AN EFFICIENT MODEL FOR HEALTHCARE MANAGEMENT

A CASE STUDY OF ASOKORO GENERAL HOSPITAL, FCT, ABUJA.

BY

GARUBA OLUWASEUN

20094771428

ATHESIS SUBMITTED TO THE POST GRADUATE SCHOOL, FEDERAL

UNIVERSITY OF TECHNOLOGY, OWERRI.

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE MASTER OF SCIENCE IN INFORMATION MANAGEMENT TECHNOLOGY

NOVEMBER 2015

CERTIFICATION

This is to certify that this work "An Efficient Model for Healthcare Management" was carried out by **Garuba Oluwaseun**. Reg. Number: **20094771428** in partial fulfillment for the award of the degree of M.Sc in Information Management Technology in the Department of Information Management Technology at the Federal University of Technology, Owerri.

.....
Dr. (Mrs) U.F Eze
(Supervisor)

11/12/15
Date

.....
Dr. (Mrs) U.F Eze
(Head of Department)

11/12/15
Date

.....
Prof. C.C. Ibe
(Dean of SMAT)

15/12/15
Date

.....
Prof. N. N. Oti
(Dean of Postgraduate School)

.....
Date

.....
Prof. V. E. Asor
External Examiner

14 Jul 2015
Date

TABLE OF CONTENTS

CONTENTS

ABSTRACT

An Efficient Model for the Healthcare Management is an element of health informatics that focuses on the administrative needs of a health environment. It is a comprehensive, integrated information platform that manages all aspects of operation in the environment. This study generates a model that seeks to integrate all aspects and major playing parts of the operation within the medical ecosystem using the Agile methodology.

Keywords: Management information system; Hospital management system and Agile methodology

CHAPTER ONE

1.0 INTRODUCTION

1.1 Background of the Study

A significant part of the operation of any hospital involves the acquisition, management and timely retrieval of great volumes of information. This information typically involves: patient‘s personal information and medical history, staff information, room and ward scheduling, staff scheduling and various facilities waiting lists. All of this information must be managed in an efficient and coastwise fashion so that an institution‘s resources may be effectively utilized.

The aim of this study is to develop an efficient model for healthcare management for Asokoro General Hospital, FCT, Abuja that will computerize and automate the Hospital‘s administrative and health processes which will help staff and the general health operations perform at their optimum in tasks assigned to them. This is done effectively and efficiently at a required time at a click of a button.

This was done by looking at the existing system, analyzing its strong and weak points design and implementation of a new system. Interviews, observation and document reviews were tools used in data collection. SQL Server was used for the database management system, C# and ASP.Net were used for design.

Data flow diagram, relationship diagram and the data dictionary were results of the design and implementation saw different interfaces as seen in the last chapter of this project report.

Asokoro General Hospital, is a Federal Government Hospital located at 11 Julius Nyerere

Crescent, off Yakubu Gowon Crescent, Asokoro, Abuja. It offers specialist care in various aspects of medicine and was established by a decision reached by the federal government to establish at least one tertiary health institution in each state of the federation. Its mission statement in line with that of the federal government is to provide qualitative, affordable, specialized/tertiary level hospital care to its citizenry and to ultimately reduce the burden of diseases within the communities, through provision of prompt and emphatic preventive, curative and rehabilitative services.

1.2 Statement of the Problem

The Asokoro General Hospital, currently runs a semi-manual system for the management and maintenance of critical information. This current system generates a sizeable amount of paper work that is difficult to deal with, in terms of storage, retrieval, maintenance and sharing among the medical personnel. The personnel spend more time looking for information than they would spend on health care delivery. A major problem with the current system is that often information is incomplete or does not follow management standards. These require corrections either at billing time or after it has been rejected by say an insurance company thereby causing late/delayed or no payments.

Duplication of records resulting from multiple registration and misplacement of some of them allows for a potentially damaging misinformation of staff. This does not favor the generation of reports in terms of timeliness and accuracy.

Finally, staff scheduling for the wards is difficult and fraught with errors under the current system. Staff‘s scheduling conflicts are common, causing havoc when a ward is either understaffed or

overstaffed. Sometimes, staffs with the wrong skills are scheduled, or staffs are required to work too many consecutive hours.

Though the current manual system is functional, the hospital‘s human and capital resources are not being utilized in an efficient fashion and thus the need for an automated healthcare management system.

1.3 Objectives of the Study

The objective of this thesis is to design and implement an automated efficient model for healthcare management to replace their existing manual, paper-based system. The general objectives of the study are:

1. To automate the core system of the hospital i.e register patients for admission, records of consultancy and consultants, registry, record investigations done on patients from various investigation departments and the pre and post natal care of patients in the Obstetrics and Gynecology.
2. To integrate all functional parts of the hospital into one location thereby enhancing communication and a good network flow.
3. To make the system completely menu-driven and hence user-friendly. This is necessary to allow non-programmers use the system effectively and system could act as catalyst in achieving objective.
4. To ensure data integrity and security by protecting their data against non-authorized users or guiding against loss of patient‘s file or record.
5. This newly proposed system helps to eliminate swapping of patient‘s record and information.

6. To develop a reliable, understandable and cost effective system.

1.4Scope of theStudy

The Scope of this research work will focus on the .net framework to provide a rapid development and deployment of an application

1. An Efficient Model for Healthcare Management Solution which comprises of
    a. Registry/records management information system
    b. Consultancy/Diagnosis Management System
    c. Investigation management information system
    d. Obstetrics and Gyaenecology management system
    e. Nursing management information system
    f. Pharmaceuticals management system
2. The users management system which takes care of the account of users, roles, logs(which takes care of who and when one came on the system and what they did while on the system) and access or permissions.
3. Written report of the project
4. User manual

Asokoro General Hospital, FCT, Abuja will be used as case study for this work.

1.5 Significance of theStudy

The significance of this study is enormous in that it benefits the patient, staff and the general administrators of the system. The benefits that the hospital; would experience upon development/implementation of this model includes, but is not limited to the following:

1. Standardizing data, resulting in fewer corrections and significantly lowering the incidence of missing or incorrect data.

2. Consolidating data stores into one location ensuring data integrity and providing a database for future statistical and management reporting.

3. Reducing time spent by staff filling out forms, freeing resources for more critical tasks

4. Speeding up the billing process by having accurate, timely data, resulting in quicker payments and a better cash flow.

5. Increased error checking to reduce errors made in scheduling, making schedules more reliable, increasing staff morale, and reducing the amount of time spent by administration creating and publishing schedules.

6. Delivers multi-user support, which would provide simultaneous record retrieval access to as many users with necessary record locking.

7. It will provide a totally secure environment through the different levels of access and use, logging attempts to breach the security restrictions and disabling copying and printing of unauthorized records.

8. The application will enable setting up a duplicate database server and updating it throughout the normal operations of the system. This is done in case of a failure of any of the two servers; the system will be able to continue without being interrupted.

1.6 **Limitations of Study**

The limitation of this work include modules like primary healthcare, human resources, operation theatre, utilities and store management systems because of the following constraints

1. Budgetary Constraints: The cost of gathering necessary materials useful of this project is enormous. This is because the hospitals runs over seventy per cent (70%) of its activities on paper.
2. Time Availability: The time available for this project will have an adverse effect on its outcomes as the project focus would be narrowed to ensure that the workload is achievable within the time specified. Taking into consideration the possibility of unforeseen circumstances.
3. Information Availability: Availability of staff members for interview is not encouraging as they were not readily available for questioning or are trying to hoard information possibly for security purposes.
4. Others include energy, convenience, finance, the research topic and the environment.

CHAPTER TWO

2.0LITERATURE REVIEW

2.1 AN EFFICIENT MODEL FOR HEALTH CARE SYSTEM

An Efficient Model for Health Care System refers to a computer system designed to manage all the hospital‘s medical and administrative information in order to enable health professionals to perform their jobs more effectively and efficiently. Moreover, HIS manages all the information processing activities within hospital to achieve high-quality patients care services and medical research. HIS consists of at least two of the following components: Clinical Information System (CIS), Financial Information System (FIS), Laboratory Information System(LIS), Nursing Information System (NIS), Pharmacy Information System (PIS), Picture Archiving and Communication System (PACS), and Radiology Information System (RIS). Each category has its own function, department and users in improving hospital services.

The application of information technology in health care is unceasingly evolving as the quality of patient care in contemporary times seems to depend on the timely acquisition and processing of clinical information related to the patient (Brailer, 2005).

Cholewka (2006) asserted that a significant paradigm shift has occurred in health care service delivery from an era of physician centeredness to emphasis on quality of patient care, from isolationist practices by caregivers to networking in a global world, and from competition to collaboration among practitioners. In tandem with this trend, improvement in technology and advancement in information systems has been adopted in the health care industry as a business strategy to improve the quality of care (Wilcke, 2008).

According to Paul R. Vegoda (1987), Hospital Information Management System (HIMS) is defined as, an integrated information system which improves patient care by increasing the user_s knowledge and reducing uncertainty allowing rational decisions to be made from the information provided.

Dujat, Haux, Schmücker and Winter(1996) view the hospital information system as the entire information processing and information storage subsystem of a hospital, whereby it is not just about computer systems and networks and the computer-based application systems that are installed on them, but it is about the information in a hospital as a whole. HIMS consist of different softwares that are integrated in order to capture data in specific sections of the hospital.

From the various definitions of HIMS which is a subset of an efficient model for health care management, it is understood that HIMS is a very broad area as it encompasses services catering to varied departments and personnel of any hospital and finally satisfying the patient care in its true sense. Hospital Information Systems (HIMSs) are supposed to make the right information and knowledge available to the right people, in the right place, at the right time and in the right form.

The use of computers in medicine dates back to the 1950s with studies that attempted to expand the mental capacity of physicians or dealt with research on electrophysiology. With the evolution of this equipment, especially with the capacity to simultaneously execute various tasks beginning in the 1960s, computers began to be used in the processing of information in large hospitals, in both administrative and financial functions for the collection of statistics and the development of research projects.

The use of microcomputers, beginning in the 1970s, introduced the concept of distributed processing, increasing the number of systems in use in large hospitals

Because this diffusion did not always occur in an organized or homogeneous manner, the initial diffusion of computers in hospitals led to the emergence of islands of computerization, with isolated systems that lacked any form of interconnection and were developed by different teams. The redundancy and the lack of data integrity deterred health professionals, who saw these systems as developed by systems professionals for systems professionals This situation was also investigated by a scientist called McDonald, who analyzed the lack of interconnection of the different systems used by the hospitals, laboratories, and service providers in the healthcare field.

A scientist, Collen in 1986 described the development of approaches in the 1970s that sought to approximate the habitual processes of decision-making with the use of artificial intelligence in differential diagnoses. In the same decade, studies were undertaken in search of a better organization of the healthcare system. With the help of computer-processed simulations, the author established an ideal relationship between medical centers and population demands.

The distributed processing was expanded during the 1980s with the development and greater availability of microcomputers, and the possibility of network communication of such equipment increased in the 1990s. This allowed for the emergence of hospital information systems (HIS), covering medical, administrative, and hospitality areas, although hospitality may be considered as integrated into the administrative area (Cortes, 2008).

In studying an integrated system for the medical field some scientist identified problems such as disbursement rates that were 159% greater than those originally predicted and long time periods needed for installation.

A scientist upon analyzing the implementation of an integrated management system in a large hospital, found that its provided important benefits that outweighed potential difficulties,

facilitating the execution and improving the quality of the services offered. The two cases demonstrated the influence of the chosen system supplier (especially with regard to care, training, and customization capacity) and of the way in which the implementation project was managed, resulting in the generation of different results for both projects.

A recent review of the literature on the computerization process in basic health care between 1980 and 1997 summarizes in its title the current situation, i.e. ―a descriptive feast but evaluative famine‖ (1). The authors pointed out the lack of research on the impact of IT on the health status of the population, and the methodological limitations in the design of the studies published so far.

Kenney(l990) presented an overview of the health care industry's trend toward multihealth system and specific adaptive strategies for social work managers in health care are suggested. Doctors of social work departments in multihealth corporations will need to resolve issues of institutional versus corporate versus professional identity. The emergence of multi health systems possesses major challenges and unique opportunities to the social work profession. Awareness of managerial strategies and critical content areas can help social work leaders enhance the role and contribution of social work in these existing arid complex health care delivery systems.

Onwujekwe (2005) and Ofovwe and Ofili (2005), in separate studies conducted to assess patient and community satisfaction, found discontent with community members who decried the poorly staffed and inadequately equipped Primary Health Centers (PHCs) in their rural settlements compared to hospitals in urban centers. Such demographic disparity in health care accessibility benefits from hospital information technologies and telemedicine to foster collaboration between clinicians in urban areas and those in rural settlements (Ouma & Herselman, 2008).

Electronic medical record systems help to improve access to health care in remote suburban areas and ensure improved maintenance of long-term care (Keenan, Nguyen, & Srinivasan, 2006).

Beneficial uses of information and associated technology as it relates to health care improvement in this model includes monitoring individual and organizational performance, facilitating information sharing among different health care organizations through a multi-agency approach, and empowering individuals by providing relevant information to consumers, thereby helping them to make informed choices (Gillies & Howard, 2005).

Sisniega (2009) asserted that the applications of information and communication technologies (ICT) facilitate ubiquitous and instantaneous communication between organizations and their stakeholders. Information communication technology enables people and organizations to achieve a seamless workflow and effective processes through improved interactions.

Wilcke(2008) defined information literacy that affects medical practice as the ability to identify the need for information and seek, evaluate, and use information in any presented format.

According to Svensson (2002), consumer informatics helps to create virtual communities for sharing of health care information.

A study on electronic medical records by Keenan et al. (2006) found improvement in daily work and enhanced patient care: (a) medication turn-around times fell from 5:28 hours to 1:51 hours; (b) radiology procedure completion times fell from 7:37 hours to 4:21 hours; and (c) lab results

reporting times fell from31:3 minutes to 23:4 minutes. In the same study, transcribing errors for orders declined, and length of hospital stay decreased.

