Impact Factor (SJIF): 5.301

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 5, Issue 4, April-2018

SMART INNOVATIVE HOSPITAL TOKEN SYSTEM FOR GENERAL PATIENT

R.Praveen Kumar -M.Tech.,

Assistant Professor,Dept of Information Technology, Panimalar Institute of Technology

V.Ajay Krishna D.Vignesh R.Vishal

Student, Dept of Information Technology, Panimalar Institute of Technology

Abstract:

Hospitals have always used queues at their premises; this makes visiting hospitals a very painful experience for the patients. In today's world, patients are customers who expect good service. It is important to ensure that the wait-time is less and the waiting experience is better. Hospitals are seeking solutions which can meet these expectations. For this expectation here we are proposing a project. However with the development of technology there is an integrated system that can simplify the process of managing hospital data, which called hospital information system. Hospital information system is a wide integrated electronic system, which is designed to regulate administrative, financial, and various aspects of the clinics in hospitals and healthcare facilities. With the use of information systems in a hospital, it's expected to provide faster service and minimize the waiting time for the patients. Due to this system we can avoid the patient who is all no need to wait for a long time, and also we can easily find the person in emergency case. For that we can minimize the patient pain and stress.

Keywords:

Embedded System, Zigbee, emergency cases, patient's heartbeat and temperature is monitored, Notification.

1.Introduction:

People don't like to be in queues when they are in serious health conditions. They need to be provided with good service and quick response from the hospitals. This project helps to connect emergency cases as soon as possible with the doctors. This system monitors the patient's temperature and heart beat. If the heart beat and temperature of the body is less then the particular scalar, a notification is sent to the hospital information system which will make the hospital to take instant action on that patient. This will reduce risk of patient's health condition.

Related work:

In earlier days, Managing visitors requires distributing tokens and then calling visitors, guiding visitors to one of the many counters in single person. Human called the priority token hospital for admission control, scheduling, and policing in integrated-services networks, where arrival processes and performance objectives vary greatly from to another. Already we known in some hospitals in counter section they allow familiar visitors to allow to consult with doctor. Because the other patients is depressions to cause. To overcome the problems in over projects.

Proposed work:

In our project we proposed the token systems which are reliable to be used at the places where patient have to wait in line for their turn. These systems allow patients to wait without having to stand in line. So this system used to doctor known as patient's emergency condition. So the serious condition patients avoid in the queue. Here we are monitoring the patients' health conditions such as heartbeat, pressure, temperature etc. These records are transmitted via wireless sensor networks to the doctor section. So it's very helpful to collecting patient data more effective and efficiently This proposed systems is considered very helpful which is they can reduce the time when processing the patient consulting. If all the various clinical disciplines use this system in a hospital, the patient outcomes can be easily shared among the different clinical disciplines.

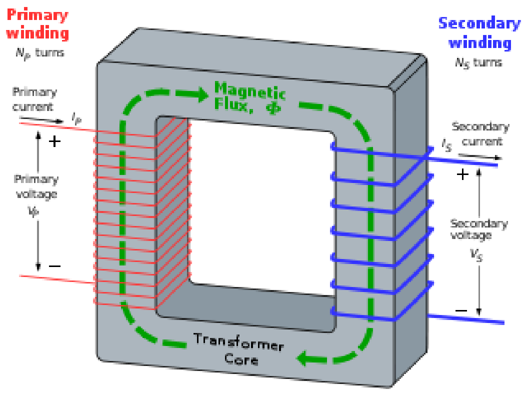
2.Components Required:

HARDWARE:

Power Supply

Transformer

The potential transformer will step down the power supply voltage (0-230V) to (0-6V) level. Then the secondary of the potential transformer will be connected to the precision rectifier, which is constructed with the help of op–amp. The advantages of using precision rectifier are it will give peak voltage output as DC, rest of the circuits will give only RMS output.



