

## **Hospital Management Software Development**

Case study: Rainbow Specialist Medical Center, Lagos Nigeria.

Olawale Ayotunde SobogunGod

Bachelor's Thesis

Business Information Technology

2012



Degree programme

<p><b>Author</b> Olawale SobogunGod</p>	<p><b>Group or year of entry</b> 2007</p>
<p><b>Title of thesis</b> The development of hospital management software</p>	<p><b>Number of pages and appendices</b> 21 + 2</p>
<p><b>Supervisor or supervisors</b> Sauli Isonikkilä</p>	
<p>The purpose of this thesis was to implement a hospital management software which is suitable for small private hospitals in Nigeria, especially for the ones that use a file based system for storing information rather than having it stored in a more efficient and safer environment like databases or excel programming software. The software developed within this thesis project was specifically designed for the Rainbow specialist hospital which is based in Lagos, the commercial neurological center of Nigeria.</p> <p>The thesis firstly gave an overview of the different types of software development models that could be used for developing hospital management software. The models considered in this thesis were the Waterfall model, the iterative and incremental models, the V-shape model and the Spiral model. Therefore, the most relevant aspects of these models were presented.</p> <p>The system development part was then considered, explaining some modules and the human interface with which a user can interact with the software to store or get information from the database. The software system itself and appendices contain the screenshots of the human interface after the completion of the system.</p> <p>Microsoft visual studio 2010 and Microsoft SQL server management studio were used in developing the user interface and the database for the software.</p>	
<p><b>Keywords</b> Modules,MS SQL,Database,Connection string,UI, EDM</p>	

## Table of contents

1	Introduction.....	1
1.1	Background.....	1
1.2	Objective.....	1
1.3	Scope.....	2
1.4	Terms.....	2
1.5	System Requirement.....	2
1.6	Methodology.....	2
1.7	User Characteristics.....	2
2	Software Development Process Models.....	4
2.1	Water Fall Model.....	6
2.2	Iterative and Incremental Model.....	7
2.3	Spiral Model.....	9
2.4	The V-Model.....	12
3	User Interfaces Design.....	14
3.1	Modules and Interfaces.....	16
3.1.1	Authentication management.....	16
3.1.2	Patient Registration.....	17
3.1.3	Diagnosis.....	17
3.1.4	Treatment information.....	18
3.1.5	Patient Invoice.....	18
3.1.6	Bill.....	18
3.1.7	Appointment.....	19
3.1.8	Staff Information.....	19
3.1.9	Rooms.....	20
3.1.10	Medicine Maintenance.....	20
3.1.11	Supplier.....	20
3.1	Data Model.....	21
4	Other functionalities.....	24
4.1	Security.....	24

4.2 Performance Requirement .....	24
4.3 Error Handling .....	24
4.4 Installation .....	25
5 Conclusions .....	26
Bioblography .....	27
Appendices .....	26
Appendix 1 .....	26
Appendix 2 .....	26

## Dedication

*This thesis is dedicted firstly to God almighty, who gave me the grace and strength to walk this path, though extrememly tough for me but pulled me up when I was down,gave me hope when there was no hope and walked with me when I was alone.*

*Secondly to my mum,Late Miss Oluranti Christiana Fakolujo, who taught me that “tough times don’t last but tough people and to always remember the son of who I am”, though you have passed on, you continue to live in my heart.*

*And lastly,I will give a big thank you to miss ANIKA Alexandersson who made all effort to so be stand with during my period of trial. May God almighty in His infinity mercy answer you prayer and remain blesses in the name of our lord Jesus Christ*

# 1 Introduction

## 1.1 Background

The health sector in Nigeria is in a very poor state due to the fact that it is under funded by the government, therefore is extremely shortage of qualified health personnel at the primary health care level. Due to shortage, maintenance and under develop of health care infrastructures, many qualified nurses, doctors and physicians are lured away to developed countries in search of lucrative jobs while a few of these went into private practice.

In Nigeria, ninety percent of government and privately owned hospital still use file systems to store their day to day activities and information; thereby putting at risk the loss of information should any of the files get missing. With this problem in mind, the Nigeria government passed a law in 2011 requiring all government and privately owned hospitals must go digital by the end of 2014 at the latest so as to avoid any information loss and to centralize the health information of all Nigerians. To this end, Rainbow Specialist Medical Center, which is situated in Lagos Nigeria, decided to build software that will store the information of their daily activities including but not limited to registration of patients in accordance to the law that was passed. (Helen Chapin Metz, ed. *Nigeria: A Country Study*. Washington: GPO for the Library of Congress, 1991.).

The company therefore required me to design software that is able to record all daily activities of their medical center. The medical center therefore suggest the name, hospital management software as the name of their software, therefore the implementation of Hospital Management Software will be called HMS from this point onward throughout this project. This project is a mini contract signed between myself and Rainbow Specialist Medical Center, as part of their effort to move from a file base system to computer based system.

## 1.2 Objective

The objective of this project is to develop hospital management software based on Microsoft window application with structured Query language ( T-SQL and SQL Server as a database) as the back-end database for Rainbow Specialist Medical center, with the hope of migrating

from file based system to a computer database system. This software will help the company to be more efficient in handling the daily activities and registration of their patients. The purpose of this project is to give a complete requirement documentation, design, and implementation of the software. It also explains the user interface, hardware and software and different models that could be used to develop software such as this.

### 1.3 Scope

The scope of this project is covers only to the requirement documentation, design and implementation due to secrecy of the project from which this thesis is derived. Therefore, the client who owns this project, Rainbow specialists Hospital, only permit me discuss only the above mention part. Hence, the testing and installation are not included here as agreed upon by both of us.

### 1.4 Used terms Definition

MS SQL	Microsoft Structured Query Language
GUI	Graphical User Interface
RAM	Read Access Memory
HMS	Hospital Management System
EDM	Entity Data Model

### 1.5 System Requirement

Hardware Requirement		
Processor	RAM	Disk space
	64 Mb or Higher	130 Mb

<b>Software Requirements</b>		
Operating System	Database	User Interface design
Window XP, Window vista, Window 7 or any other higher window version	Microsoft SQL Server Man- agement studio	C# 3.0 Data Report Visual studio 2010

## **1.6 Methodology**

This project, HMS, is developed using Microsoft Visual Studio 2010 and Microsoft SQL Server 2008 database on iterative and incremental development models with the collaboration of the system's user

## **1.7 User Characteristic**

From the requirement specification of my client, it is stated that all users of the software program are required to have the at least all of following characteristic

- All users must be comfortable with using a computer.
- Users must be knowledgeable in medical field.
- Basic knowledge is a must to use this software



## 2 Software Development Process Model

In an attempt to build any software system that follows the required standards requested by the client, it is necessary to follow some laid down software modeling rules. Therefore, software development process models will be considered in at this point.

Software development process model is the abstract representation process which serves as the basis for planning, organizing, coordinating, budgeting and directing a software development. (Dogru, A. & Tanik, M. 1991)

All models follow the basic foundation as follows:

*Planning:* This is the initial evaluation of the software to be developed or in most cases; it is an attempt to upgrade an existing system.

*Requirement Analysis and Specification:* In this stage, the problem the new system is identified; its operational capabilities are defined and the resources needed for the system and maintenance goals are set. (M.Ridley, Requirement analysis and specification, 2006)

*Functional Specification or Prototyping:* The computational aspect, relation and attributes of the software system are identified. Also, the operations that will transform these objects along with the constraints of the system behaviors are considered at this point.

(Archer, P. Personalized Access to Cultural Heritage Space. Project, 2009)

*Architectural Design and Configuration Specification:* The interconnection between all the component and modules of the system are defined in such a way that they meet the detailed design.

(Bersoff, E. H. (1984).)

*Detailed Component Design Specification:* This is the stage in which the modules of the defined component are transformed from required inputs into provided outputs.