Sammon, et al. (2009) associated patient data analysis systems (PDAs) with enhanced storage and analysis of patient data enabling physicians to reach improved clinical decisions on patient care.

Stone, Patrick, and Brown (2005) opined that effective organization creates specific and strategic objectives, including objectives related to the clinical and operational strategies. Failing to address the interrelationships that exist between the strategies can result in unforeseen negative consequences.

Morath and Turnbull (2005) recommended creating a culture of safety in health care organizations by recognizing and accommodating the multiple complexities of those organizations. A laudable approach would be to take advantage of the ability of large-scale data systems to amass information as means of identifying significant trends, and enable creation of blame-free sanctuaries in which care errors and observations of incompetence receive prompt solutions. Data production and collection requires knowledge to facilitate this undertaking. Various forms of knowledge are essential business asset used for development of new products and services, thereby useful in developing a competitive advantage in the marketplace (Rennolls & AL-Shawabkeh, 2008).

Cohan (2005) expressed a contrary view that investment in information technology does not necessarily transcend to improvement in productivity. Cohan stressed that shortfall in productivity expectations have made industrial leaders more cautious in adopting information technology in their organizational processes. Presenting a balanced view, Farquharson (2009) asserted that adoption of information technology increases productivity but falls short of

expectation in improvement of productivity considering the high capital investment required for implementation. Farquharson surmised that industry productivity paradox exists to some extent with implementation of ICT. Furukawa, Raghu, Spaulding, and Vinze (2006) argue that hospital information systems enhance quality of health care delivery and safety.

Fuji and Galt surmised that some elements of hospital information systems increase patient participation in care process, thereby reducing unwanted outcome of treatment.

Harrison and McDowell (2008) linked the evolution of the LIS technology to advancements in information technology solutions, stressing that LIS has led to an increased awareness in the development of technological solutions designed to minimize medical errors.

Woodside (2007) concur that health care organizations use electronic data interchange to share patient histories, treatment plans, lab results, and insurance information. Sharing the patient's history in an exchange facilitates initiation of care and decreases the chances of errors. Data interchanges that involve physician's orders and pharmacies can protect the patient by detecting prescriptions of incompatible drug combinations, and highlighting potential allergens to patients. Another vital function of the electronic interchange is the verification of insurance benefits. Many providers do not run tests, ship supplies, or provide care without assurance that the patient has insurance coverage and that the insurance company has authorized the expenditures. Electronic interchange between entities helps avoid delays in the approval process and decrease the possibility of poor outcome because of a delay in treatment.

Crane and Crane (2006) reported that numerous solutions for the medication error problem in hospital settings might be averted with the use of an integrated systems approach. However, execution of an organization‘s integrated electronic medical record without use of communication billing software may escalate process breakdowns.

Phillips (2009) stated that the use of an integrated system offers considerable conceptual flexibility and data integration capabilities instead of using one module for electronic records. An integrated records system promotes a user-interface with e-records repository to facilitate storage and eventual retrieval of records

Keenan et al. (2006) opined that electronic medical records system provide an effective educational tool for training of resident doctors and medical students. Health care information technology and e-health offer strong potential in research and development of clinical protocols. Future studies in this area may provide broader implications of health care information technologies applications (Keenan et al., 2006).

As a result of decades of neglect, there is a serious shortage of modern health care facilities. The government has taken steps to promote the development of a basic national primary care program in the villages, but concerns abound about serious lack of specialized health care facilities (Ouma & Herselman, 2008).

A gap in knowledge exists about the exact number of hospital information systems functionally available in Nigeria, but the subjective data project less than 5% implementation of any form of hospital information technology in a country of more than 150 million people (Idowu et al., 2006).

The available literature provides a common position among various authors that disparities exist in the implementation of hospital information systems in developing and developed countries (Grimm & Shaw,2007; Williams & Boren, 2008). Speculated reasons include poor technological and funding support in developing nations, poor management capacity at all levels that ensures seamless workflow, and a complex milieu of health care service delivery. Other possible factors for low implementation include the continual evolution of technology, confidentiality problems with use of hospital information systems, and the poor technological background of the Nigerian society (Herbst et al., 1999; Grimm & Shaw 2007;Krishna et al., 2007).

Hern-Underwood M. J. and Workman(1993) saw that in today's technical and demanding patient care system within the hospital organisation, there is a need for head nurses as nurse managers to be ever more attuned to the climate of their staff. In this study of 34 nurse managers in seven pediatric hospital organizations across a Midwestern portion of the United States, an analysis of fielder leader match scales showed the significance of group climate on retention.

Fallon says that organisation is technology in the broadest sense: processes, procedures, policies, controls, formal authority structures and techniques. Among groups or organisations, it is unusual for changes in sentiment to precede action or organisational rearrangement. Technology and structure must be changed first. The components of each organisational theory and structure were identified and discussed. In addition to understanding one's subordinate and peers, the effective manager understands the organisational forces that exist in the work place. A willingness to listen, communicate, innovate. And lead should result in both effectiveness and rewarding experiences for a manager in the hospital too.

Pineault, Raynald et al(1989) examined the extent to which health counseling practices in 3 hospitals were influenced by patient characteristics, medical care processes and organisational factors. It was seen organisational factors were more important than the patient characteristics in determining health counseling.

Seim, Lerner (1988) presented a useful overview of the planning , design and construction process, emphasizing the importance of the practical application of management skills in the hospital.

In other developing countries, structural deficiencies due to the current economic situation have led to considerable deficits in social policies — including those related to public health care.
Changes in demographic and epidemiological profiles, in urbanization and in the level of industrialization have created a need for new models of health care. Such models attribute an increasing level of importance to primary health care, the strengthening of which is considered central to the improvement of health-care coverage.

CHAPTER THREE

3.0METHODOLOGY AND SYSTEM ANALYSIS

3.1 METHODOLOGY

There are various methodologies some of which are listed below

1. SSADM,
2. OOADM,

3. Prototyping,
4. Expert and
5. Agile Methodology.

The Agile Methodology was employed in this project and below is an expose on the Agile Methodology and how I employed its use in this research.

3.1.1 Agile Methodology

Agile methods are a subset of iterative and evolutionary methods and are based on iterative enhancement and opportunistic development processes. The purpose of having short iterations is so that feedback from iterations N and earlier, and any other new information, can lead to refinement and requirements adaptation for iteration N + 1. The customer adaptively specifies his or her requirements for the next release based on observation of the evolving product, rather than speculation at the start of the project. There is quantitative evidence that frequent deadlines reduce the variance of a software process and, thus, may increase its predictability and efficiency.

The pre-determined iteration length serves as a time-box for the team. Scope is chosen for each iteration to fill the iteration length. Rather than increase the iteration length to fit the chosen scope, the scope is reduced to fit the iteration length. A key difference between agile methods and past iterative methods is the length of each iteration. In the past, iterations might have been three or six months long. With agile methods, iteration lengths vary between one to four weeks, and intentionally do not exceed 30 days. Research has shown that shorter iterations have lower complexity and risk, better feedback, and higher productivity and success rates.

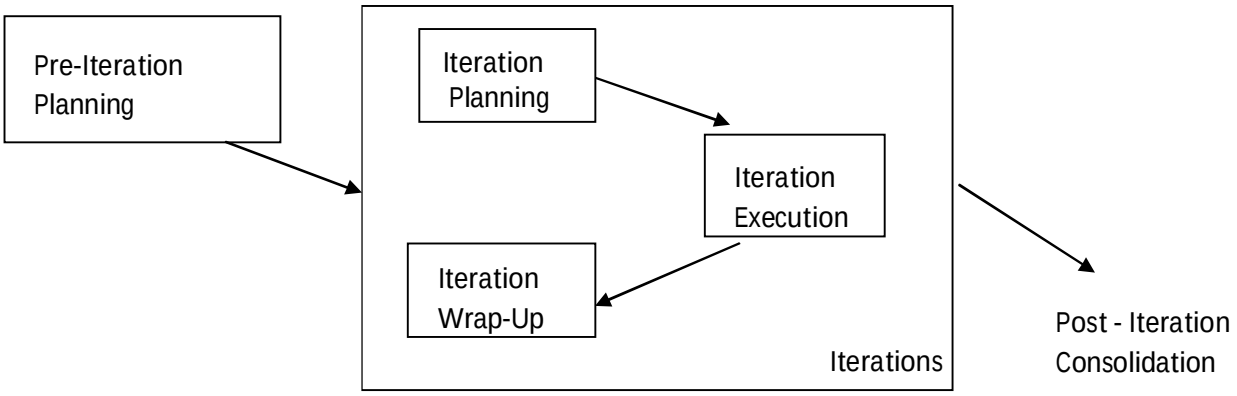


Fig 3.1: A diagram showing the agile development process (Source: Serena, 2007. Pg 6. An Introduction to Agile Software Development)

Why the Agile Methodology?

The agile methodology is preferred to other methodologies because:

1. Formalization of the software process hinders the human and practical component of software development, and thus reduces the chance for success.
2. This method welcomes changing requirements, even late in development. Agile processes harness change for the customer‘s competitive advantage.
3. Agile processes promote sustainable development.
4. The sponsors, developers, and users are able to maintain a constant pace indefinitely.
5. Continuous attention to technical excellence and good design is the watch-word.
6. Simplicity – the art of maximizing the amount of work not done – is essential.
7. At regular intervals, there are reflections on how to become more effective, then tunes and adjusts its behavior accordingly.

How Agile Methodology was employed

In the analysis of the existing system of the Asokoro General Hospital, the users who are key players in various departments relevant to the development write ―user stories‖ to describe the

need the software should fulfill as it concerns their individual departments. User stories help the team to estimate the time and resources necessary to build the release and to define user acceptance tests. A user or a representative is part of the team, so he or she can add detail to requirements as the software is being built. This allows requirements to evolve as both users and developers define what the product will look like. This takes us to the next stage which is the release plan.

To create a release plan, the team breaks up the development tasks into iterations. The release plan defines each iteration plan, which drives the development for that iteration. At the end of an iteration, users perform acceptance tests against the user stories. Iterative user acceptance testing, in theory, can result in release of the software. When the users decide that enough user stories have been delivered, the team can then choose to terminate the project before all of the originally planned user stories have been implemented.

3.1.2 THE ORGANIZATION AND ITS ENVIRONMENT

The Asokoro General Hospital, FCT, Abuja is a public health sector designed to provide health care services to Nigerians and especially the people of the Federal Capital Territory – Abuja to ensure that they all have access to quality and affordable healthcare services. Asokoro General Hospital, is a health care providing body with perpetual succession established under the Nigerian Act, to provide social health care services in Nigeria with an aim to achieve qualitative and efficient health Care delivery at affordable prices within the public health sector.

3.1.3 DEMOGRAPHIC VARIABLE

Demographic variables examined for the Asokoro General Hospital, FCT, Abuja included: sex of the patient, age, patient‘s ethnic group, annual family income, patient's education level, mother's age at the time of the child's birth (19 years or younger, 20 to 29 years, or 30 years or older); number of siblings in the family (one, two, three, or four or more); whether any family member was employed during the preceding month; and mother's marital status. Other demographic variables examined in the Asokoro General Hospital, FCT includes ethnic group of patient, sex of patient, annual family income, highest education of adult patient, poverty index (at or above poverty threshold or below it) and size of the family.

The list below shows the different types of identifying information that will be found in patient‘s table in the database. The use of first and last names, sex, date of birth, and an additional one or more physical characteristic was used.

Patients Having different Kinds of Identifying Information recorded in the Unit Database

a. First and last names
b. Sex
c. Physical attribute (e.g., skin color) (at least one)
d. Date of birth
e. Ethnic group
f. Annual family income
g. Education level
h. Alias

3.2 SYSTEM ANALYSIS

3.2.1 Analysis of the Present System

The records and registry department is the first to be visited upon entry into the hospital, in this department a patient‘s record details are captured, registrations of in/out patient, new born etc are done and they also manage repeated visits and appointment scheduling.

The nursing station is the next station to be visited upon entry into the hospital. In this department, the patient‘s vital signs are captured, the patient is directed to the appropriate clinic or consultant, and if investigations are needed they direct the patient to the investigation department. If it‘s an out-patient, the follow up visits are taken note of. In this department, shift management of both nurses and doctors are taken care of, report on everything that happened during the shift e.g the state of the fan, the number of vacant beds etc as at time of handover. Also in this department, in-patients are divided into medical or surgical patients.

The diagnosis/consultancy section, this is the next department a patient visits. In this section a consultant is met who examines the patient and determines if he/she is an in/out patient, they also determine if the patient is a medical or surgical patient. They treat and prescribe drugs to patient or refer patient if case identified cannot be handled by the hospital.

The Pharmaceuticals department handles all the drugs and medical items available in the hospital, they administer drugs as prescribed by the consultant to the patient, and they take stock of the drugs available.

The investigation department handles clinical investigations done in the hospital e.g hormonal assay, hematology, they keep record of blood bank data and maintain the status of various investigations done in the hospital.