Bridge rectifier:

When four diodes are connected as shown in figure, the circuit is called as bridge rectifier. The input to the circuit is applied to the diagonally opposite corners of the network, and the output is taken from the remaining two corners.
Let us assume that the transformer is working properly and there is a positive potential, at point A and a negative potential at point B. the positive potential at point A will forward bias D3 and reverse bias D4.
The negative potential at point B will forward bias D1 and reverse D2. At this time D3 and D1 are forward biased and will allow current flow to pass through them; D4 and D2 are reverse biased and will block current flow.
The path for current flow is from point B through D1, up through RL, through D3, through the secondary of the transformer back to point B. this path is indicated by the solid arrows. Waveforms (1) and (2) can be observed across D1 and D3.
One-half cycle later the polarity across the secondary of the transformer reverse, forward biasing D2 and D4 and reverse biasing D1 and D3.Current flow will now be from point A through D4, up through RL, through D2, through the secondary of T1, and back to point A. This path is indicated by the broken arrows. Waveforms (3) and (4) can be observed across D2 and D4. The current flow through RL is always in the same direction. In flowing through RL this current develops a voltage corresponding to that shown waveform (5). Since current flows through the load (RL) during both half cycles of the applied voltage, this bridge rectifier is a full-wave rectifier.

One advantage of a bridge rectifier over a conventional full-wave rectifier is that with a given transformer the bridge rectifier produces a voltage output that is nearly twice that of the conventional full-wave circuit.

IC voltage regulators:

Voltage regulators comprise a class of widely used ICs. Regulator IC units contain the circuitry for reference source, comparator amplifier, control device, and overload protection all in a single IC. IC units provide regulation of either a fixed positive voltage, a fixed negative voltage, or an adjustably set voltage. The regulators can be selected for operation with load currents from hundreds of milli amperes to tens of amperes, corresponding to power ratings from milli watts to tens of watts.

A fixed three-terminal voltage regulator has an unregulated dc input voltage, Vi, applied to one input terminal, a regulated dc output voltage, Vo, from a second terminal, with the third terminal connected to ground.

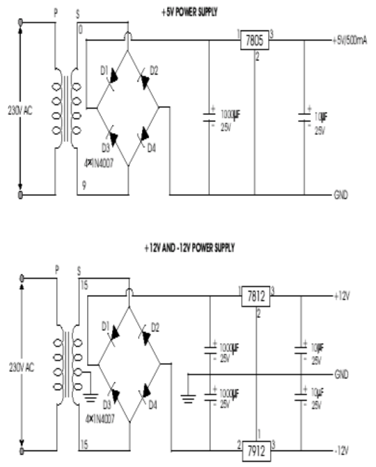
The series 78 regulators provide fixed positive regulated voltages from 5 to 24 volts. Similarly, the series 79 regulators provide fixed negative regulated voltages from 5 to 24 volts.

PIC MICROCONTROLLER

The PIC microcontroller PIC16f877a is one of the most renowned microcontrollers in the industry. This controller is very convenient to use, the coding or programming of this controller is also easier. One of the main advantages is that it can be write-erase as many times as possible because it use FLASH memory technology. It has a total number of 40 pins and there are 33 pins for input and output. PIC16F877A is used in many pic microcontroller projects. PIC16F877A also have many application in digital electronics circuits.

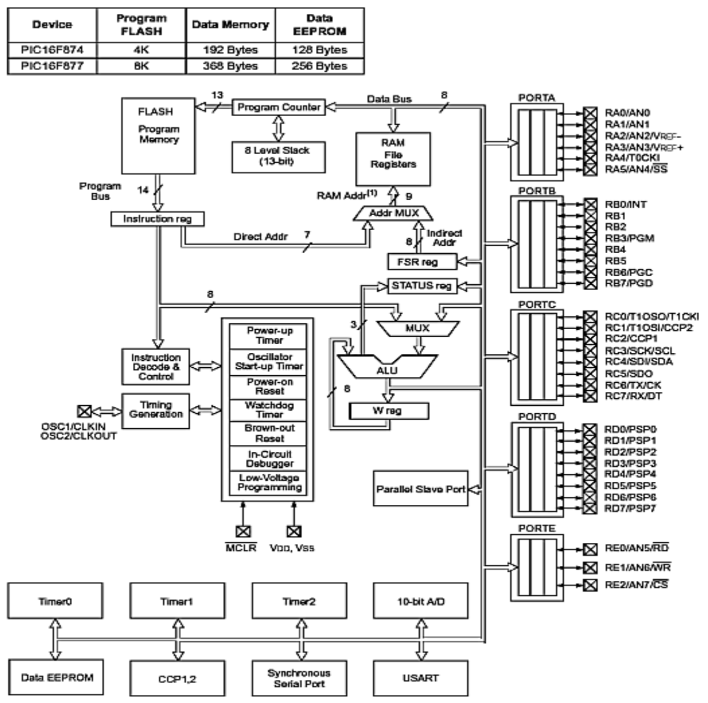
PIC16F877A microcontroller

PIC16f877a finds its applications in a huge number of devices. It is used in remote sensors, security and safety devices, home automation and in many industrial instruments. An EEPROM is also featured in it which makes it possible to store some of the information permanently like transmitter codes and receiver frequencies and some