(Spriestersbach, A. Resource development and Management, 2011)

*Component Implementation and Debugging:* codifies the preceding specifications into Operational source code implementations and validates their basic operation.  
(P. Graham, B. Nelson, and B. Hutchings, 2001)

*Software Integration and Testing:* This is the stage where integrity of the software system configuration is tested and verified for consistency and completeness. All developed modules, interfaces and resources are verified against their specification.  
(Goa, J. San Jose state University, 2002)

*Documentation Revision and System Delivery:* At this stage, the technical know-how of the developed software system is documented as user's guides.  
(Goa, J. San Jose state University, 2002)

*Deployment and Installation:* Installation of the software system into local computing environment, operating systems configuration and diagnostic test cases to ensure the workability of the system is done at this stage.  
(Goa, J. San Jose state University, 2002)

*Training and Use:* Providing training support for the users of the software system is as important as the development of the software system itself, since the user of the system needs to have a good understanding of the system itself.  
(Goa, J. San Jose state University, 2002)

*Software Maintenance:* Maintaining the system in the environment in which it is created for is also an important aspect that must be taken into consideration.  
(Goa, J. San Jose state University, 2002)

There are many models available for software development processes and many of these have been used for many years. Although, most of these models have found their places in software development processes but only few are suitable for window application program. This thesis will view the following models:

1. Waterfall model
2. Iterative and incremental model
3. Spiral model
4. V-shape Model

## **2.1 Waterfall Model**

This is one of the most classical and oldest models of software engineering. It is one of the oldest models which are mostly used by many organizations for their projects in order to reach their desired goal. (Royce, W, 1970. IEEE WESCON)

In this model, planning every phase at the early stage is important because it ensure the flow of design before they are developed. Additionally, the extensive documentation and planning of the project makes this work extremely well for projects which quality control is important. This model starts with the definition /analysis phase which is the research and brainstorming the requirements like for the project, like the software and hardware need for the project. The basic design phase comes next which is the making formulation for the required software on paper. When these two phases are agreed upon, then technical design / detail design are planned. In this part, the technical details are elaborated, functions like modules and program of each software parts are agreed upon. Thereafter, implementation phase which include coding and debugging are started before the testing phase which includes testing the whole system to make sure all the functionalities are well implemented. When it is certain that all these phases have worked out as required, the integration phase, which is putting into use the whole of the system by the company which requested the development of the software. The maintenance phase is required to ensure that this software continues to work properly as needed. Many people who are in support of this model believe that, to fix a problem at early stages of a project is more cost effective and requires less effort than doing so after months or year

when the project has been completed. The figure below shows the progress flow of waterfall model (Royce, W, 1970. IEEE WESCON)

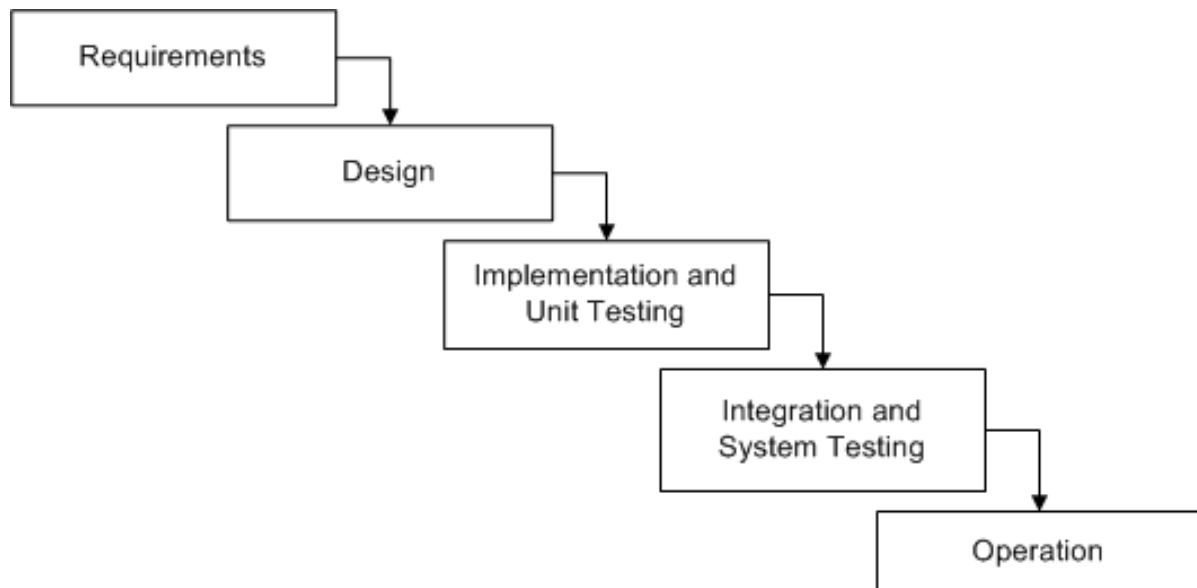


Figure 1. Waterfall development model (Ustudy student portal, 2012)

The waterfall models do have it short comings because once a step is completed; one cannot go back to it to make adjustment again. If for instance after the requirement and the design phases are completed already but somehow new feature are to be included in the final solution of the project, then there will be problem as long as the waterfall model is followed to the letters but as stated in the background of this thesis, only the requirement and design of the software will be the focus point.

## 2.2 Iterative and incremental Model

This model is developed in response to the shortcoming of waterfall model. It does not start with full specification requirement of a project, rather specify and implement some part of the software one at a time in other to review it at every step along the line to identify any further requirement. These processes have to be done again and again to produce a new requirement

version for the software and as shown in the figure below, the process starts with the initial planning, and then moves further to the real planning stage before the requirement stage. (Larman, C. & Basili, V (Dr). History of Iterative and Incremental Model 1980)

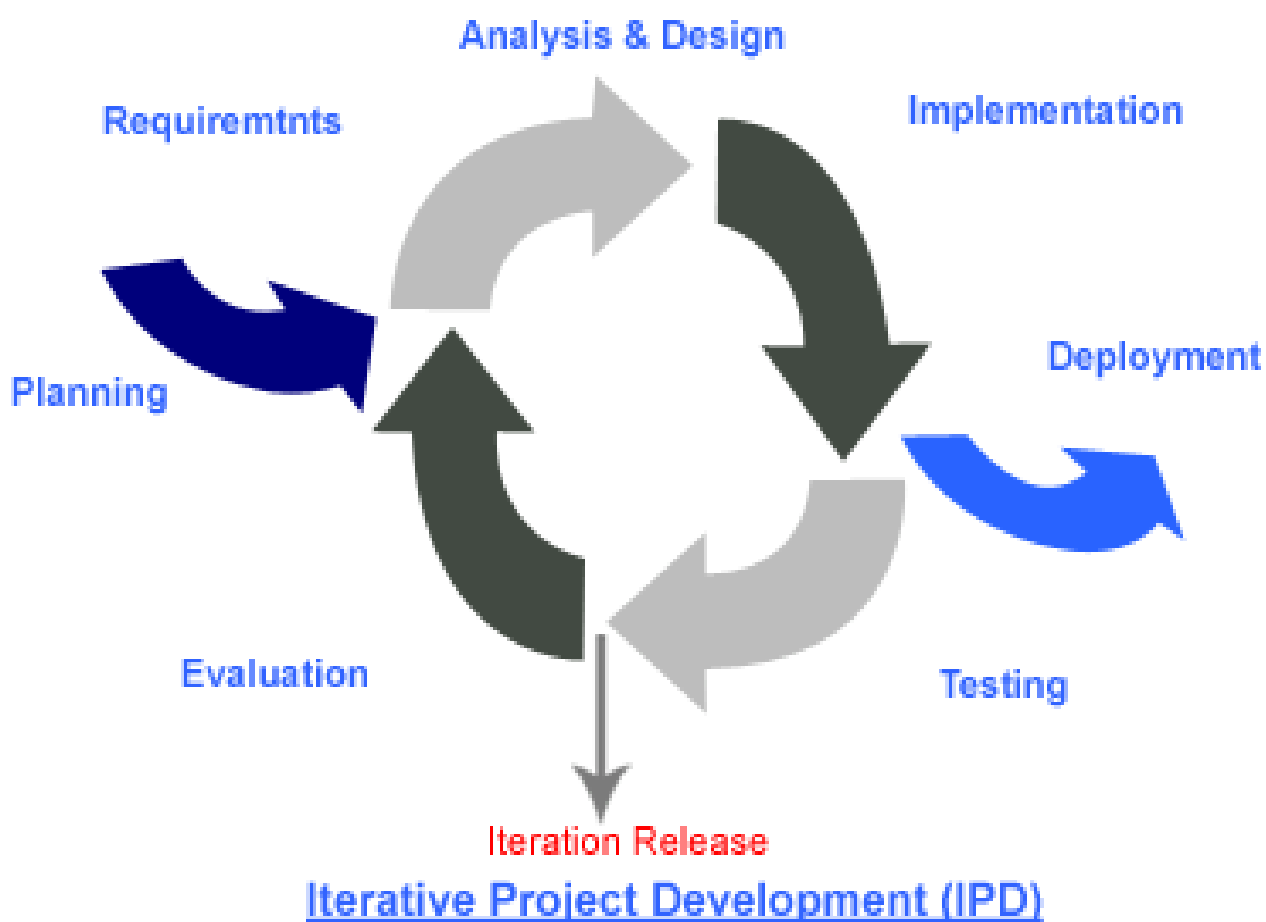


Figure 2. Iterative and incremental software model (Harpal, S., 2012)

Iterative development divides the deliverables into different stages. Within each stage, functionality is implemented using the cross-discipline work, which actually starts with the identification of the model to execution.

(Larman, C. & Basili, V (Dr). History of Iterative and Incremental Model 1980).