3.3 Information flows

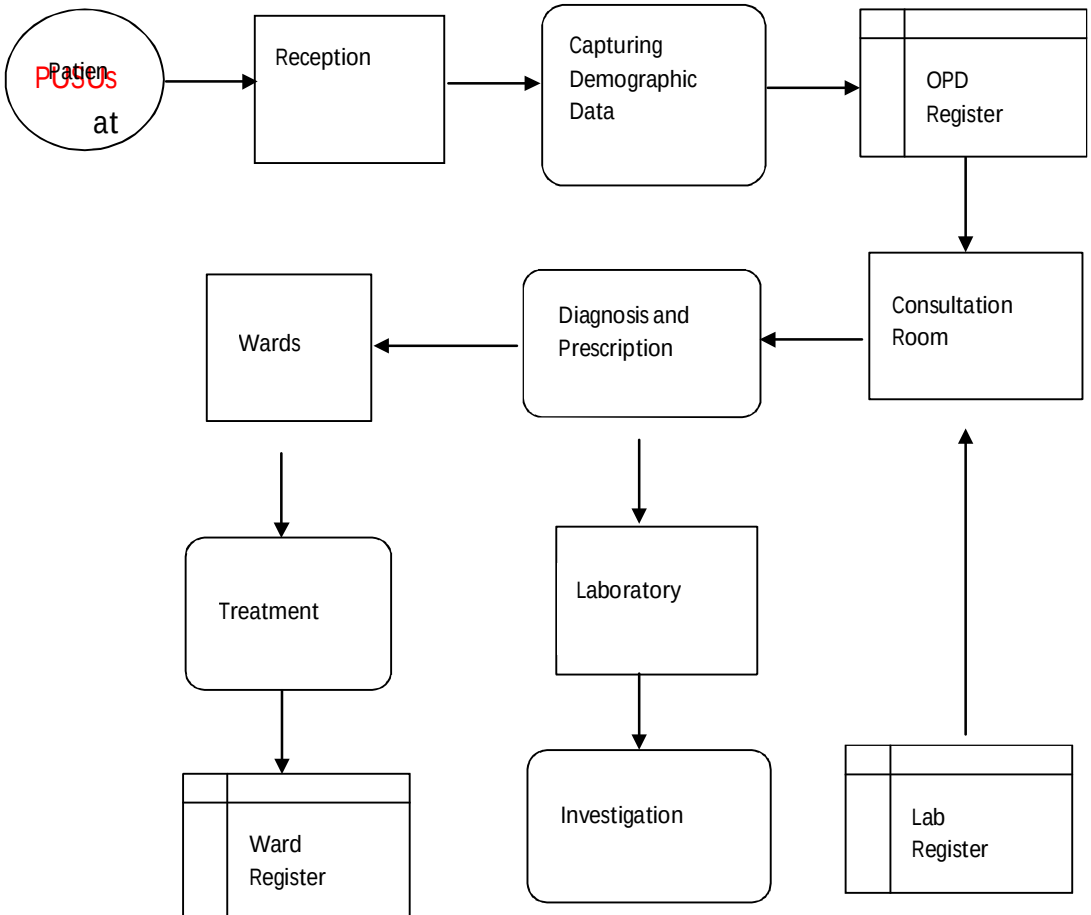


Fig 3.2: Information Flows of the Organization

Pharmacy Data Flow Diagram

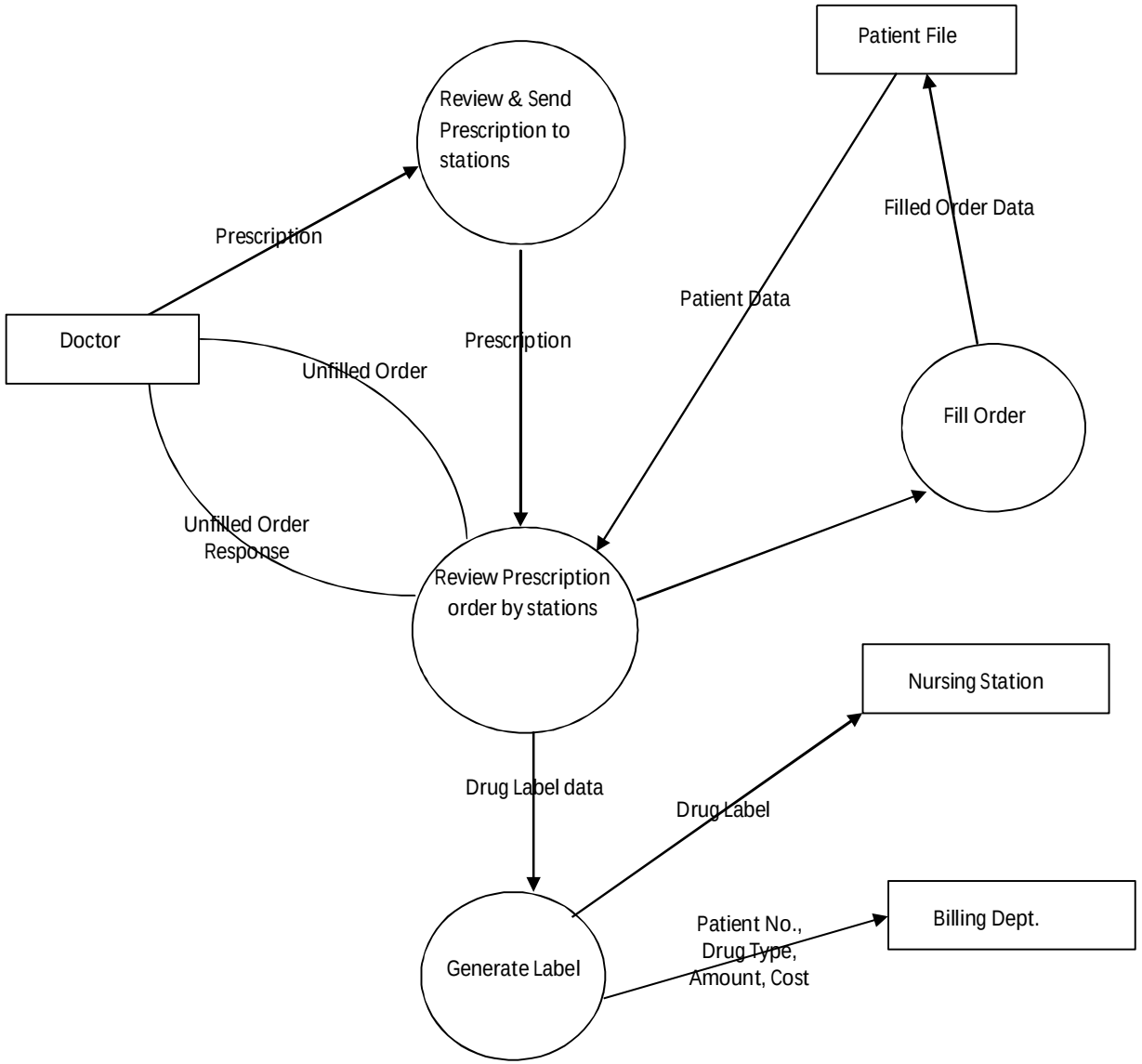


Fig 3.1 Pharmacy Data Flow Diagram

Investigation Data Flow Diagram

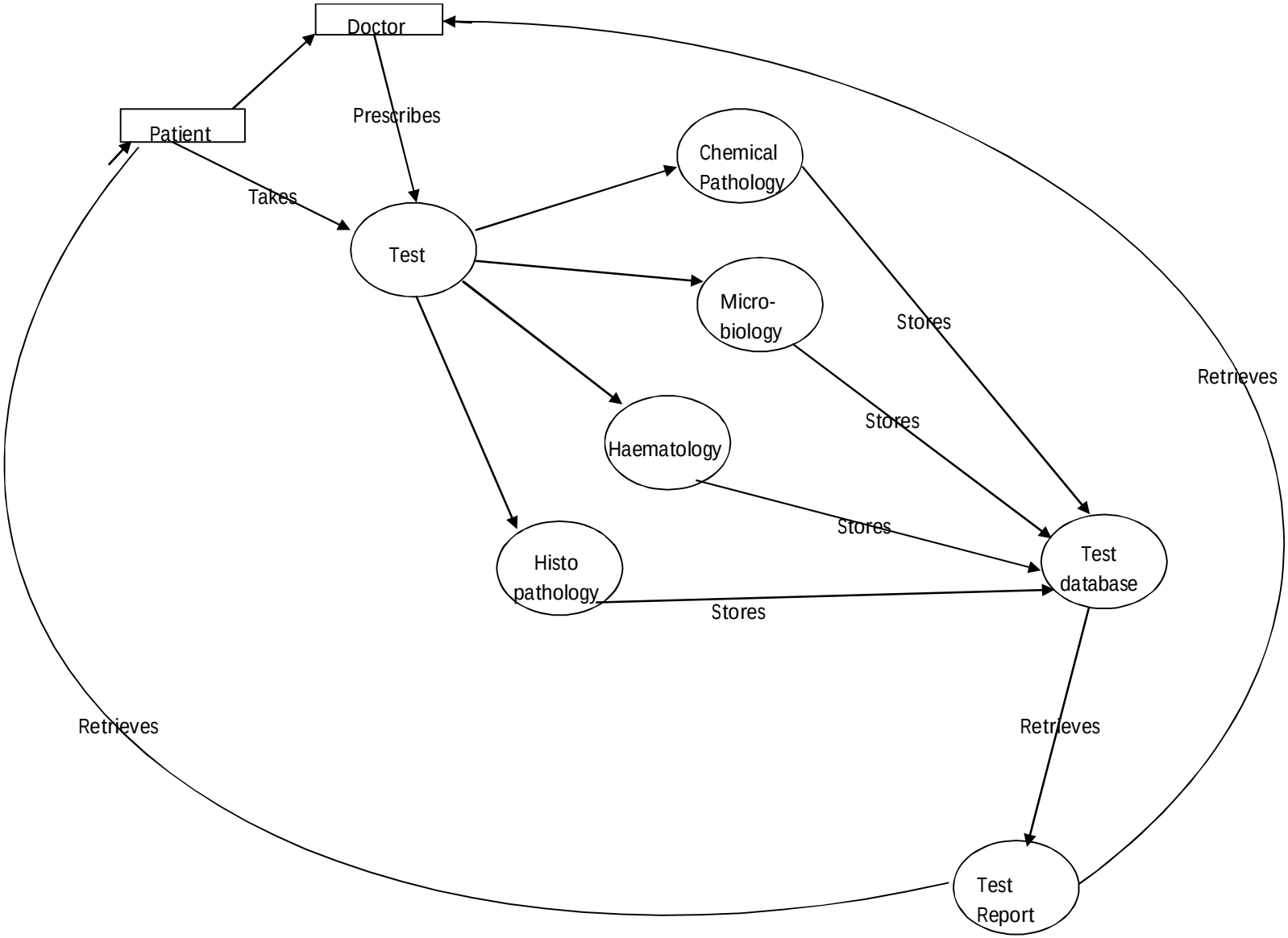


Fig 3.2: Investigation Data Flow Diagram

Obstetrics and Gynecology Data Flow Diagram

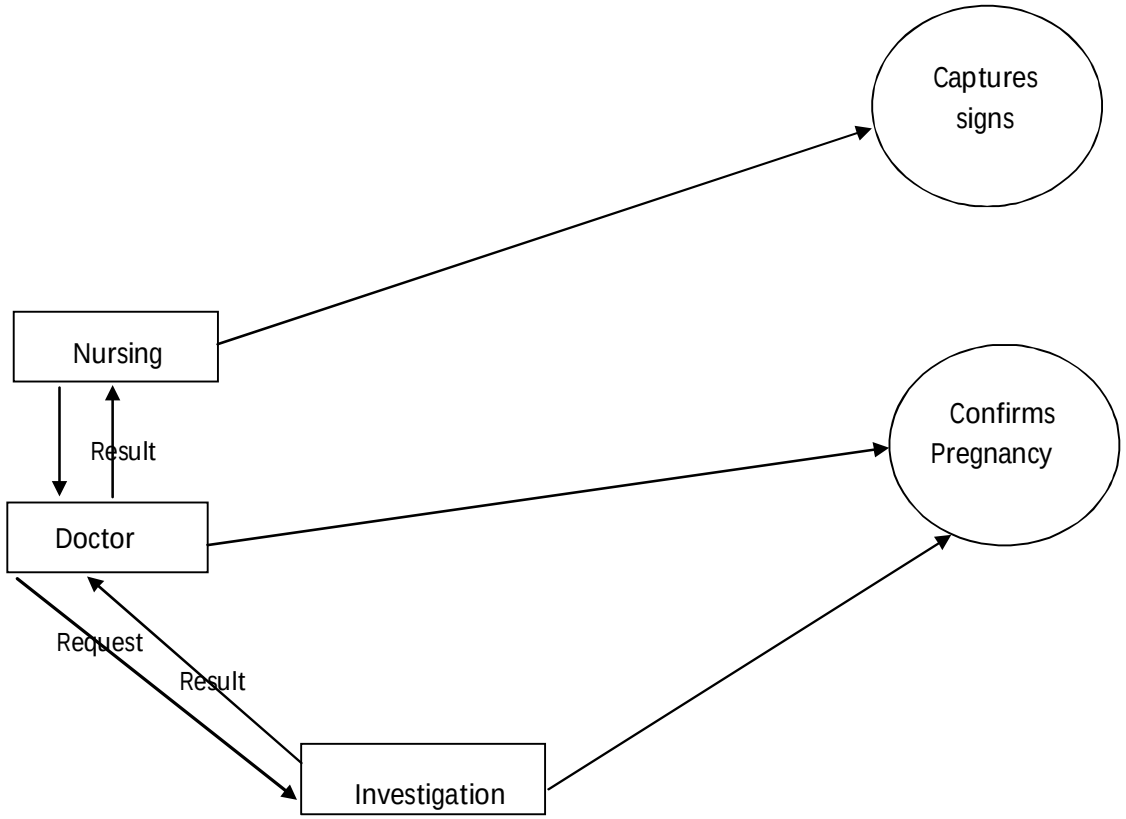


Fig 3.3: Obstetrics and Gynaecology Data Flow Diagram

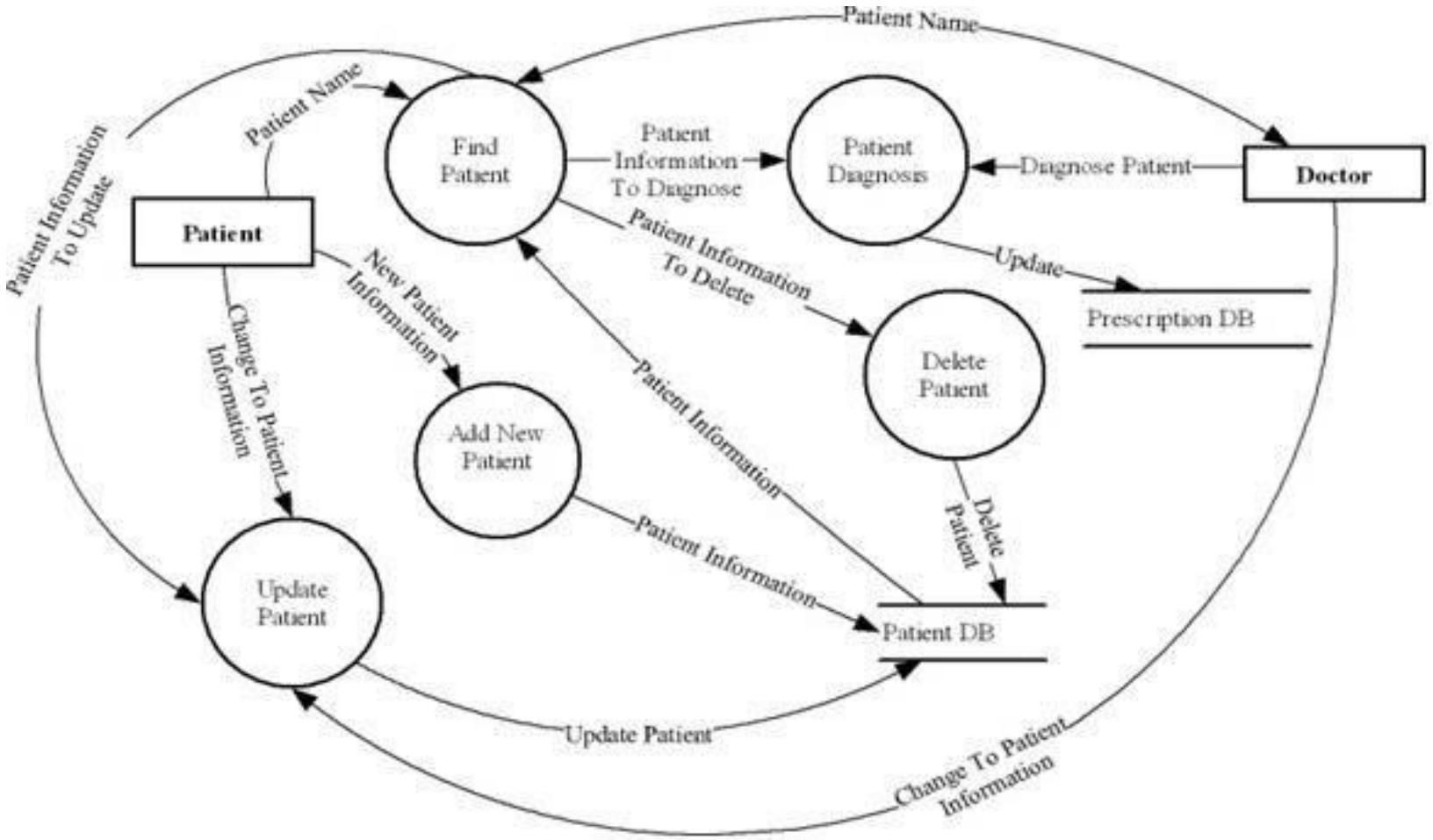


Fig 3.4: Patient Information Data Flow Diagram

Nursing Data Flow Diagram

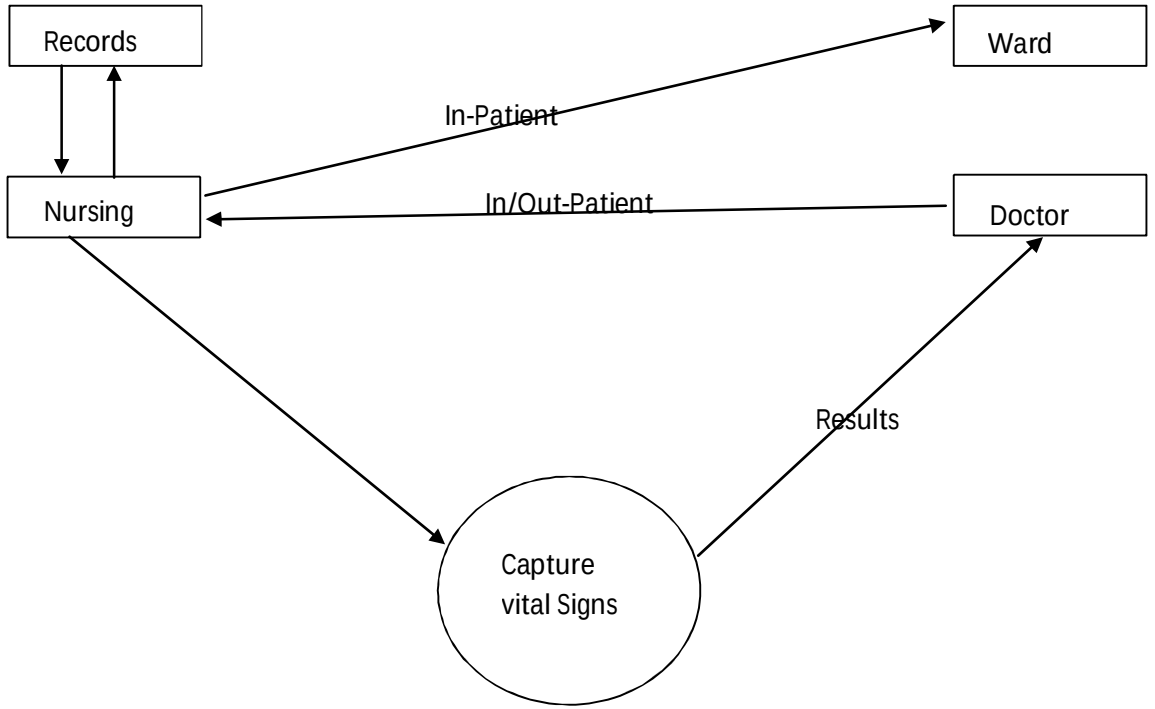


Fig3.5: Nursing Data Flow Diagram

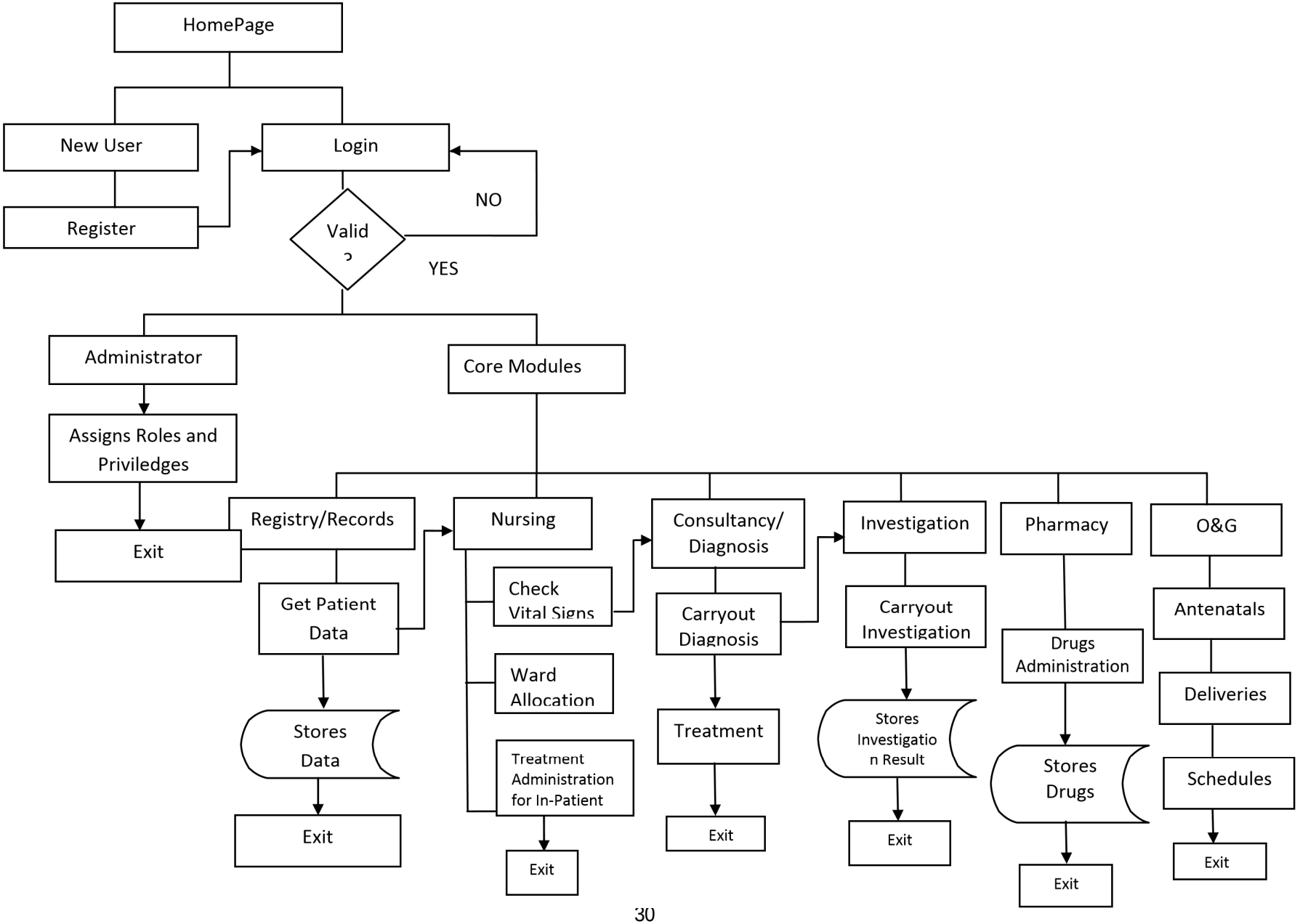

HomePage
New User
Login
Register
Valid
NO
YES
Administrator
Core Modules
Assigns Roles and Priviledges
Exit
Registry/Records
Nursing
Consultancy/ Diagnosis
Investigation
Pharmacy
O&G
Get Patient Data
Check Vital Signs
Ward Allocation
Treatment Administration for In-Patient
