.Add("Tag ID", 90, HorizontalAlignment.Left)

        arclist.Columns.Add("Age", 45, HorizontalAlignment.Left)
        arclist.Columns.Add("Gender", 90, HorizontalAlignment.Left)
        arclist.Columns.Add("Disease", 90, HorizontalAlignment.Left)

        'inmList.Columns.Add("Enrolled", 140, HorizontalAlignment.Left)
        Strt = New Thread(AddressOf Thread1)
        Strt.Start()
    End Sub

    Private Sub list()

        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf list))
        Else
            'Establish connection
            Dim i As Integer = 0
            Dim No As Integer = 0

            Dim cn As New MySqlConnection(strCn)
            Dim cmd As New MySqlCommand("SELECT * FROM archive", cn)
            Dim myReader As MySqlDataReader


            arclist.Items.Clear()

            If cn.State = ConnectionState.Closed Then
                cn.Open()
            End If

            myReader = cmd.ExecuteReader()
            While myReader.Read
                No = No + 1
                ID = arclist.Items.Add(No.ToString)
                'ID.SubItems.Add(myReader.Item("count").ToString)
```

```
                ID.SubItems.Add(myReader.Item("id").ToString)
                ID.SubItems.Add(myReader.Item("name").ToString)
                ID.SubItems.Add(myReader.Item("Room").ToString)
                ID.SubItems.Add(myReader.Item("house").ToString)
                ID.SubItems.Add(myReader.Item("bed").ToString)
                ID.SubItems.Add(myReader.Item("tag_id").ToString)

                ID.SubItems.Add(myReader.Item("age").ToString)
                ID.SubItems.Add(myReader.Item("gender").ToString)
                ID.SubItems.Add(myReader.Item("disease").ToString)
                'ID.SubItems.Add(myReader.Item("enroll").ToString)
            End While

            cmd.Connection.Close()

            'coloring background
            While i <= arclist.Items.Count - 1
                If i Mod 2 = 0 Then
                    arclist.Items(i).BackColor = Color.Aquamarine
                Else
                    arclist.Items(i).BackColor = Color.White
                End If
                i = i + 1
            End While

        End If

    End Sub

    Sub Thread1()
        list()
    End Sub



    Private Sub arclist_SelectedIndexChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles arclist.SelectedIndexChanged

    End Sub
End Class
```