**Iterations models follows the phases listed below:**

**Inception:** In this phase, the scope of the project, requirements for functional and non-functional parts included.

**Elaboration:** Complete detailed assessment of the project which includes risk assessment

**Construction:** This phase is the most important and delicate part of the project because at this stage, the architectural part with the already written code from the analysis, designs, implementation and testing are incrementally inscribed into the project.

**Transition:** The last phase of the project is the point where the system is facilitated with the operation environment

Each of the aforementioned phases can be illustrated in single or multi-face iterations. These iterations are mainly time-bound instead focused on features. The role of the experts is clearly allocated and distributed, that is architects and analysts assess and write out the iteration leaving the work-product backlogs to developers and testers.

(Larman, C. & Basili, V (Dr). History of Iterative and Incremental Model 1980)

### 2.3 Spiral Model

In 1988, Barry Boehm was the first person to define the spiral model when he wrote an article, model of software development and enhancement. He defined the spiral model as a system development method (SDM) which combines the features of the prototyping model and waterfall model but it is only appropriate for some certain types of software development projects and most often it is intended for large, expensive and complicated projects.

(Boehm, B. *Software Risk Management*, IEEE Computer Society Press, 1989)



Figure 2. A spiral development model (Society for Risk Analysis (SRA), 2002)

There are series of four steps phase involved in the spiral model as shown in the figure above and they can be summarized as follows:

1. The first step of the first phase is to determine the objectives, alternatives, constraints, milestones and schedules.
2. The goal- the requirement for the software and all the constraint that might be encountered and possible alternatives that may be used during the development.
3. The risk analysis task is also assessed from the point of view of technical and management's aspect.
4. The engineering task- one or more prototype or samples of the application is designed.

The main difference between spiral model and all other software models is that the spiral model explicitly evaluates the risks involved in the development of the software. Other models do have risk management as part of their process but they do not have representation in their business process. In other model, the risk management is not part of the main tasks but

just part of the sub-task. Another difference that the spiral model has is that it has no particular phase for specification requirement, design or testing although, requirement could be found and defined most likely by prototyping.

Spiral model has much more advantage over other models because the developmental approach it reflects in many industries are lacking in other process model available and also uses an approach whereby some part of hardware sample phases are maintain in cases where the product being produced in a particular environment is not just software alone but also involve hardware development. Therefore, in such scenario, the developers and customer will better understand the risks involved much early and measures could be taken to correct them during the developmental stage of the soft-ware. This in particular gives spiral model more advantage over other developmental models. Some other advantages of the spiral model are listed below. (Kan, S. Metro and Model in Software Quality Engineering, 2002)

### **Advantages**

1. It is good for large and mission-critical projects
2. Most part of the software is produced early in the software life cycle
3. It has high amount of risk analysis.

Spiral model is not without its own disadvantage as listed below.

### **Disadvantages**

1. It has risk assessment as an essential part of it structure even when it may not be necessary in when re-engineering or updating an already developed program.
2. Highly specific expertise is required to assess the risk analysis but it may be difficult to find an expert for such assignment.
3. This model can be very costly.
4. The success of any project that this model is used for is dependent on the risk analysis phase.
5. It does not work well for smaller projects.



## 2.4 The V-Model

This model puts a lot of emphasis on the sequential path of execution of the process. That is, each phase developed must be finished before the next one is embarked. This is exactly like the waterfall model, although this model emphasizes more on testing than the waterfall. The procedures for testing the software is developed well ahead before the codes are written for each of the phases that precede the software implementation.

(Ratcliffe, A. 2011)

The V-shape model starts with the requirements analysis exactly like waterfall model. The first step taken before the development phase is to create a test plan for the system which focuses a lot on meeting the functionality specified in the requirements analysis phase. The architecture and design phases along with integration test plan are all created and developed at the high-level design phase. This phase is important because the software system needs to be tested so as to know if the pieces developed so far are able to work together as a unit. However, the software components are developed at the low-level design phase along with the unit testing required to test the functionalities of the phase. The coding of the software takes place at the implementation phase and once complete, the remaining developmental phase continues along the right hand side of V of which the testing plans which were earlier created are now put into use. The V-shape model diagram is shown in figure 3 below.

It should be noted that V-shaped model also have its own advantages and disadvantages. They are listed below.

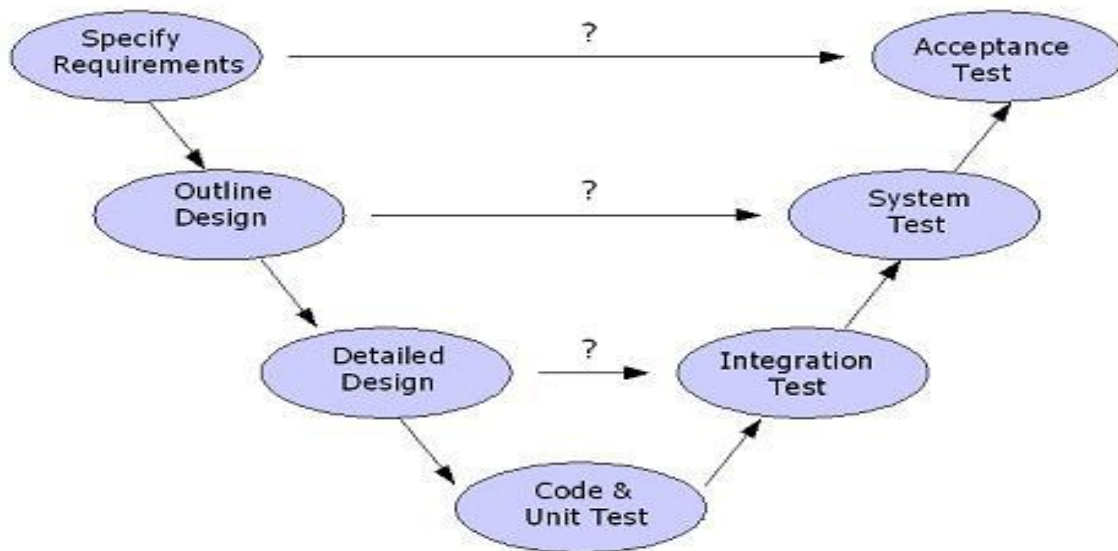


Figure 3. V-shape Model (THahmann Tutorial SW Development Process, 2009)

It should also be noted that V-shaped model also have its own advantages and disadvantages also which are listed below.

### Advantages

1. Very simple and easy to use
2. There are specific deliverables for each of the phases
3. It has higher chances of success because of the early test plan development during the life cycle which the waterfall model does not have.
4. This is an ideal model for small projects which have requirements that can easily be understood.

### Disadvantages

- 1 Developing the software during implementation phase without having to develop a prototype is a big disadvantage of this model.
2. It has a very rigid procedure just like the waterfall model.
3. This model is not flexible; therefore making changes can be very difficult and expensive.
4. If a problem is found during the developmental state of this model, there is no clear path provided during the testing phase

### 3 User Interface Design

There are many factors that must be considered when designing the user interface of a software because the user must be able to interact with the system in a way that the system will understand whatever the input given by the user. Therefore, the quality of the interface and software in general must pass the usability testing standard. Some usability factors, such as fit for use, ease of learning, task efficiency, ease to remember, subjective satisfaction and understandability but all be put into consideration when designing the user interface.

*Fit for use* deals with the system's task support that the user has in the real life. The other factors only deal with the *ease of use* or how user friendly the system is. (Soren Lusen, 2005). Making user interaction as simple as possible when designing the user interface is very important and this must be considered during functional requirement phase of a software design. To create an operational, usable and user friendly interface, the technical functionality with visual element must put into consideration.

In actual fact, there are ten fundamental "heuristics" principles that must be followed when a user interface program is to be design. These are more of a rule of thumb rather than usability specific rules. (Nielsen, J. 2005. Ten Usability Heuristics)

#### **Visibility of system status**

During the design of a system, the user interface design should make sure the interface is able to inform the user what is going on at every point in time with appropriate feedback.

#### **Match between system and the real world**

The system should be able to communicate to the user in the language the user understands rather than computer language that is, making information appear in natural and logical order. The following are 10 heuristics general principles stated by Jakob Nielsen in his book titled "The ten usability heurists":

### **User control and freedom**

User often make mistake by clicking buttons which are not relevant to his/her current task, therefore a clearly mark “exit” button should designed on the interface. Or better still, “undo” and “redo” should to place on the interface too.

### **Consistency and standards**

Platform conventions are important part of user interface design because user should be able to know that a word that is read before still means the same thing when the same is encountered on another line

### **Error prevention**

It is better to carefully design a software where there is no is no error. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

### **Recognition rather than recall**

The user should not be forced to remember what he/she had seen before from one part of the dialogue to another. Objects, actions and many other options should be made very visible so that the user will easily remember while instructions for use of the system should be visible or easily retrievable whenever appropriate

### **Flexibility and efficiency of use**

Accelerators, unseen by the novice user, may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

### **Aesthetic and minimalist design**

Information which is not relevant to the dialogue should not be placed on the design because all other dialogue with compete the relevant information thereby diminishing their visibility.