Carryout Diagnosis
Carryout Investigation
Drugs Administration
Antenatals
Deliveries
Schedules
Stores Data
Treatment
Stores Investigation Result
Stores Drugs
Exit
Exit
Exit
Exit
Exit
Exit
Exit

Fig 4.2: Program Main Menu

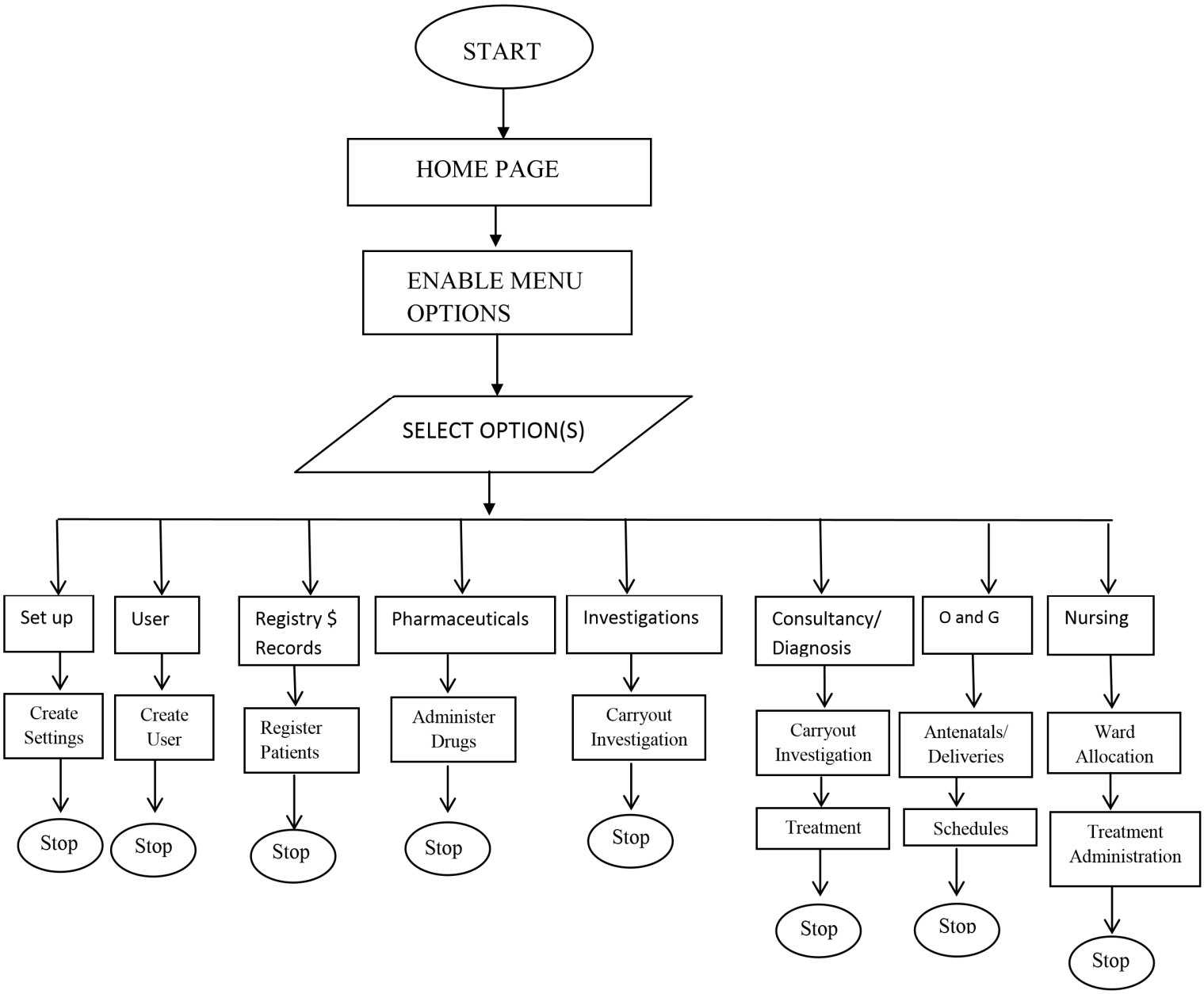
This space was left blank intentionally. Pls insert data flow diagrams here

3.4 Problems Identified

After the research work done in (Asokoro General Hospital,)FCT, the following problems were identified:

1. A lack of planned approach towards working.
2. An inaccurate information system.
3. There was nothing unique as could be used to identify patients(both IP and OP) thereby causing chaos in the entire system.
4. Delays in the retrieval of information.
5. There was no defined record track of IP and OP previous and current medical history poor and limited availability of accurate, complete and timely information.
6. Lack of proper appointment scheduling.
7. Lack of proper documentation for repeated visits.