other related data. The cost of this controller is low and its handling is also easy. Its flexible and can be used in areas where microcontrollers have never been used before as in coprocessor applications and timer functions etc.

Architecture Diagram:



Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes

Memory Organization of PIC16F877

The memory of a PIC 16F877 chip is divided into 3 sections. They are

1. Program memory
2. Data memory and
3. Data EEPROM

Program memory

Program memory contains the programs that are written by the user. The program counter (PC) executes these stored commands one by one. Usually PIC16F877 devices have a 13 bit wide program counter that is capable of addressing 8K×14 bit program memory space. This memory is primarily used for storing the programs that are written (burned) to be used by the PIC. These devices also have 8K*14 bits of flash memory that can be electrically erasable /reprogrammed. Each time we write a new program to the controller, we must delete the old one at that time. The figure below shows the program memory map and stack.

PIC16f877 Program Memory

The PIC16F87XA family has an 8-level deep x 13-bit wide hardware stack. The stack space is not a part of either program or data space and the stack pointers are not readable or writable. In the PIC microcontrollers, this is a special block of RAM memory used only for this purpose.

Each time the main program execution starts at address 0000 – Reset Vector. The address 0004 is “reserved” for the “interrupt service routine” (ISR).

PIC16F87XA Data Memory Organization

The data memory of PIC16F877 is separated into multiple banks which contain the general purpose registers (GPR) and special function registers (SPR). According to the type of the microcontroller, these banks may vary. The PIC16F877 chip only has four banks (BANK 0, BANK 1, BANK 2, and BANK4). Each bank holds 128 bytes of addressable memory.

Data EEPROM and FLASH

The data EEPROM and Flash program memory is readable and writable during normal operation (over the full VDD range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers. There are six SFRs used to read and write this memory:

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

The EEPROM data memory allows single-byte read and writes. The Flash program memory allows single-word reads and four-word block writes. Program memory write operations automatically perform an erase-before write on blocks of four words. A byte write in data EEPROM memory automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device for byte or word operations.

SPECIAL FUNCTION REGISTERS (SFR)

The special function registers are also memory registers which is used for special dedicated functions. These registers perform various dedicated functions inside the PIC chip. Each special function inside this PIC chip is controlled by using these registers. These registers are used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are normally implemented as in the form of static RAM memory. A list of these registers is given in the tables below. The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. The figures below shows SFR memory map of PIC16F877.

3.TEMPERATURE SENSOR

The first slave connected to a temperature sensor LM35. This senses the temperature of an engine and provides the level of temperature.

General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling.

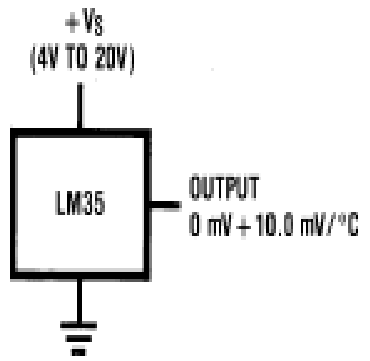


Figure 3 Basic Centigrade Temperature Sensor (+2°C to +150°C)

The LM35 does not require any external calibration or trimming to provide typical accuracies of ±1⁄4°C at room temperature and ±3⁄4°C over a full −55 to +150°C temperature range. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a −55° to +150°C temperature range, while the LM35C is rated for a −40° to +110°C range (−10° with improved accuracy).
The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

Capacitive Load

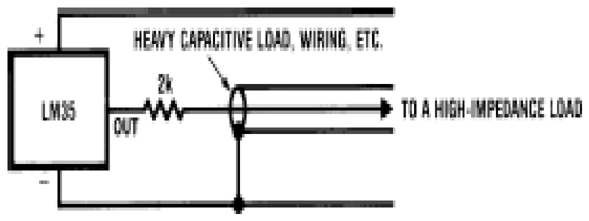
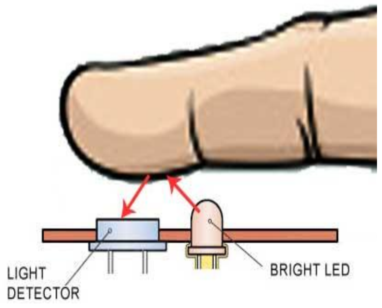


Figure 3.1 LM35 With Decoupling From Capacitive Load

Like most micro power circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pf without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor.