## **Help users recognize, diagnose, and recover from errors**

Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

## **Help and documentation**

Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

### **3.1 Modules and Interfaces**

There are forty-two major modules and 65 interfaces that are designed for this software but for this thesis, only twenty-two is shown designed so as to avoid bogus time consuming efforts. Few of modules and interfaces designed are listed and explained in details below:

- a. Authentication Management.
- b. Patient Registration.
- c. Diagnosis.
- d. Treatment Information.
- e. Invoice.
- f. Bills.
- g. Patient Appointment.
- h. Staff Information.
- i. Rooms.
- j. Medicine Maintenance.
- k. Suppliers.

#### **3.1.1 Authentication management**

Every user who wants to use this software is authenticated by means of username and password which is already created by the administrator and is stored in the database. All entered parameter of the password are matched with information stored in the database, therefore only authenticated users can log on to the program with limited access. The administrator is

the only one who has access to the entire module and interfaces of the software and can create a new user to access the software.

### **3.1.2 Patient Registration**

Any person who visits the hospital for the very first time either for consultancy, treatment, diagnoses or emergency is required to register as a patient of the hospital. As at the time of registration, a patient identification number is automatically generated by the system for the patient and this remain the same for the patient as long as he/she is a client at the hospital. Patient can either be registered as inpatient(IPD) i.e. patient that are admitted into the hospital or outpatient (OPD) i.e. patients who only come for diagnosis, consultation or other medical issues but are not admitted into the hospital for treatment.

The salient feature of this module is divided into three interfaces which are:

The salient feature of this module is divided into three interfaces which are:

- a. Patient personal Information are recorded.
- b. Guardian personal Information.
- c. Doctor Incharge Information

Patient Information to be included in this application are all defined by the client and are as follows: Firstname, Lastname, Date of birth, Marital status, Religion, Occupation, Telephone and telephone number.

Guardian Information includes: Guardian Name, Address, Occupation and Telephone number.

Doctor incharge information includes: Firstname, Lastname, doctor ID, patient ID, room number, mobile telephone number and office telephone number.

### **3.1.3 Diagnosis**

When a patient had been admitted into the hospital, a test is carried out to know the cause of his/her sickness. All the result gathered from the tests is recorded in these modules.

The salient feature of this module is:

- i. It generate a diagnosis Number automatically.
- ii. All test carried out on the patient can be recorded.
- iii. Provisional diagnosis and remarks can also be recorded.

### **3.1.4 Treatment Information**

This module entails the treatment a patient receives while being admitted to the hospital, the doctor who prescribe the treatment and the staff who updated the information into the system. The salient features of this module are:

- i. Patient treatments information
- ii. Information of the doctor who gave the prescription or medication
- iii. The information of the staff who update the information into the database

### **3.1.5 Patient Invoice**

Indoor billing module has a supervisory role. The entries for payments are automatically sent to the patient payments account from any of the departments, which provide the service. The services are charged as per the category/panel/package applicable. Here the bill is compiled and the payment collected from time to time. Provisional and final bills are generated which provides complete information about the services provided, while in some case payment in advance may be requested before the appropriate receipts for the payments are made. Sometimes refund are made if the patient spend less time in the hospital and the money paid as deposit is greater than the time spent. The salient features of this module are listed below:

- i. Various test that were carried out on the patient
- ii. Date for which the diagnosis is carried out.
- iii. Date in which the patient is admitted
- iv. Remark by the tester if further test is required.

### **3.1.6 Bill**

This is the module that takes care of the patient billing. This is where all the receipt of payments made for the service that were offered to the patient during his/her stay in the hospital

and this module is linked to the invoice in the software system. The registration number and name of the patient is of each patient who is issued a receipt is printed boldly on the receipt so as to identify the patient. Services that are charged in this module are predefined in the software as required by the hospital. The salient features of the module are:

- i. Patient Identification
- ii. total amount due to be paid for the service rendered
- iii. invoice number which is generated from the invoice

### **3.1.7 Appointment**

This module is takes care of outpatients and discharge patients appointment to the hospital either for checkups or consultation with the doctors. The visiting patient either calls in or come in person to visit the available doctor as long as the situation is not critical or an emergency. The salient features of this module are:

- i. Keep track of visiting patient
- ii. Informs the doctor of patient appointment schedule.

### **3.1.8 Staff Information**

All the staffs that work in the hospital have their information stored in this module. When a new staff is employed in the hospital, all information regarding him/her which include name, date of birth, home address, telephone number, next of kin, marital status are store and could be updated should of the information changes. The salary earned by each member staff is also computed in this part of the database, therefore, only the administrator or any person with the financial authority can access this module. It must be noted that this module is where staff identification number is allotted to every person that works in the hospital. The following are the salient feature of this module:

- i. Keep record of all staff.
- ii. The registration of new staff is done here



- iii. Staff ID is allocated in the module

### **3.1.9 Rooms**

The module for the rooms stores information about the room in the hospital, including the occupied and the available ones. When a patient is brought to the hospital and he/she has to be admitted for any reason, the software user checks if there is any available rooms in the hospital each room has its own allocated number and contain a certain number of beds which are numbered too. The salient features of the module are:

- i. Rom allocation
- ii. Bed allocation

### **3.1.10 Medicine Maintenance**

When the hospital buy a whole sale medicine form the pharmacy, all the medicine information are store individual with a specific code attached to each of when it is stored into this data-base. This will give good detailed information of how many were bought and how many were used while also preventing an out of stock situation.

The salient features of this module are:

- i. Keeps record of date the medicine/ drugs were ordered
- ii. keeps information on the availability of each medication
- iii. Gives notification to the administrator when any of the medication reach below a specified amount.

### **3.1.11 Suppliers**

The supplier in this software refers to the drug suppliers to the hospital. It gives information to about the companies who supply all the drugs to the hospital, the contact information, the expiry date of the supplied drug and next day of delivery of the drug should be low in quantity of the already delivered ones and the description of the drug supplied by the supplier. The salient features of this module are:

- i. Gives information about the supplier.
- ii. Gives information about the drug supplied to the hospital by the supplier.
- iii. Gives in detail the content of the

Note: All the above mention interfaces are displayed in the appendices attached to this thesis.

### **3.2 Data Model**

The entity data model (EDM) gives a glimpse of the structure data in the database without regards to the form in which it is stored. As shown in the diagram below, the relationship between all the tables in the database and how each one of them is related to one another are explained. (Microsoft, 2012)