3.5 High-Level Model



CHAPTER FOUR

4.0 SYSTEM DOCUMENTATION AND IMPLEMENTATION

In this chapter we have discussed the system design which has taken place to ensure a structured implementation process will follow. We have examined both the database element and the interface element stages, outlining the important decisions made and the processes carried out to achieve the design guidelines required. In this chapter a main objective was met which meant the analysis data had been successfully modeled into a design ready for implementation.

The data modeling was carried out to a point at which the data had a balance between the level of integrity and the level that could feasibly be implemented still allowing for a flexible system. As well as discussing the database design we also examined the interface design considering its usability, its layout and its structure.

The design processes and outcomes discussed in this chapter will be followed through into the implementation stage.

4.1 Objectives of the Design

The newly proposed system for the Federal Medical Center, Owo is expected to provide an automated process for the administrative and clinical operations that health professionals need to perform their jobs effectively and efficiently. Therefore the objectives of the system to be designed are:

1. To automate the core system of the hospital i.e register patients for admission, records of consultancy and consultants, registry, record investigations done on patients from various investigation departments and the pre and post natal care of patients in the Obstetrics and Gynecology.

2. To integrate all functional parts of the hospital into one location thereby enhancing communication and a good network flow.

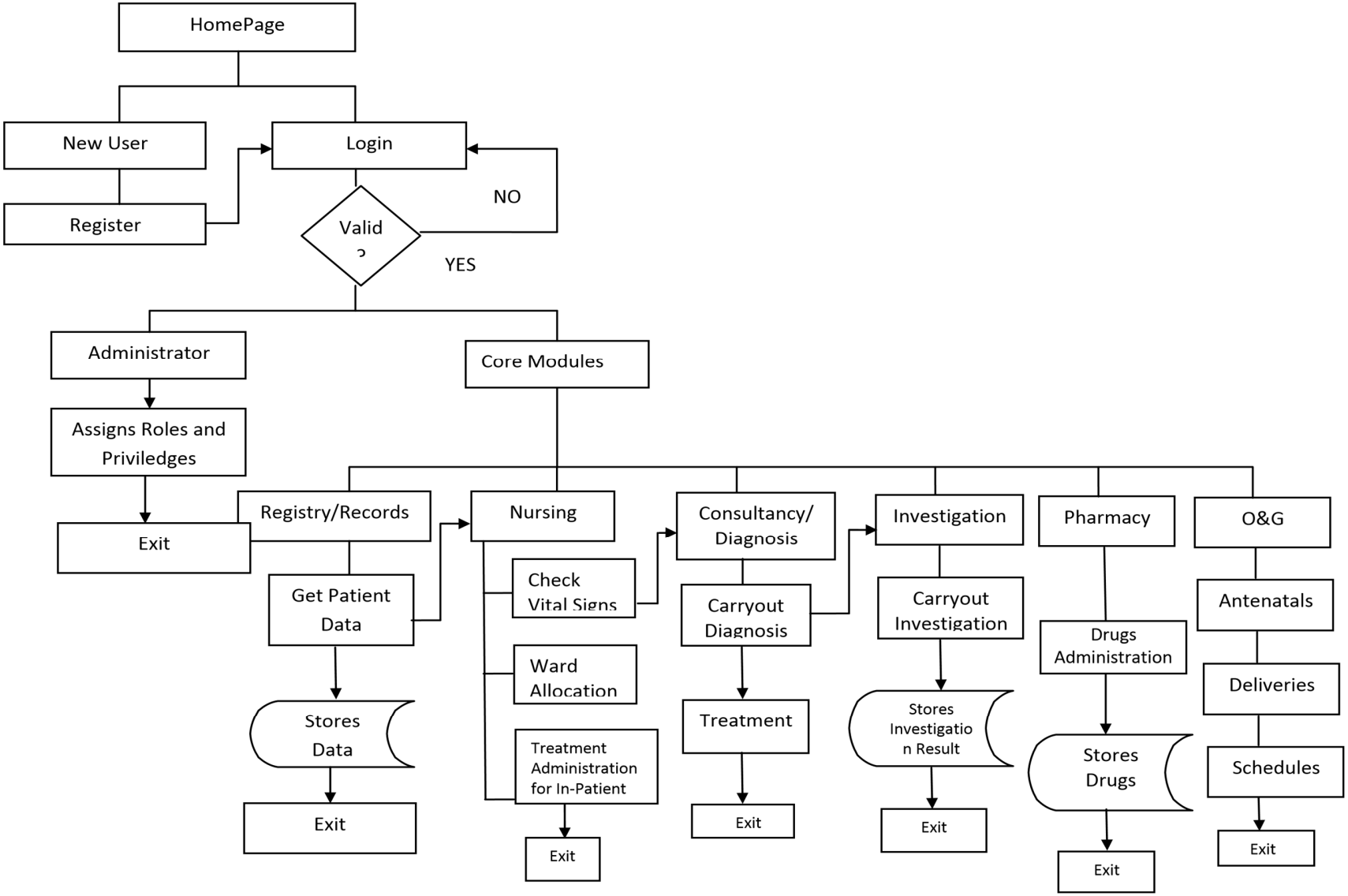
3. To make the system completely menu-driven and hence user-friendly. This is necessary to allow non-programmers use the system effectively and system could act as catalyst in achieving objective.

4. To ensure data integrity and security by protecting their data against non-authorized users or guiding against loss of patient‘s file or record.

5. This newly proposed system helps to eliminate swapping of patient‘s record and information.

6. To develop a reliable, understandable and cost effective system.

4.2 Main menu (control centre)



4.3 Database Specification

The database software to be used in handling the backend of this thesis is the SQL Server 2008 version. This solution uses the client/server architecture to access the records stored in the database. The database also provides security to the information stored there in by supporting techniques that grants access to the data examples are password tables, assigning roles and privileges to users etc. The database server handles only data retrieval and updates transactions, and does not participate in the applications interface in any way.

Data Dictionary

This contains all data definitions for cross-referencing and for managing and controlling access to the information repository / database. It provides a very thorough interface description (comparable to Interface Control Documents) that is independent of the model itself. Changes made to a model may be applied to the data dictionary to determine if the changes have affected the model‘s interface to other systems.

Data dictionaries do not contain any actual data from the database, only book keeping information for managing it. Without a data dictionary, however, a database management system cannot access data from the database. Below is the illustration of the data dictionary of the database:

Table 4.1: Data Dictionary of the database

S/N	VARIABLE NAME	VALUE	DESCRIPTION
1	Oid	uniqueidentifier	It uniquely identifies records
2	OptimisticLockField	Int	Handles concurrency in records
3	GCRecord	Int	Takes record of all deleted data
4	Description	Nvarchar(100)	

5	PatientName	uniqueidentifier	Name of the Patient
6	Doctor	uniqueidentifier	Name of the Doctor
7	Pharmacist	Nvarchar(100)	Name of the Pharmacist
8	Dosage	uniqueidentifier	Dosage of drug to be taken
9	Description	Nvarchar(100)	Uniquely identifies the client‘s record in the database
10	Amount	Nvarchar(100)	Amount the drug/service cost
11	Quantity	Nvarchar(100)	No. of items
12	TotalAmount	Nvarchar(100)	Sum total of drug/service cost
13	SheetCode	Int	Uniquely identifies sheet.
14	Notes	Nvarchar(100)	Doctor/Nurse/Pharm‘s notes
15	DrugCategory	Uniqueidentifier	Group the drug falls under
16	AnyPrevious	Nvarchar(100)	Any previous appointment
17	ClinicalDetails	Nvarchar(100)	Clinical notes
18	StateNo	Nvarchar(100)	No. assigned to a state
19	DateRecieved	Datetime2(7)	Date the drug was received
20	DateIssued	Datetime2(7)	Date drug was issued
22	NoOfPieces	Nvarchar(100)	No of pieces of the item
23	NoOfBlocks	Nvarchar(100)	No of Blocks of item
24	Type	Nvarchar(100)	Type of investigation
25	LabNo	Uniqueidentifier	Laboratory Number
26	LabScientist	Uniqueidentifier	Name of the Laboratory Staff
27	ChiefLabScientist	Uniqueidentifier	Name of the Head of Lab Staff
28	Diagnosis	Uniqueidentifier	Consultants Diagnosis
29	RequiredExam	Uniqueidentifier	Investigation Required
30	Quantity	nvarchar(100)	Quantity of Drugs
31	DiscountPercent	nvarchar(100)	Percentage Discounted
32	DiscountComputed	nvarchar(100)	Computed Discount

33	DateOfCollection	Datetime2(7)	Date the drugs were collected
34	BatchNo	nvarchar(100)	Drug batch number
35	NAFDACRegNo	nvarchar(100)	NAFDAC‘s registration number
36	ExpiryDate	Datetime2(7)	Expiry date of drug
37	CollectedBy	nvarchar(100)	Name of who collected the drugs
38	ManufactureDate	Datetime2(7)	Drug Manufacture Date
39	DirectionOfUse	nvarchar(100)	how the drug should be used
40	Constituents	nvarchar(100)	Constituents of the drug
41	StockDate	Datetime2(7)	Date stock was taken

Programs Identified In The Module

1. Registry/Records

This module is responsible for the assigning of unique number to and registration of InPatient(IP), Out-Patient(OP), Emergency-Patient and new born babies. It manages repeated visits, appointment scheduling. Keeps track of IP and OP previous and current medical history.

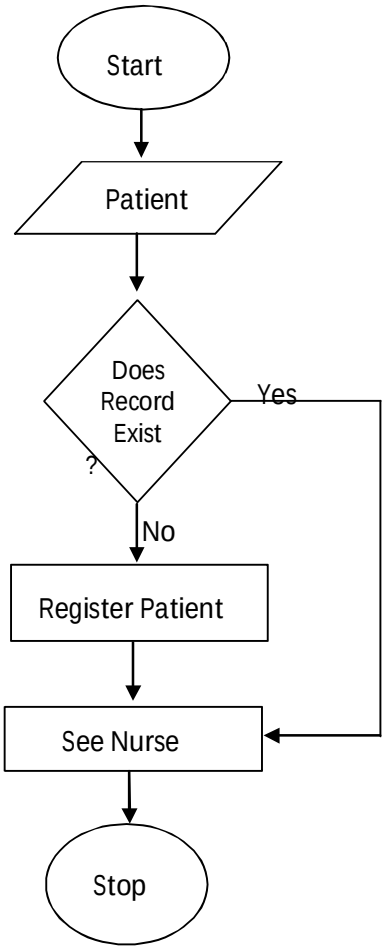


Fig 4.2 Registry/Records Flowchart

2. Nursing Module

In this module the capturing of patient‘s vital signs, fluid chart, dietary restrictions, admissions, discharges, transfers in/out. Also keeps track of prescription sheet and ward management.

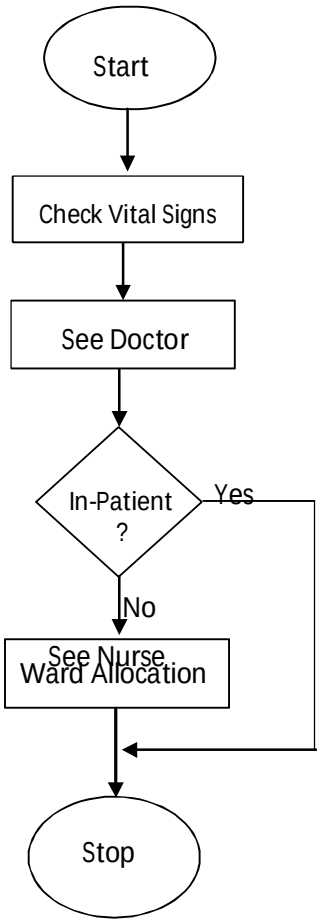


Fig 4.3 Nursing Flowchart

3. Consultancy and Diagnosis Module

This module is automates diagnosis, treatment analysis and prescription, places request for investigation and accesses investigation result.

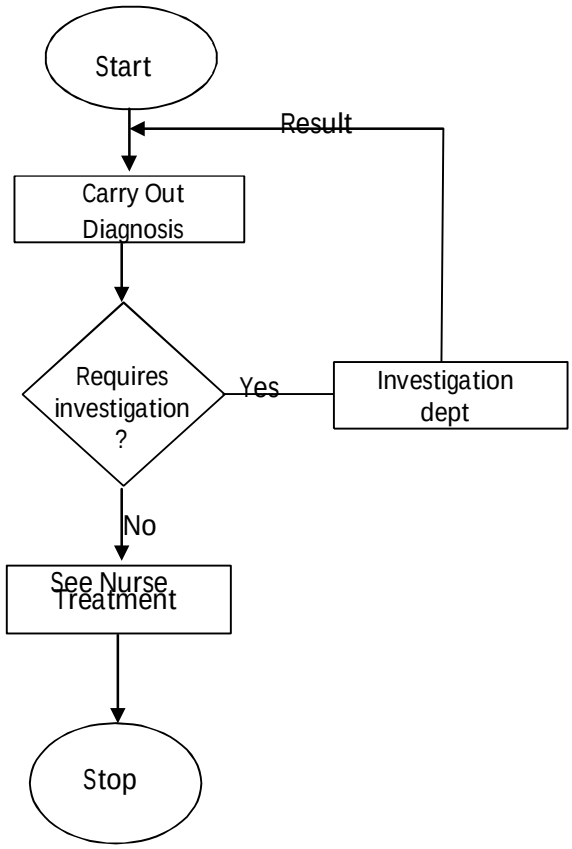


Fig 4.4: Consultancy Flowchart

4. Investigations

This module automates various investigations done in the hospital(e.g histopathology, chemical pathology, microbiology, haematology and radiology). Enable entry of test request from the any department and consultancy unit in free text format. Handles scheduling and automation of examinations for inpatients, casualties and outpatients. It also maintains the status and the results of various investigations done at the hospital.