4.HEART BEAT SENSOR

HEART BEAT sensor is designed to give digital output of heat beat when a finger is placed on it. When the HEART BEAT detector is working, the beat LED flashes in unison with each heart beat. This digital output can be connected to microcontroller directly to measure the Beats Per Minute (BPM) rate. It works on the principle of light modulation by blood flow through finger at each pulse.
Medical heart sensors are capable of monitoring vascular tissue through the tip of the finger or the ear lobe. It is often used for health purposes, especially when monitoring the body after physical training.



HEART BEAT is sensed by using a high intensity type LED and LDR. The finger is placed between the LED and LDR. As Sensor a photo diode or a photo transistor can be used. The skin may be illuminated with visible (red) using transmitted or reflected light for detection. The very small changes in reflectivity or in transmittance caused by the varying blood content of human tissue are almost invisible. Various noise sources may produce disturbance signals with amplitudes equal or even higher than the amplitude of the pulse signal. Valid pulse measurement therefore requires extensive preprocessing of the raw signal.
The new signal processing approach presented here combines analog and digital signal processing in a way that both parts can be kept simple but in combination are very effective in suppressing disturbance signals.
The setup described here uses a red LED for transmitted light illumination and a LDR as detector. With only slight changes in the preamplifier circuit the same hardware and software could be used with other illumination and detection concepts. The detectors photo current (AC Part) is converted to voltage and amplified by an operational amplifier (LM358).

5.ZIGBEE

Introduction

ZigBee is a technological standard designed for control and sensor networks. Based on the IEEE 802.15.4 Standard Created by the ZigBee Alliance . It Operates in Personal Area Networks (PAN's) and device-to-device networks Connectivity between small packet devices Control of lights, switches, thermostats, appliances, etc. Developement started 1998, when many enginereers realized that WiFi and Bluetooth were going to be unsuitable for many applications. IEEE 802.15.4 standard was completed in May 2003. Organization defining global standards for reliable, cost- effective, low power wireless applications. A consortium of end users and solution providers, primarily responsible for the development of the 802.15.4 standard. Developing applications and network capability utilizing the 802.15.4 packet delivery mechanism.

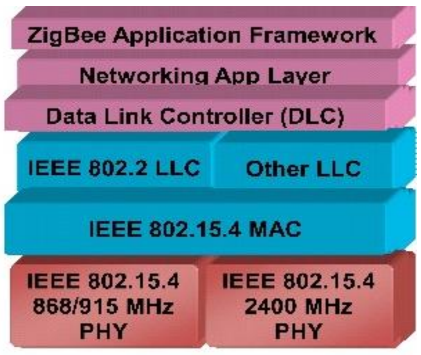

ZigBee Application Framework
Networking App Layer
Data Link Controller (DLC)
IEEE 802.2 LLC
Other LLC
IEEE 802.15.4 MAC
IEEE 802.15.4 868/915 MHz PHY
IEEE 802.15.4 2400 MHz PHY

Figure 5 IEEE 802.15.4 Architecture

SOFTWARE:

EMBEDDED C

High-level language programming has long been in use for embedded-systems development. However, assembly programming still prevails, particularly for digital-signal processor (DSP) based systems. DSPs are often programmed in assembly language by programmers who know the processor architecture inside out. The key motivation for this practice is performance, despite the disadvantages of assembly programming when compared to high-level language programming.

If the video decoding takes 80 percent of the CPU-cycle budget instead of 90 percent, for instance, there are twice as many cycles available for audio processing. This coupling of performance to end-user features is characteristic of many of the real-time applications in which DSP processors are applied. DSPs have a highly specialized architecture to achieve the performance requirements for signal processing applications within the limits of cost and power consumption set for consumer applications. Unlike a conventional Load-Store (RISC) architecture, DSPs have a data path with memory-access units that directly feed into the arithmetic units. Address registers are taken out of the general-purpose register file and placed next to the memory units in a separate register file.