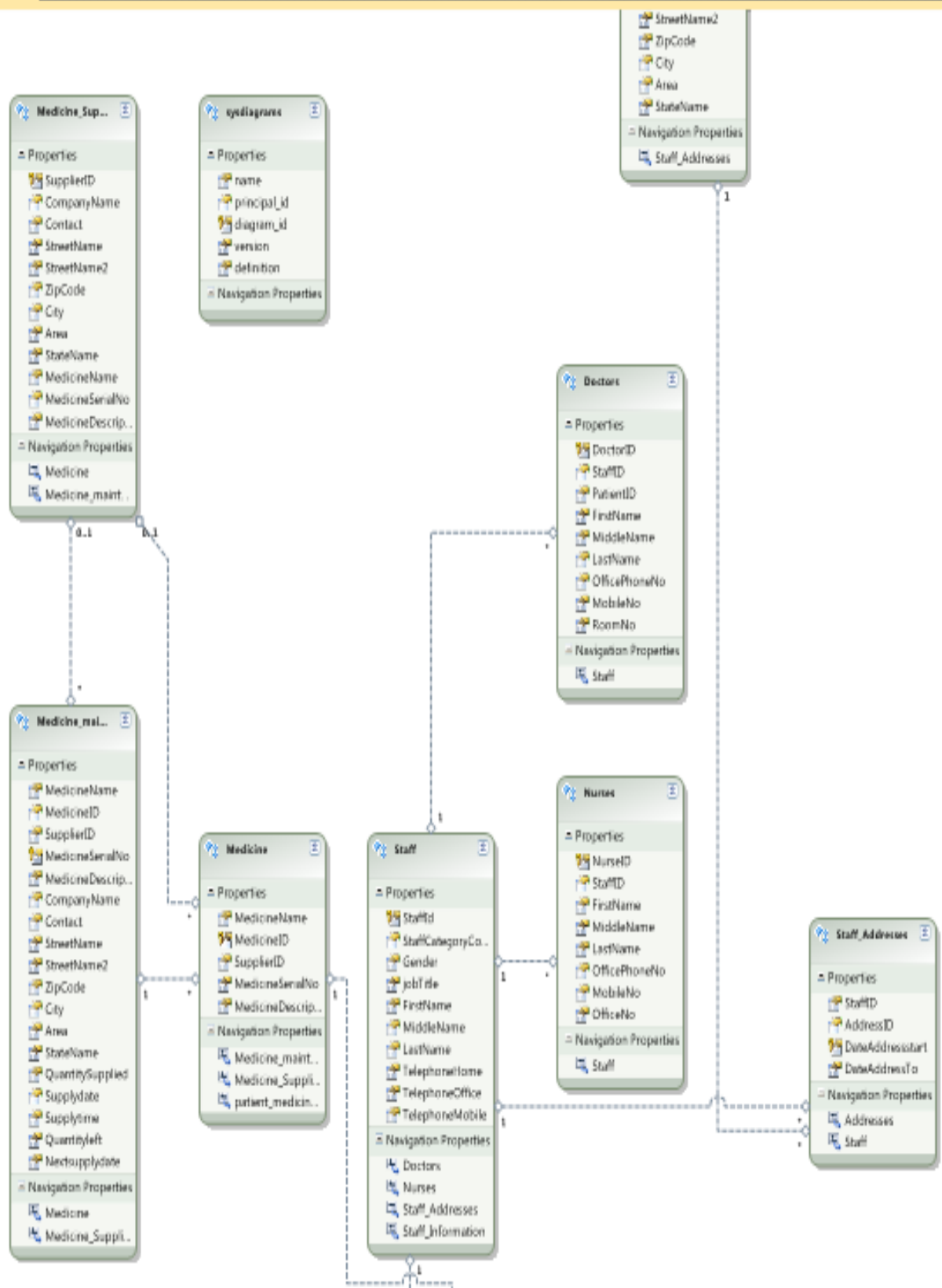


Figure 4. Entity data mode

The about picture shout the shows how the structure each table and relationship between each one of the within the database of the software.

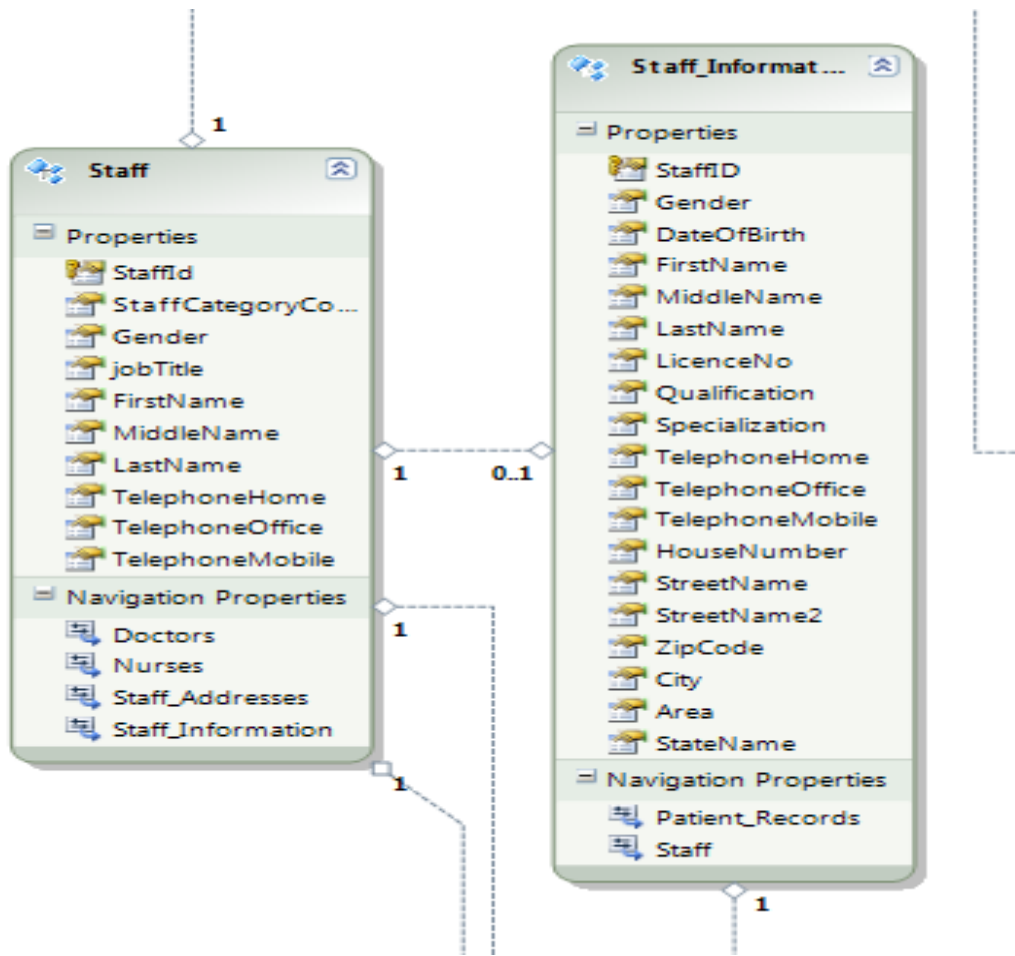


Figure 5

For instance in the above picture there is one and only one group of staff which is indicted next to the table STAFF (1) on the diagram and each staff can either have no information or maximum information which is indicated by STAFF INFORMATION (0..1). It must be noted that all the tables resides in the database (The size of the database is 4.00 MB and automatic expansion is enable in relation the volume of data saved in it)

## 4 Other Functionalities

### 4.1 Security

Every member staff of the hospital requires a username and password to log on to the system. The administrator of this system register each member staff allotting username and password to each and he/she can also revoke access if it is deemed fit for any reason. The data of database are protected through multiple layers of security which includes but not limited to passwords which are encrypted (should the hospital decide to take the software online with networking available but since this is a stand-alone software, the password is not encrypted) but each member is required to protect their password and change on the first logging when created by the administrator.

### 4.2 Performance requirement

For any software developed in this modern time, one of the most important things to do regularly is to update and upgrade and fix whatever bugs found. The following are the list of maintenance required for this software:

- Database archiving
- Password encryption
- Anti-virus protection.
- Password update every 72 days

### 4.3 Error handling

The system also has error and debugging code within the software to prevent system collapse and **krypton 2 professional** encryption programs is embedded within the software security layer to prevent hacking it when installed over internet network. Also, within the code itself, SQL inject susceptible characters had being cleaned up.

#### **4.4 Installation**

After developing and testing the software, the next thing is to deploy the software and run it. For the purpose of this thesis, the installation and deployment of the software will be done from a compact disc (CD) from which the software had been copied to and will be run on a window platform.

## 5 Conclusion

Hospital management software is software meant to computerize the day to day average small hospital management activities and capable of providing easy and very effective storage information including patient registration, patient medical records, doctors and nurses information. Test reports, medication prescription details which include diet advice can also be performed by the system. The billing facility of either inpatient or outpatient is also an attribute of this software and most importantly, a backup facility is included in the software in case of unexpected crash.

Understanding the complexity of software development process and life circle was quite challenging and demanding, therefore a lot of man power, coding and research were done for this project to be completed. The main scope of this project is to develop complete package Hospital management software for Rainbow Specialist Medical Center, which was one of the reason development and de-signed took so long but what is shown in this thesis is limited to few modules because it is contract between me and the company.

At the beginning of this project, everything look really simple and easy but as time went by, it was realized that there is more to the development of the software especially the coding part, though at the end of it all it was worth it because the result is satisfying because the main objective was realized

There are still limitations to the development of this software because at the moment, the software is stand alone and cannot communicate with other another computer from another branch of the hospital. Improvement is still needed to further enhance it so that in future it can communicate with other computers which have same software installed on them.

Furthermore, at the moment the response of the software is delayed due to some technicalities that cannot be resolved at the moment due to time constrain of this the thesis. It must be stated at this point that the project is still on-going and many bugs are being fixed as testing goes on.

## Bibliography

Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* 33, 3 (March), 338-348. Quoted: 03.03.1990

Murat M. Tanik, Eric S. Chan, Fundamentals of Computing for Software Engineers, Van Nostrand Reinhold, 1991. ISBN: 0-442-00525-3

URL: <http://www.cs.helsinki.fi/u/przybils/courses/CBD06/papers/01184164.pdf>

(Quoted, 18.05.2012)

Digital Library: <http://computer.org/publications/dlib>. (Quoted, 18.05.2012)

Ridley, M. (2006). Requirement analysis and specification Guide (Crocus Information Limited, Devon, UK). URL: <http://www.cilco.co.uk/index.html> (Quoted, 18.05.2012)

Spriestersbach, A. (2009) Component Design and Open Specification (Resource development and Management), Seventh Framework Program. FPF-ICT-2009-5. A research sponsored by the European Union.