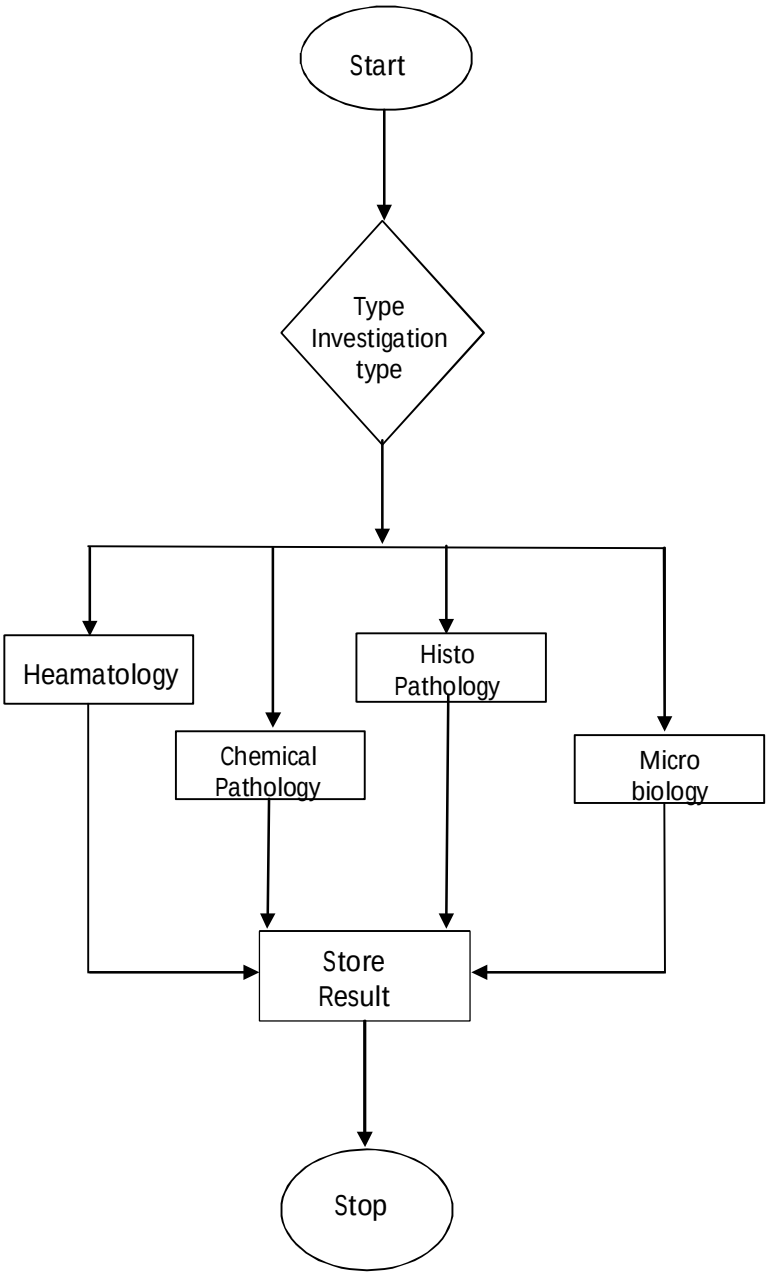


Fig 4.5: Investigation Flowchart

5. Pharmaceuticals

This module maintains details relating to all the pharmaceuticals and other general medical items available in the hospital. It captures entry of prescriptions and medication orders at both outpatients and ward levels, online updating of stock quantity, captures stock returns and automates selection of the earliest expiring batches.

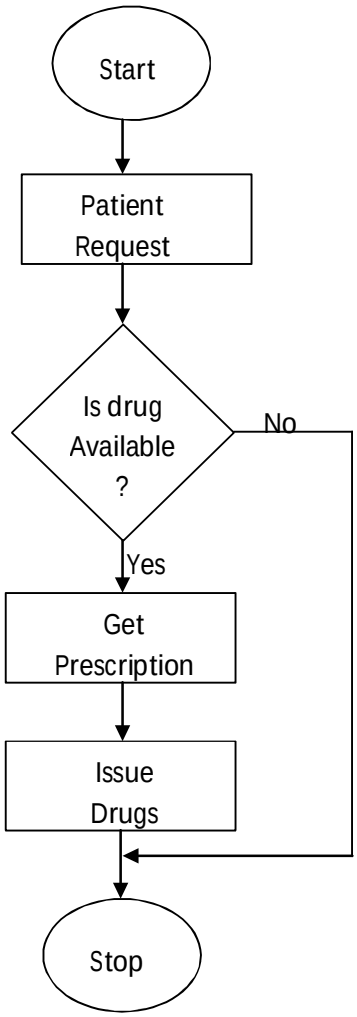


Fig 4.6: Pharmacy(Prescription) Flowchart

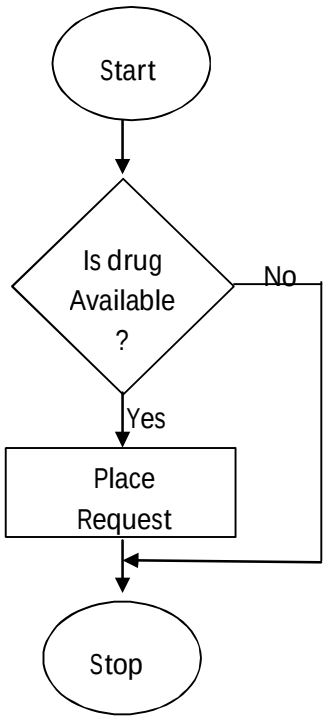


Fig 4.7: Pharmacy (Stock) Flowchart

4.4.2 Identifying information of the various modules in the database

Investigations

Table 4.2 Chemical Pathology data

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
Notes	nvarchar(100)	☑
SpecimenDate	nvarchar(100)	☑
SpecimenTime	nvarchar(100)	☑
RecievedDate	nvarchar(100)	☑
RecievedTime	nvarchar(100)	☑
SpecimenCondition	nvarchar(100)	☑
ClinicalDetails	nvarchar(100)	☑
Specimen	nvarchar(100)	☑
PatientName	uniqueidentifier	☑
Diagnosis	uniqueidentifier	☑
Consultant	uniqueidentifier	☑
Pathologist	uniqueidentifier	☑
LabScientist	uniqueidentifier	☑
ChiefLabScientist	uniqueidentifier	☑
LabNo	uniqueidentifier	☑
		☐

Table 4.3 Heamatology data

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
Notes	nvarchar(100)	☑
RecievedDate	nvarchar(100)	☑
ReportedDate	nvarchar(100)	☑
ReportedTime	nvarchar(100)	☑
SpecimenNature	nvarchar(100)	☑
SpeciemenCollection	nvarchar(100)	☑
SpeciemenCollectionDate	nvarchar(100)	☑
PatientName	uniqueidentifier	☑
LabNo	uniqueidentifier	☑
RequiredExamination	uniqueidentifier	☑
Diagnosis	uniqueidentifier	☑
Haematologist	uniqueidentifier	☑
LabScientist	uniqueidentifier	☑
ChiefLabScientist	uniqueidentifier	☑
		☐

Table 4.4 Histopathology data

Column Name	Data Type	Allow Nulls
DateIssued	nvarchar(100)	☑
NoOfPieces	nvarchar(100)	☑
NoOfBlocks	nvarchar(100)	☑
Type	nvarchar(100)	☑
PatientName	uniqueidentifier	☑
LabNo	uniqueidentifier	☑
LabScientist	uniqueidentifier	☑
ChiefLabScientist	uniqueidentifier	☑
Diagnosis	uniqueidentifier	☑
RequiredExam	uniqueidentifier	☑
Consultant	uniqueidentifier	☑
		☐

Table 4.5 Microbiology

Column Name	Data Type	Allow Nulls
Code	bigint	☐
EntityCode	nvarchar(50)	☑
EntityType	nvarchar(50)	☑
ReferenceCode	nvarchar(50)	☑
HospitalCode	nvarchar(50)	☑
BranchCode	nvarchar(50)	☑
Notes	nvarchar(MAX)	☑
UnitCode	bigint	☑
LabNo	bigint	☑
Type	nvarchar(50)	☑
Antibiogram	nvarchar(50)	☑
IsolateStage	nvarchar(50)	☑
LabScientist	nvarchar(50)	☑
Date	datetime2(7)	☑
ChiefLabScientist	nvarchar(50)	☑
MachineName	nvarchar(50)	☑
IpAddress	nvarchar(50)	☑
MachineUserName	nvarchar(150)	☑

Pharmaceticals

Table 4.6 Stocks data

Column Name	Data Type	Allow Nulls
Notes	nvarchar(100)	☑
CostPrice	nvarchar(100)	☑
Quantity	nvarchar(100)	☑
SellingPrice	nvarchar(100)	☑
DiscountedAmount	nvarchar(100)	☑
DiscountPercent	nvarchar(100)	☑
DiscountComputed	nvarchar(100)	☑
DateOfCollection	nvarchar(100)	☑
BatchNo	nvarchar(100)	☑
NAFDACRegNo	nvarchar(100)	☑
PharmacyCode	nvarchar(100)	☑
ExpiryDate	nvarchar(100)	☑
CollectedBy	nvarchar(100)	☑
ManufactureDate	nvarchar(100)	☑
DirectionOfUse	nvarchar(100)	☑
Constituents	nvarchar(100)	☑
StockDate	nvarchar(100)	☑
DrugName	uniqueidentifier	☑
DrugCategory	uniqueidentifier	☑
SupplierCode	uniqueidentifier	☑
Dosage	uniqueidentifier	☑

Table 4.7 Prescriptions data

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
Description	nvarchar(100)	☑
Amount	nvarchar(100)	☑
Quantity	nvarchar(100)	☑
TotalAmount	nvarchar(100)	☑
SheetCode	nvarchar(100)	☑
Notes	nvarchar(100)	☑
Pharmasist	nvarchar(100)	☑
PatientName	uniqueidentifier	☑
DrugCategory	uniqueidentifier	☑
Pharmacist	uniqueidentifier	☑
Dosage	uniqueidentifier	☑
Doctor	uniqueidentifier	☑
Duration	uniqueidentifier	☑
		☐

Registry and Records Table 4.8 Patients

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
FirstName	nvarchar(100)	☑
MiddleName	nvarchar(100)	☑
LastName	nvarchar(100)	☑
Gender	uniqueidentifier	☑
OptimisticLockField	int	☑
GCRecord	int	☑
OtherNames	nvarchar(100)	☑
MaritalStatus	uniqueidentifier	☑
Notes	nvarchar(100)	☑
Tribe	uniqueidentifier	☑
Occupation	uniqueidentifier	☑
Profession	uniqueidentifier	☑
Religion	uniqueidentifier	☑
Photo	varbinary(MAX)	☑
		☐

Table 4.9 Addresses

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
Notes	nvarchar(100)	☑
AddressType	uniqueidentifier	☑
City	nvarchar(100)	☑
Country	uniqueidentifier	☑
State	uniqueidentifier	☑
LGA	uniqueidentifier	☑
Town	nvarchar(100)	☑
StreetNumber	nvarchar(100)	☑
OptimisticLockField	int	☑
GCRecord	int	☑
Patients	uniqueidentifier	☑
		☐

Consultancy

Table 4.10 Clinical Notes

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
Patient	uniqueidentifier	☑
Title	nvarchar(100)	☑
Notes	nvarchar(100)	☑
Date	datetime	☑
		☐

Table 4.11 Discharges

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
Patient	uniqueidentifier	☑
Witness	nvarchar(100)	☑
Date	datetime	☑
Notes	nvarchar(100)	☑
DischargeType	uniqueidentifier	☑
DischargeNurse	uniqueidentifier	☑
DischargeDoctor	uniqueidentifier	☑
		☐

Nursing

Table 4.12 Treatments

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
Notes	nvarchar(100)	☑
DrugType	nvarchar(100)	☑
DrugDescription	nvarchar(100)	☑
Dosage	nvarchar(100)	☑
Route	nvarchar(100)	☑
Instructions	nvarchar(100)	☑
Physician	nvarchar(100)	☑
Instruction	nvarchar(100)	☑
Pharmacist	nvarchar(100)	☑
AttendantNurse	nvarchar(100)	☑
Frequency	nvarchar(100)	☑
Duration	nvarchar(100)	☑
PatientName	nvarchar(100)	☑
		☐

Table 4.13 Vital Signs

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
Type	nvarchar(100)	☑
DateVisited	datetime	☑
Notes	nvarchar(100)	☑
PatientName	uniqueidentifier	☑
ConsultantNurse	uniqueidentifier	☑
		☐

Table 4.14 Admissions

Column Name	Data Type	Allow Nulls
Code	bigint	☐
EntityCode	nvarchar(50)	☑
EntityType	nvarchar(50)	☑
EntityCategory	nvarchar(50)	☑
ReferenceCode	nvarchar(50)	☑
HospitalCode	nvarchar(50)	☑
BranchCode	nvarchar(50)	☑
UnitCode	bigint	☑
AttendantNurse	nvarchar(100)	☑
Physician	nvarchar(100)	☑
BedNo	nvarchar(50)	☑
Notes	nvarchar(MAX)	☑
Date	datetime2(7)	☑
Type	nvarchar(50)	☑
ScreenCode	nvarchar(50)	☑
ShiftCode	nvarchar(50)	☑
MachineName	nvarchar(50)	☑
IpAddress	nvarchar(50)	☑
MachineUserName	nvarchar(150)	☑