For example, is required in many algorithms and is supplied as a primitive in many DSPs. However, there is no such primitive in Standard C. To express saturated arithmetic in C requires comparisons, conditional statements, and correcting assignments. Instead of using a primitive, the operation is spread over a number of statements that are difficult to recognize as a single primitive by a compiler.

DESCRIPTION

Embedded C is designed to bridge the performance mismatch between Standard C and the embedded hardware and application architecture. It extends the C language with the primitives that are needed by signal-processing applications and that are commonly provided by DSP processors. The design of the support for fixed-point data types and named address spaces in Embedded C is based on DSP-C. DSP-C [1] is an industry-designed extension of C with which experience was gained since 1998 by various DSP manufacturers in their compilers. For the development of DSP-C by ACE (the company three of us work for), cooperation was sought with embedded-application designers and DSP manufacturers.

MULTIPLE ADDRESS SPACES

Embedded C supports the multiple address spaces found in most embedded systems. It provides a formal mechanism for C applications to directly access (or map onto) those individual processor instructions that are designed for optimal memory access. Named address spaces use a single, simple approach to grouping memory locations into functional groups to support MAC buffers in DSP applications, physical separate memory spaces, direct access to processor registers, and user-defined address spaces.

The Embedded C extension supports defining both the natural multiple address space built into a processor's architecture and the application-specific address space that can help define the solution to a problem.

Embedded C uses address space qualifiers to identify specific memory spaces in variable declarations. There are no predefined keywords for this, as the actual memory segmentation is left to the implementation. As an example, assume that **X** and **Y** are memory qualifiers. The definition:

```
X int a[25] ;
```

Means that **a** is an array of 25 integers, which is located in the **X** memory. Similarly (but less common):

```
X int * Y p ;
```

Means that the pointer **p** is stored in the **Y** memory. This pointer points to integer data that is located in the **X** memory. If no memory qualifiers are used, the data is stored into unqualified memory.

For proper integration with the C language, a memory structure is specified, where the unqualified memory encompasses all other memories. All unqualified pointers are pointers into this unqualified memory. The unqualified memory abstraction is needed to keep the compatibility of the **void *** type, the **NULL** pointer, and to avoid duplication of all library code that accesses memory through pointers that are passed as parameters.

I/O HARDWARE ADDRESSING

The motivation to include primitives for I/O hardware addressing in Embedded C is to improve the portability of device-driver code. In principle, a hardware device driver should only be concerned with the device itself. The driver operates on the device through device registers, which are device specific. However, the method to access these registers can be very different on different systems, even though it is the same device that is connected. The I/O hardware access primitives aim to create a layer that abstracts the system-specific access method from the device that is accessed. The ultimate goal is to allow source-code portability of device drivers between different systems. In the design of the I/O hardware-addressing interface, three requirements needed to be fulfilled:

The device-drive source code must be portable.
The interface must not prevent implementations from producing machine code that is as efficient as other methods. The design should permit encapsulation of the system-dependent access method. The design is based on a small collection of functions that are specified in the <iohw.h> include file. These interfaces are divided into two groups; one group provides access to the device, and the second group maintains the access method abstraction itself.
To access the device, the following functions are defined
by Embedded C:

```
unsigned int iord(  ioreg_designator );
void iowr(  ioreg_designator, unsigned int value );
void ioor(  ioreg_designator, unsigned int value );
void ioand( ioreg_designator, unsigned int value );
void ioxor( ioreg_designator, unsigned int value );
```

These interfaces provide read/write access to device registers, as well as typical methods for setting/resetting individual bits. Variants of these functions are defined (with **buf** appended to the names) to access arrays of registers. Variants are also defined (with l appended) to operate with **long** values.
All of these interfaces take an I/O register designator **ioreg_designator** as one of the arguments. These register designators are an abstraction of the real registers provided by the system implementation and hide the access method from the driver source code. Three functions are defined for managing the I/O register designators. Although these are abstract entities for the device driver, the driver does have the obligation to initialize and release the access methods. These functions do not access or initialize the device itself because that is the task of the driver. They allow, for example, the operating system to provide a memory mapping of the device in the user address space.

```
void iogroup_acquire( iogrp_designator );
void iogroup_release( iogrp_designator );
void iogroup_map( iogrp_designator, iogrp_designator );
```

The **iogrp_designator** specifies a logical group of I/O register designators; typically this will be all the registers of one device. Like the I/O register designator, the I/O group designator is an identifier or macro that is provided by the system implementation. The map variant allows cloning of an access method when one device driver is to be used to access multiple identical devices.