URL:[http://4caast.morfeoproject.org/wpcontent/uploads/2011/09/4CaaSt\\_D4.2.1\\_Components\\_Design\\_and\\_Open\\_Specification.pdf](http://4caast.morfeoproject.org/wpcontent/uploads/2011/09/4CaaSt_D4.2.1_Components_Design_and_Open_Specification.pdf). (Quoted, 18.05.2012)

Royce, W (1970). Managing the development Of Large Software System. *IEEE WESCON proceeding page 1-9*

Archer, P. (2009) Personalized Access to Cultural Heritage Space. Project sponsored by European Union. Grant agreement number: ICT-2009-270082

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256. Quoted: 03.03.1990

Boehm, B. (1989), *Software Risk Management*, IEEE Computer Society Press, Los Alamitos, CA.

Goa, J. (2002) Software Integration and Testing, San Jose state University. URL: <http://www.engr.sjsu.edu/gaojerry> (Quoted, 18.05.2012)

Nielsen, J. (1994a). Enhancing the explanatory power of usability heuristics. *Proc. ACM CHI'94 Conf.* (Boston, MA, April 24-28), 152-158.



Nielsen, J. (1994b). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.

URL: [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html) (Quoted, 18.05.2012)

P. Graham, B. Nelson, and B. Hutchings, "Instrumenting bit streams for debugging FPGA circuits," in *Proc. of IEEE Symposium. on Field-Programmable Custom Computing Machines (FCCM)*, 2001.

Society for Risk Analysis (SRA), 2002.

URL: <https://acc.dau.mil/CommunityBrowser.aspx?id=17607> (Quoted, 18.05.2012)

Mullins, C. 2002. *Database Administration*

Published by Pearson Education Corporate Sales Division, ISBN: 0-201-74129-6 Quoted: 15.12.2011

Vieira, R. (2007). *SQL Server 2005 Programming, Professional*, ISBN:0-7645-8434-0

Quoted: 15.12.2007

Molina, H., Ullman, J. & Widom, J. *Database System The Complete Book* 2nd edition

Published by Pearson Education International, ISBN (13): 0978-0-13-135428-9 Quoted: 02.03.2012

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256. Quoted: 15.4.2012

Connolly, T. & Begg, C. *Database System* 5th edition

Published by Pearson Education, ISBN (13): 978-0-321-52306-8 Quoted: 12.03.2012

Elmasri, R. & Navathe, S. *Database System* 6th edition Published by Berkeley, CA: Apress.

ISBN: 0-13-214498-0 Quoted: 15.4.2012

Bersoff, E. H. (1984). "Elements of Software Configuration Management." *IEEE Trans. Software Engineering*, vol. SE-10, no. 1, January, pp. 79-87

Kan, S. (2002) *Metro and Model in Software Quality Engineering*, Addison-Wesley Professional; 2 editions (September 30, 2002) ISBN-10: 0201729156 Quoted: 18.5.2012

Ratcliffe, A. (2011) *SAS Software Development with the V-Model*, SAS Global Forum 2011.

Paper 124.

URL: <http://support.sas.com/resources/papers/proceedings11/124-2011.pdf>

Microsoft Corporation (2012).

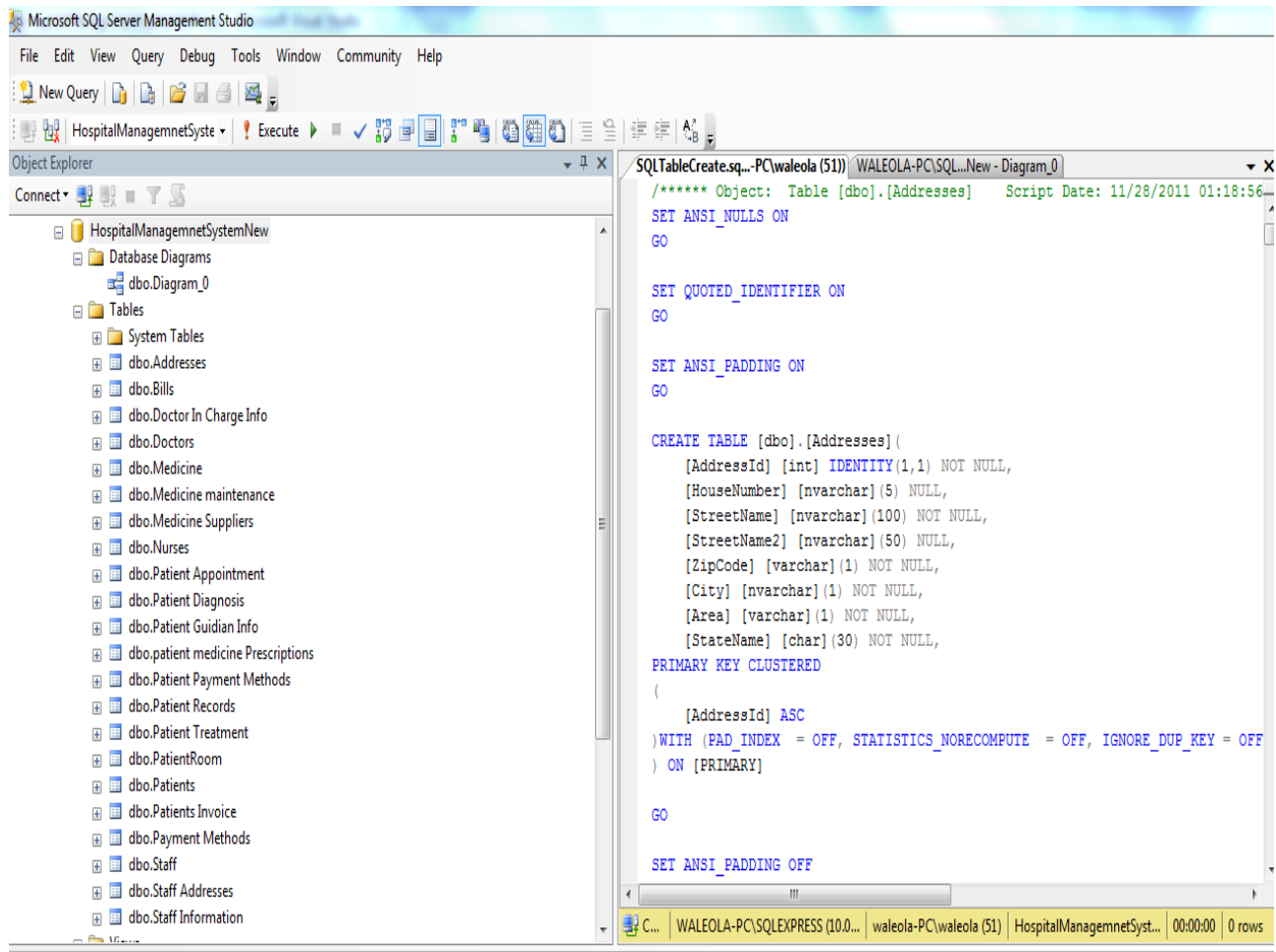
URL: <http://msdn.microsoft.com/en-us/library/ee382825.aspx>