Obstetrics and Gynaecology

Table 4.15 Antenatals

Column Name	Data Type	Allow Nulls
Code	bigint	☐
EntityCode	nvarchar(50)	☑
EntityType	nvarchar(50)	☑
EntityCategory	nvarchar(50)	☑
ReferenceCode	nvarchar(50)	☑
HospitalCode	nvarchar(50)	☑
BranchCode	nvarchar(50)	☑
UnitCode	bigint	☑
AttendantNurse	nvarchar(100)	☑
Physician	nvarchar(100)	☑
BedNo	nvarchar(50)	☑
Notes	nvarchar(MAX)	☑
Date	datetime2(7)	☑
Type	nvarchar(50)	☑
ScreenCode	nvarchar(50)	☑
ShiftCode	nvarchar(50)	☑
MachineName	nvarchar(50)	☑
IpAddress	nvarchar(50)	☑
MachineUserName	nvarchar(150)	☑

Table 4.16 Deliveries

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
BloodLoss	nvarchar(100)	☑
Physician	nvarchar(100)	☑
Booked	nvarchar(100)	☑
WhereBooked	nvarchar(100)	☑
Parity	nvarchar(100)	☑
ChildrenAlive	nvarchar(100)	☑
ChildrenDead	nvarchar(100)	☑
PregnancyAge	nvarchar(100)	☑
RuptureOfMembrane	nvarchar(100)	☑
DiagnosisCode	nvarchar(100)	☑
DeliveryTime	nvarchar(100)	☑
DeliveryType	nvarchar(100)	☑
ChildSex	nvarchar(100)	☑
CircumferenceHead	nvarchar(100)	☑
BirthLength	nvarchar(100)	☑
BirthWeight	nvarchar(100)	☑
PerinumState	nvarchar(100)	☑

Table 4.17 Schedules

Column Name	Data Type	Allow Nulls
Oid	uniqueidentifier	☐
OptimisticLockField	int	☑
GCRecord	int	☑
FondusHeight	nvarchar(100)	☑
PresentationAndPosition...	nvarchar(100)	☑
FoetalHeart	nvarchar(100)	☑
Oedema	nvarchar(100)	☑
Urine	nvarchar(100)	☑
Weight	nvarchar(100)	☑
Heamoglobin	nvarchar(100)	☑
BloodPressure	nvarchar(100)	☑
TreatmentCode	nvarchar(100)	☑
DateVisited	datetime	☑
Notes	nvarchar(100)	☑
PatientName	nvarchar(100)	☑
		☐

4.5 Program Structure

1. Registry/Records

This module is responsible for the assigning of unique number to and registration of InPatient(IP), Out-Patient(OP), Emergency-Patient and new born babies. It manages repeated visits, appointment scheduling. Keeps track of IP and OP previous and current medical history.

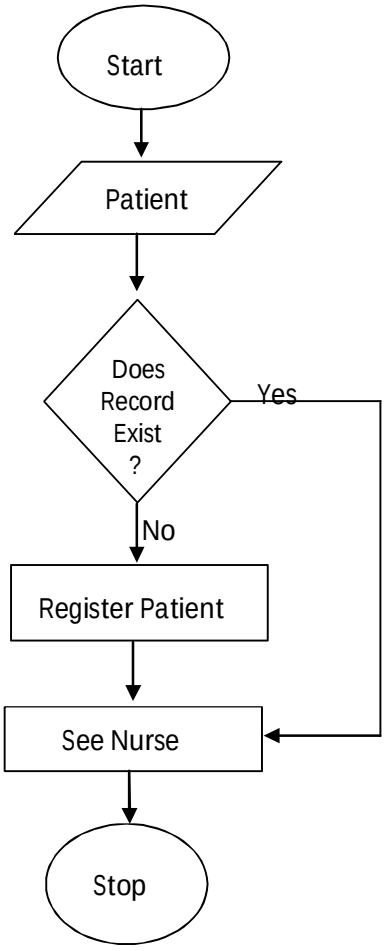
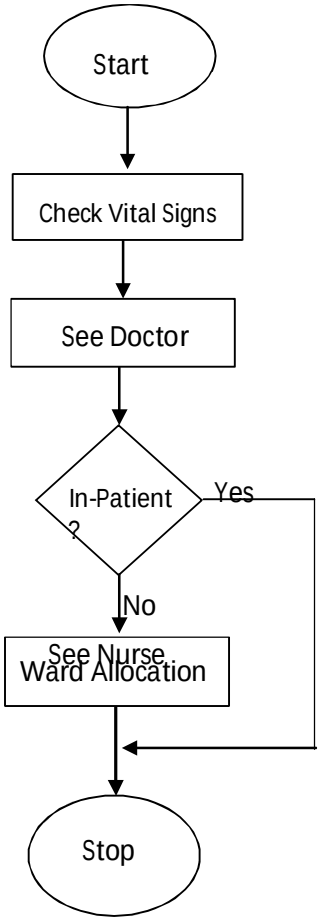


Fig 4.8 Registry/Records Flowchart

2. Nursing Module

In this module the capturing of patient‘s vital signs, fluid chart, dietary restrictions, admissions, discharges, transfers in/out. Also keeps track of prescription sheet and ward management.

Fig 4.9 Nursing Flowchart



3. Consultancy and Diagnosis Module

This module is automates diagnosis, treatment analysis and prescription, places request for investigation and accesses investigation result.

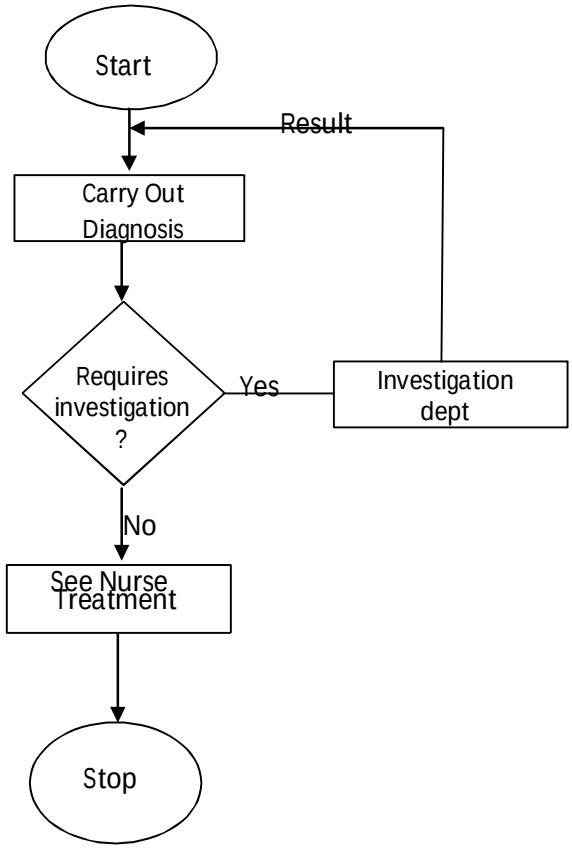


Fig 4.10: Consultancy Flowchart

4. Investigations

This module automates various investigations done in the hospital(e.g histopathology, chemical pathology, microbiology, haematology and radiology). Enable entry of test request from the any department and consultancy unit in free text format. Handles scheduling and automation of examinations for inpatients, casualties and outpatients. It also maintains the status and the results of various investigations done at the hospital.

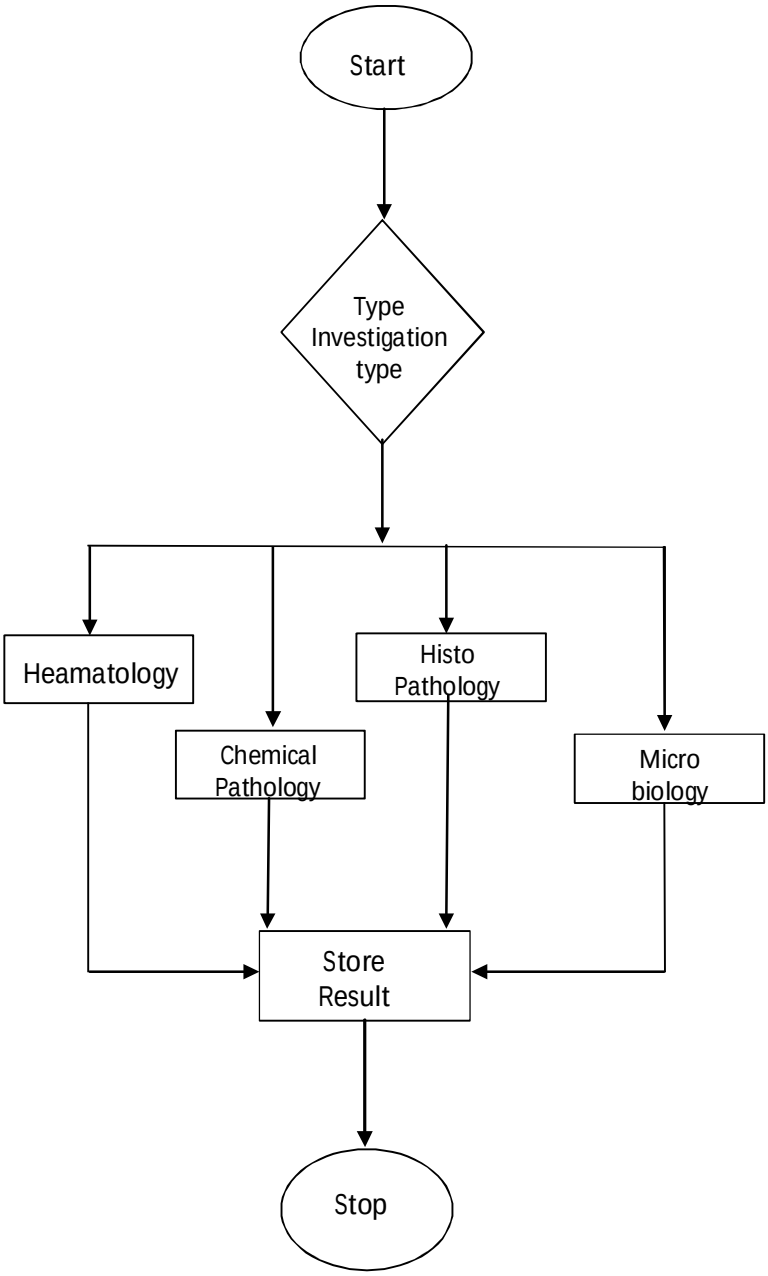


Fig 4.11: Investigation Flowchart

5. Pharmaceuticals

This module maintains details relating to all the pharmaceuticals and other general medical items available in the hospital. It captures entry of prescriptions and medication orders at both outpatients and ward levels, online updating of stock quantity, captures stock returns and automates selection of the earliest expiring batches.

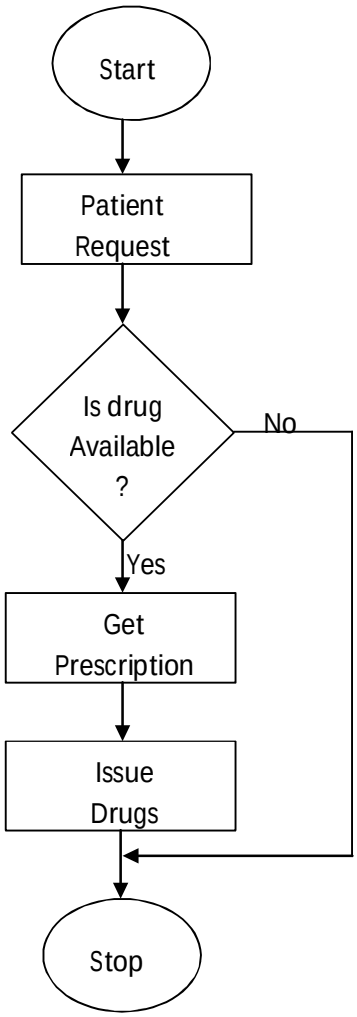


Fig 4.12: Pharmacy(Prescription) Flowchart

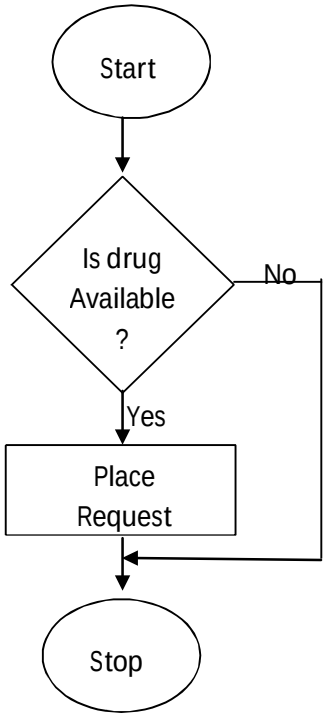


Fig 4.13: Pharmacy (Stock) Flowchart

4.6 Input and Output Format

Investigations Histopathology/Cytology Form (Existing Form)

(HISTOPATHOLOGY/CYTOLOGY)

HISTOPATHOLOGY/CYTOLOGY NO……

HOSPITAL NO…….....