EMBEDDED C PORTABILITY

By design, a number of properties in Embedded C are left implementation defined. This implies that the portability of Embedded C programs is not always guaranteed. Embedded C provides access to the performance features of DSPs. As not all processors are equal, not all Embedded C implementations can be equal.

For example, suppose an application requires 24-bit fixed-point arithmetic and an Embedded C implementation provides only 16 bits because that is the native size of the processor. When the algorithm is expressed in Embedded C, it will not produce outputs of the right precision.

Writing C code with the low-level processor-specific support may at first appear to have many of the portability problems usually associated with assembly code. In the limited experience with porting applications that use Embedded C extensions, an automotive engine controller application (about 8000 lines of source) was ported from the eTPU, a 24-bit special-purpose processor, to a general-purpose 8-bit Freescale 68S08 with about a screen full of definitions put into a single header file. The porting process was much easier than expected. For example, variables that had been implemented on the processor registers were ported to unqualified memory in the general-purpose microprocessor by changing the definitions in the header definition and without any actual code modifications. The exercise was to identify the porting issues and it is clear that the performance of the special-purpose processor is significantly higher than the general-purpose target.

References:

[1] D. K. Sharma and R. C. Goyal, Hospital Administration and Human Resource Management. Delhi, India: PHI, 2013.

[2] J. Pinchin, "Getting lost in hospitals costs the NHS and patients," The Guardian

http://www.theguardian.com/healthcarenetwork/2015/mar/05/lost-hospitals-costs nhspatients- navigation, 2015.

[3] N. Richards and A. Coulter, Is the NHS Becoming more Patient-Centred? Oxford, UK: Picker Institute Europe, 2007.

[4] F. Ueckert, M. Goerz, M. Ataian, S. Tessmann, and H.-U. Prokosch, "Empowerment of patients and communication with health care professionals through an electronic health record," International Journal of Medical Informatics, vol. 70, pp. 99-108,2003.

[5] J. L. van den Brink, P. W. Moorman, M. F. de Boer, J. F. A. Pruyn, C. D. A. Verwoerd, and J. H. v. Bemmel, "Involving the patient: A prospective study on use,appreciation and effectiveness of an information system in head and neck cancer care " International Journal of Medical Informatics, vol. 74, pp. 839-849, 2005.

[6] L. Zhang and X. Xu, "A Community Public Health System Design based on HL7 Criterions," Computer and Information Science, vol. 4, pp. 148-151, 2011.

[7]H. S. Ensour, "The Impact of Management Information Systems (MIS) Technologies on the Quality of Services Provided at the University of Tabuk," International Journal of Network Security & Its Applications (IJNSA), vol. 6, no. 2, pp. 1-20, 2014.

[8]M. Armony, On Patient Flow In Hospitals: A Data-Based Queueing-Science Perspective, New York: Stochastic Systems, 2015.

[9]A. Jobori and A. S. A. Jumaily, "Automatic Queueing Model for Banking Applications," International Journal of Advanced Computer Science and Applications, vol. 2, p. 5, 2011.

[10] Y. Susanti, E. Widiyastuti, Y. Dewi, E. Reza and I. Hanifah, "Kinerja Sistem Antrian dan Simulasi Model Antrian pada Appointment Registration System di Instalasi Rawat Jalan Rumah Sakit Al-Islam Bandung," Bandung, 2015.

[11] J. P. Caulkins, "Might Randomization in Queue Discipline Be Useful When Waiting Cost is a Concave Function of Waiting Time?," Research Showcase @ CMU, 2007.

[12] C. E. Chinwuko, E. C. Daniel, O. P. Ugochukwu and O. O. J., "Analysis of a queueing system in an organization (a case study of First Bank PLC, Nigeria)," American Journal of Engineering Research, vol. 03, no. 02, pp. 63-72, 2014