# Appendices

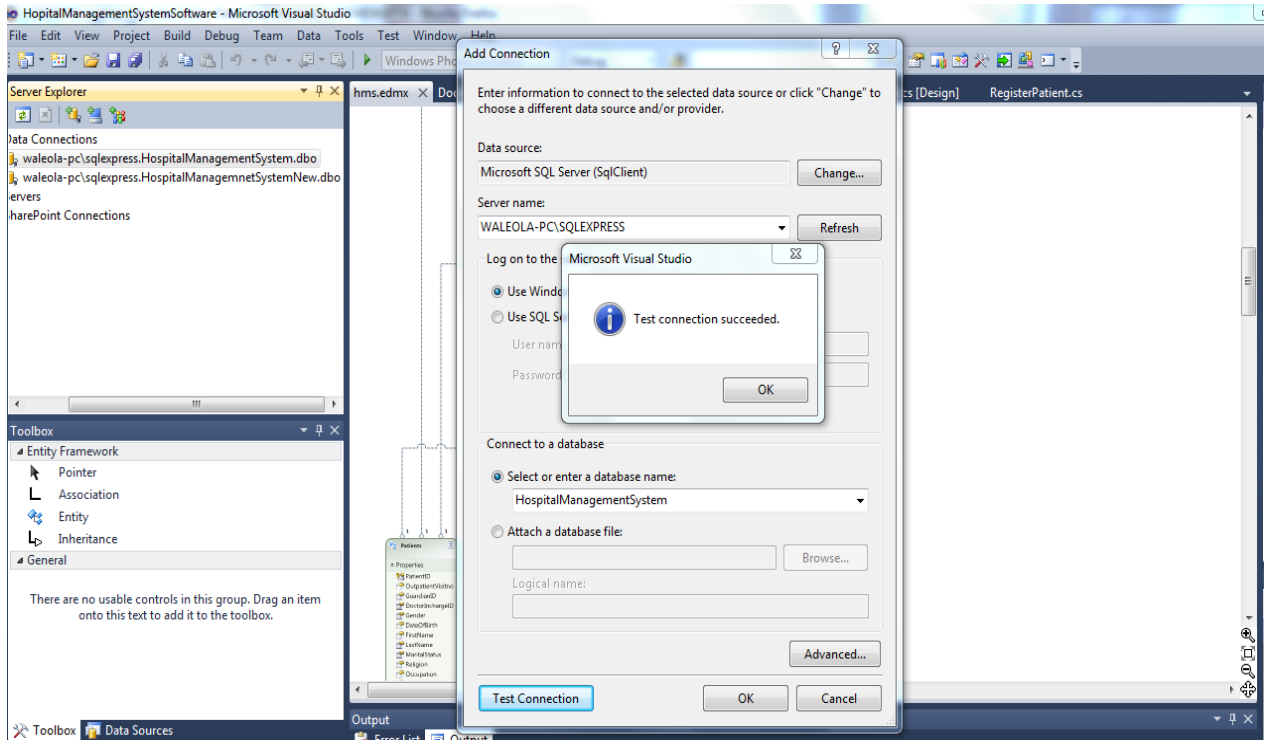
## Appendix 1

### Database and Table Design

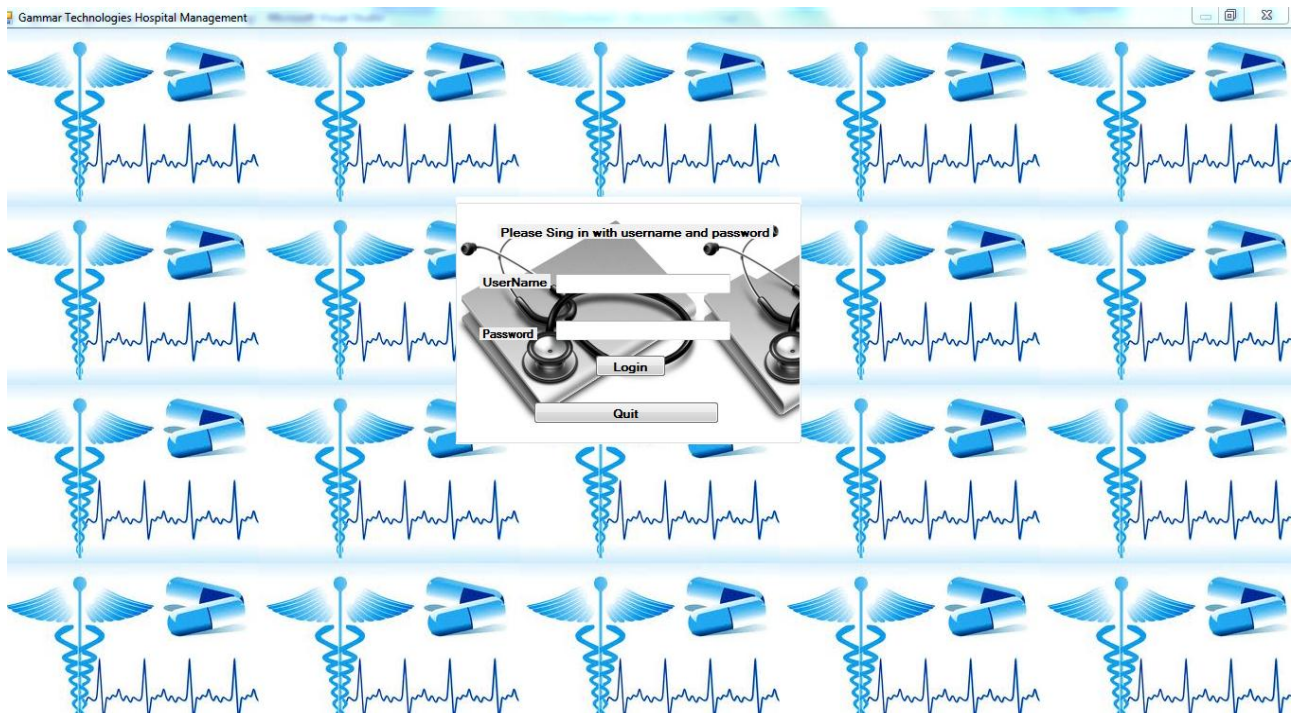
The database and table are very essential to the development of this software because the tables hold the information or records that needed to be stored in the database. Therefore, the script is written in Microsoft management studio and the tables are generated automatically as the script runs successfully as shown in the diagram.



## 1. Database connection to visual studio from Microsoft management studio



## 2. User login interface. This is the first point of entry into the software.



### 3. Patient registration page

The screenshot shows a web browser window titled "Patient Registration". The interface includes a navigation bar with tabs for "Patient Information", "Patient's Guardian Information", "Doctor In Charge Information", and "Room Allocation". The "Patient Information" tab is active. The form contains the following fields:

- Patient ID:
- Outpatient Visitno:
- Guardian ID:
- Doctor In Charge ID:
- Gender:
- Date Of Birth: 14. huhtikuu ta 2012 (calendar icon)
- First Name:
- Last Name:
- Marital Status:
- Religion:
- Occupation:
- Home Phone:
- Work Phone:
- Cell Phone:
- Admission Date: 14. huhtikuu ta 2012 (calendar icon)

Below the form is a section titled "Nature of Illness" with a large text area. At the bottom, there are three buttons: "Save", "Update", and "Delete".

### 4. On this page, all the test and diagnosis result are enter into the database

The screenshot shows a web browser window titled "Patient Diagnosis Information". The interface includes a "Search" button and a "Patient ID" field. The form contains the following fields:

- Patient ID:  Search
- Diagnosis No:
- Diagnosis Date: 14. huhtikuu ta 2012 (calendar icon)
- Bio Chemistry:
- Blood:
- Stool:
- Colonoscopy:
- Urine:
- XRay:
- EGC:
- Sonography:
- Provisional Diagnosis:
- Remarks:

At the bottom, there are three buttons: "Add", "Update", and "Delete".

5. All the information about the patient's treatment and medication given while admitted to the hospital is store to the database from here

Patient Treatment

0 of 0

Treatment ID:

Patient ID:  Search

First Name:

Last Name:

Medicine Name:

Medicine Description:

Quantity:

Issued Date: 14. huhtikuu ta 2012

Doctor Note:

Staff Id:

Firstname:

Lastname:

Telephone Office:

Telephone Mobile:

Add Treatment Update Treatment Delete Treatment

Search Doctor's Information

Patient's Doctor Information

Search Staff Information

Staff updating this Information

6. All payment needed to be paid by the patient is collated at this point

Patient Invoice

0 of 0

Patient ID:  Search

First Name:

Last Name:

Date Admitted: 14. huhtikuu ta 2012

Patient Bill Id:

Diagnos Date: 14. huhtikuu ta 2012

Invoiceno:

XRy Fees:

Remarks:

Diagnosis Fee:

Lab Test Fee:

Blood Test Fee:

Injection Fees:

Consultation Fee:

Medicine Fee:

Total Fee:

Add Delete Update

7. Here the patient is bill and receipt is issued

The screenshot shows a window titled "Bills" with a search bar at the top. The form contains the following fields and controls:

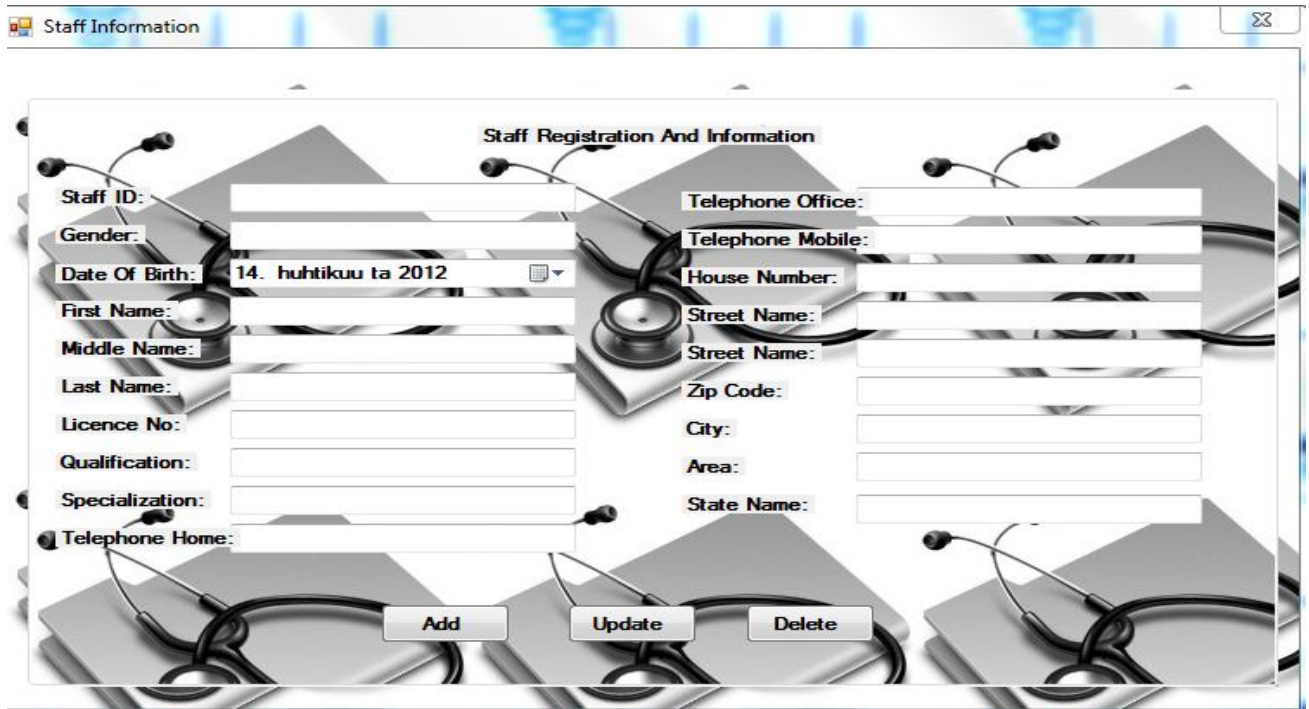
- Patient ID:  Search
- First Name:
- Last Name:
- Patient Bill ID:
- Invoice No:  Search
- Total Amount Due:
- Date To Pay: 14. huhtikuu ta 2012
- Receipt Date: 14. huhtikuu ta 2012
- Other Bill Details:
- Buttons: Add, Update, Delete

8. When a patient needs to see a doctor for checkup or any other reason, this form is filled and is accessible to the doctor

The screenshot shows a window titled "Patient Appointments" with a search bar at the top. The form contains the following fields and controls:

- Patient ID:  Search
- First Name:
- Last Name:
- Outpatient Visitno:
- Doctor ID:
- Appointment Date: 14. huhtikuu ta 2012
- Buttons: Add, Update, Delete

9. Everyone that work in the hospital has his/her information stored here

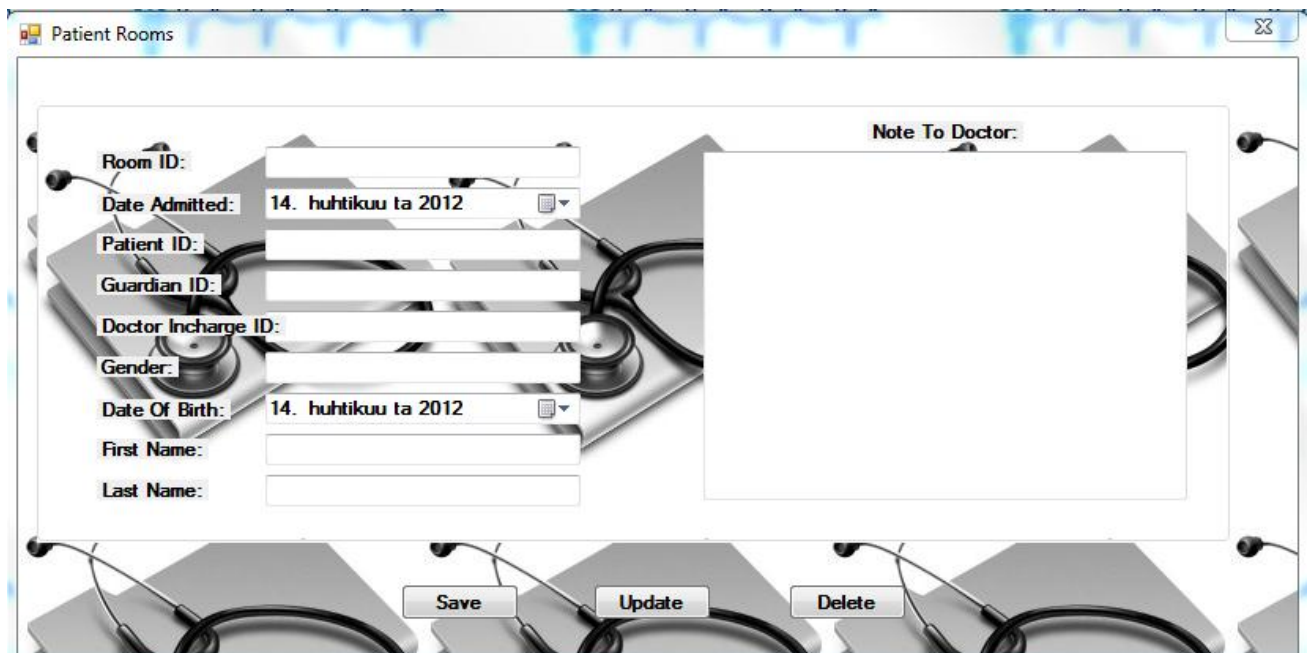


The screenshot shows a window titled "Staff Information" with a close button in the top right corner. The main content area is titled "Staff Registration And Information" and contains a form with the following fields:

- Staff ID:
- Gender:
- Date Of Birth: 14. huhtikuu ta 2012
- First Name:
- Middle Name:
- Last Name:
- Licence No:
- Qualification:
- Specialization:
- Telephone Home:
- Telephone Office:
- Telephone Mobile:
- House Number:
- Street Name:
- Street Name:
- Zip Code:
- City:
- Area:
- State Name:

At the bottom of the form, there are three buttons: "Add", "Update", and "Delete".

10. Information about vacant and occupied room is stored here



The screenshot shows a window titled "Patient Rooms" with a close button in the top right corner. The main content area is titled "Note To Doctor:" and contains a form with the following fields:

- Room ID:
- Date Admitted: 14. huhtikuu ta 2012
- Patient ID:
- Guardian ID:
- Doctor Incharge ID:
- Gender:
- Date Of Birth: 14. huhtikuu ta 2012
- First Name:
- Last Name:

At the bottom of the form, there are three buttons: "Save", "Update", and "Delete".

11. Information about drug supplier to the hospital is stored here

Suppliers

Supplier ID:

Company Name:

Contact:

Street Name:

Street Name:

Zip Code:

City:

Area:

State Name:

Medicine Name:

Medicine Serial No:

Medicine Description:

Add Delete Update

12. Information about all the drug/medicine in the hospital store can be found here

Medicine Maintenance

Medicine Name:

Medicine ID:

Supplier ID:

Medicine Serial No:

Company Name:

Contact:

Street Name:

Street Name:

Zip Code:

City:

Area:

State Name:

Quantity Supplied:

Supply date: 14. huhtikuu ta 2012

Quantity left:

Next Supply date: 14. huhtikuu ta 2012

Search

Medicine Description:

Save Update Delete



## Appendix 2

### Prepared Script Execution with MS SQL management studio which generate and join tables

```
USE [HospitalManagemnetSystem]
GO
```

```
/****** Object: Table [dbo].[Addresses] Script Date: 11/28/2011 01:18:56
*****/
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[Addresses] (
    [AddressId] [int] IDENTITY(1,1) NOT NULL,
    [HouseNumber] [nvarchar](5) NULL,
    [StreetName] [nvarchar](100) NOT NULL,
    [StreetName2] [nvarchar](50) NULL,
    [ZipCode] [varchar](1) NOT NULL,
    [City] [nvarchar](1) NOT NULL,
    [Area] [varchar](1) NOT NULL,
    [StateName] [char](30) NOT NULL,
    PRIMARY KEY CLUSTERED
    (
        [AddressId] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

```
GO
```

```
SET ANSI_PADDING OFF
GO
```

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
SET ANSI_PADDING ON
GO
```

```
CREATE TABLE [dbo].[Bills] (
    [PatientID] [int] NOT NULL,
    [FirstName] [char](20) NOT NULL,
    [LastName] [char](20) NOT NULL,
    [PatientBillID] [int] NOT NULL,
    [InvoiceNo] [varchar](20) NOT NULL,
```

```

        [TotalAmountDue] [money] NOT NULL,
        [DateToPay] [datetime] NOT NULL,
        [ReceiptDate] [datetime] NOT NULL,
        [OtherBillDetails] [nvarchar](max) NULL,
    CONSTRAINT [PK_Bills] PRIMARY KEY CLUSTERED
    (
        [PatientBillID] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[Bills] WITH CHECK ADD CONSTRAINT [FK_Bills_Patients] FOREIGN
KEY([PatientID])
REFERENCES [dbo].[Patients] ([PatientID])
GO

ALTER TABLE [dbo].[Bills] CHECK CONSTRAINT [FK_Bills_Patients]
GO

ALTER TABLE [dbo].[Bills] WITH CHECK ADD CONSTRAINT [FK_Bills_Patients Invoice]
FOREIGN KEY([InvoiceNo])
REFERENCES [dbo].[Patients Invoice] ([Invoiceno])
GO

ALTER TABLE [dbo].[Bills] CHECK CONSTRAINT [FK_Bills_Patients Invoice]
GO

SET ANSI_NULLS ON

```