SURNAME OTHER NAMES AGE SEX TRIBE:

NATIONALITY:

PROVISION DIAGNOSIS…….EXAM REQUIRED……

CLINICAL

DETAILS…….....……

……

……

ANY PREVIOUS BIOPSY/CYTOLOGY YES/NO

IF YES, STATE NUMBER……

FOR LABORATORY USE ONLY

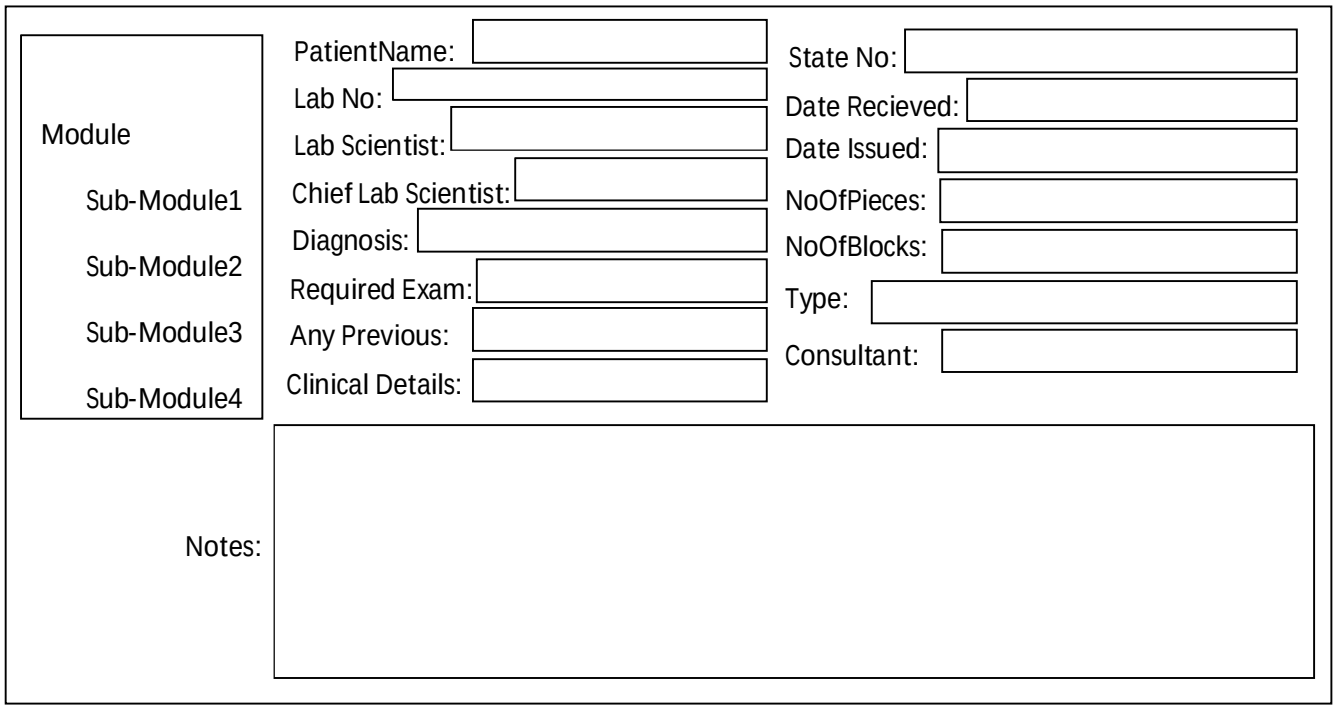
DATE RECEIVED…… NO. OF PIECES…… NO. OF BLOCKS…….....

DATE ISSUED…….

NAME OF DOCTOR(NOT SIGNATURE PLS)

Request for frozen section should be submitted to the department of morbid anatomy 24 hrs before surgery

Table 4.18: Histopathology (Sample Output)



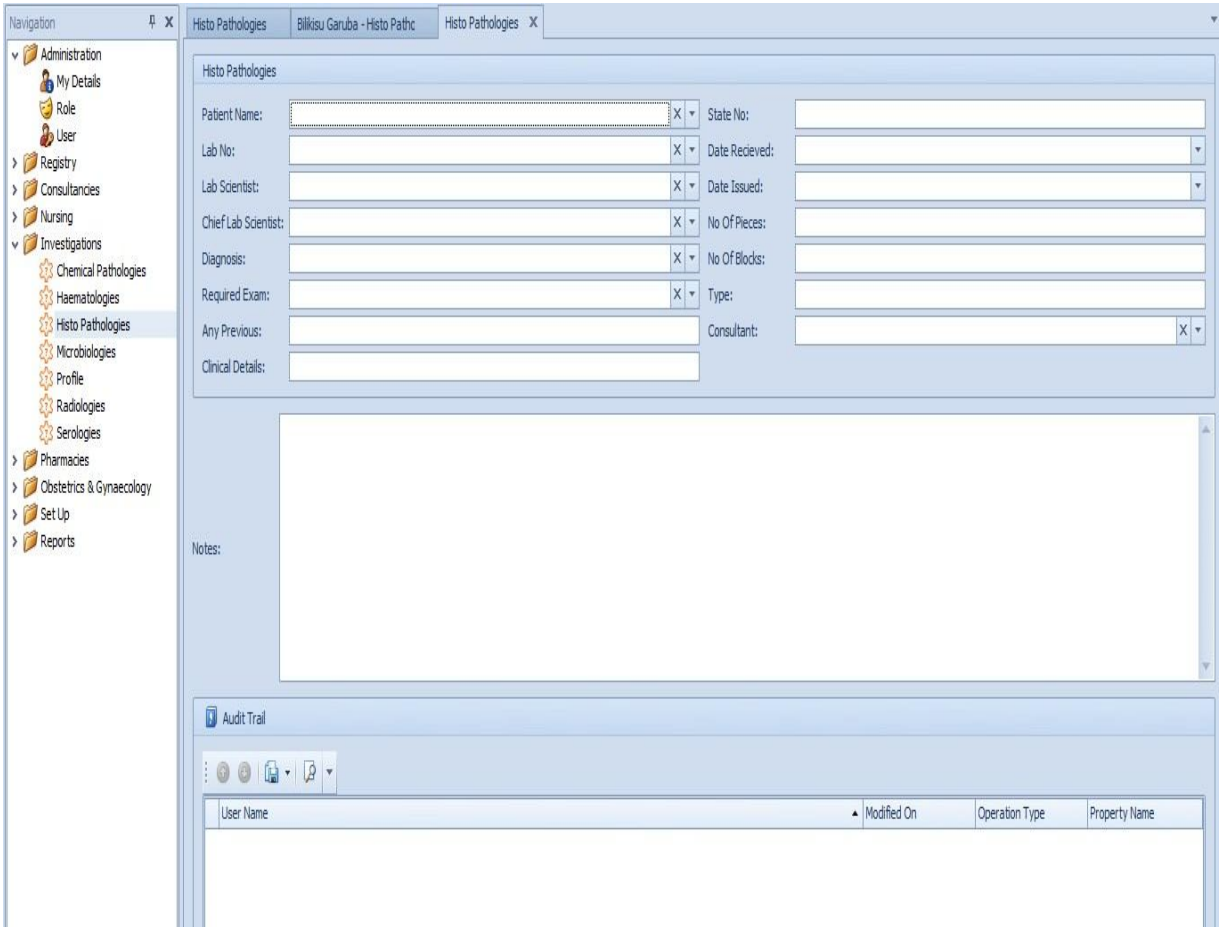


Fig 4.14: Proposed Automated Histopathology Form Table 4.19: LABORATORY REPORT FORM - MICROBIOLOGY

NAME	AGE	SEX	CLINIC	HOSPITAL NO.	LAB. NO
REPORT	ANTIBIOGRAM		ISOLATES		
			1	2	3
Macroscopy/Appearance	Penicillin				
	Ampicillin				
	Streptomycin				
	Chloramphenical				
	Tetracycline				
	Erythromycin				
	Septrin				
	Cloxacillin				

	Sulponamide			
	Naliddixic Acid			
Microscopy/Gram Staining	Ntrofurantoin			
	Collstin S.			
	Genticin			
	Pefloxacine			
	Augumentine			
Culture	Amoxyllin			
	Cerftriazone			
	Ofloxacin			

Other Drugs(Specify)

S = Sensitivity

R = Resistance **MED. LAB SCIENTIST** **DATE**

Table 4.20: Microbiology Sample Output

Module

Sub-Module1

Sub-Module2

Sub-Module3

Sub-Module4

Unit

Lab No:

Type:

Lab

Date:

Code: Diagnosis:

Date Issued:

Scientist:

Antibiogram:

Notes:

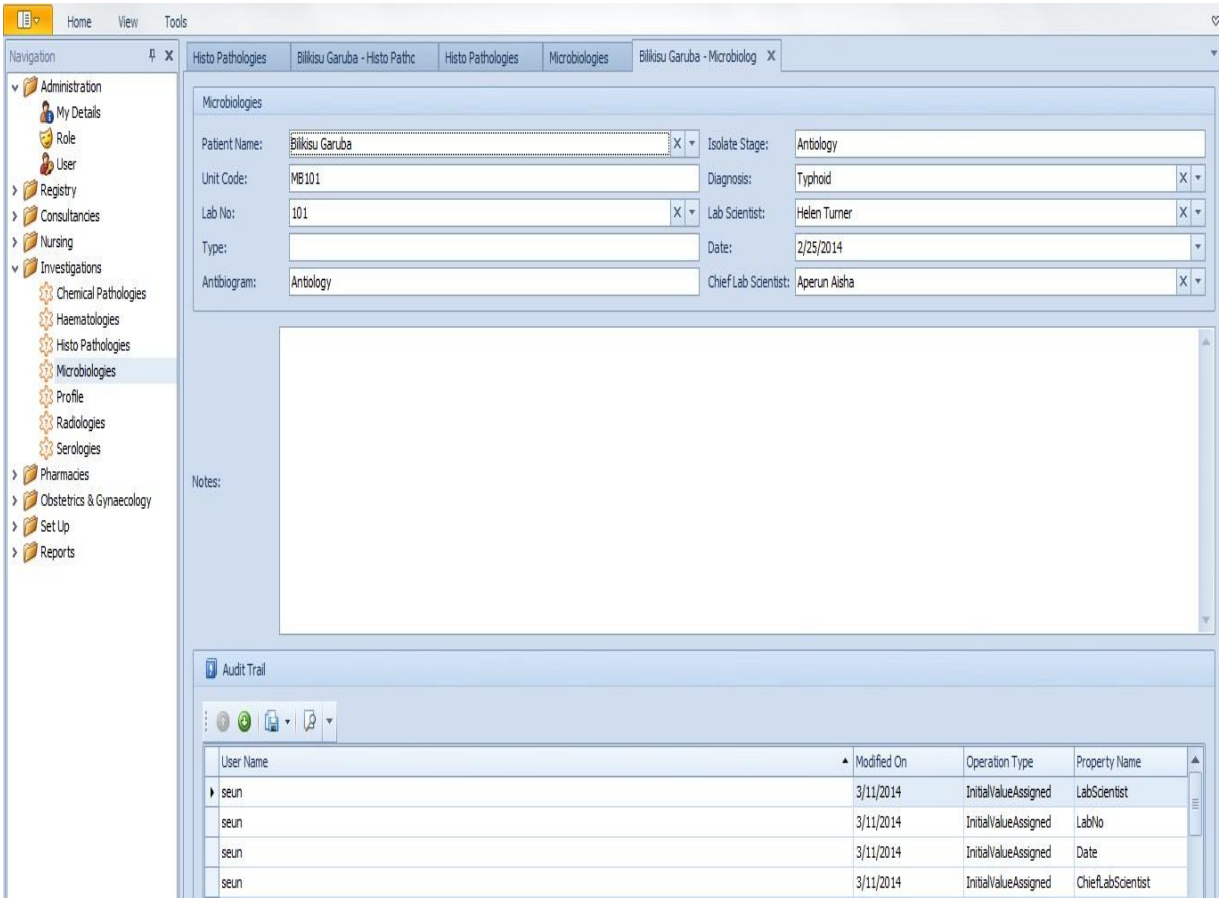


Fig 4.15 Proposed Automated Microbiology Form

CHEMICAL PATHOLOGY REQUEST/REPORT FORM(**Existing System**)

SURNAME	OTHERNAMES		TICK HERE	BLOOD CONT.	RESULT
TIME & DATE OF SPECIMEN COLLECTION		AGE		Sodium (120-140)	Mmol/l
				Lithium (<1)	
HOSPITAL	SPECIMEN	SEX		ENZYMES	
				Amylase (70-300)	I.U/L
WARD/CLINIC	CONSULTANT			Alkaline Phosphatase	
LAB NO.				Gamma GTP (4-28)	
DATE & TIME RECEIVED				HB DH (55-140)	
CONDITION OF SPECIMEN				CPK (up to 50)	
CLINICAL DETAILS				SGOT(up to 18)	
				SGPT (up to 22)	
				Fast Glucose (2.8-5.5)	
				2hpp Glucose	

TICK HERE	BLOOD	RESULT	TICK HERE	BLOOD CONT.	RESULT
	Bilirubin Total (up to 20)	Umol/l		Calcium (2.25-2.75)	
	Conj. Bilirubin (up to 5)	Umol/		Ing.Phosphase (0.65-1.3)	
	LIPID PROFILE			Total Protein (58-80)	
	Total Cholesterol (2.5-6.5)			Albumin (35-50)	
	Trigiyceride (0.45-1.72)	Mmol/l		Globulin (20-45)	
	LDL-C (1.68-4.35)	Mmol/l		Creatinine(50-132)	
	HDL-C(0.78-2-20)	Mmol/l		Urea (2.5-5.8)	
	ELECTROLYTES			Uric Acid (0.12-0.36)	
	Bicarbonate(20-30)	Mmol/l		URINE (Spot/124hrs)	
	Chloride (95-110)	Mmol/l		Glucose	
	Potassium (3-5)	Mmol/l		Protein	
	GTT (Oral/Parental) Fasting …… 1hr…… 2hr……	1/2hr…… 1 1/2hrs…… 2 1/2hrs……		Bilirubin	
				Urobilinogen	
				Ketones	
				pH	
				Emu (Preg Test)	

CONSULTANT CHEM. PATHOLOGIST'S COMMENT		CSF	
		Glucose (2.7-3.9)	
		Protein (15-45)	
		FAECES	
		Occult blood	
		Faecal Fat	
Signature CONSULTANT CHEM.PATHOLOGIST	MED. LAB SCIENTIST SIGNATURE:		CHIEF MED. LAB SCIENTIST SIGNATURE:

HAEMATOLOGY REQUEST/REPORT FORM (**Existing System**)

SURNAME	FIRST NAME(S)	Unit No._____ GP_____ CAS_____			Lab No.	Date	Time
					Received		
WARD/CLINIC	PHYSICIAN/SURGEON	Age	Sex	State	Reported		

Nature of Specimen (Tick appropriate box)

Finger prick — Blood in sequestrene

Clotted blood — Blood in heparin

Blood in ACD — Defibrinated blood

Marrow — Other (Specify)

Clinical Details _____

_____

_____

Examination(s) Required _____

_____

_____

Specimen Collected:

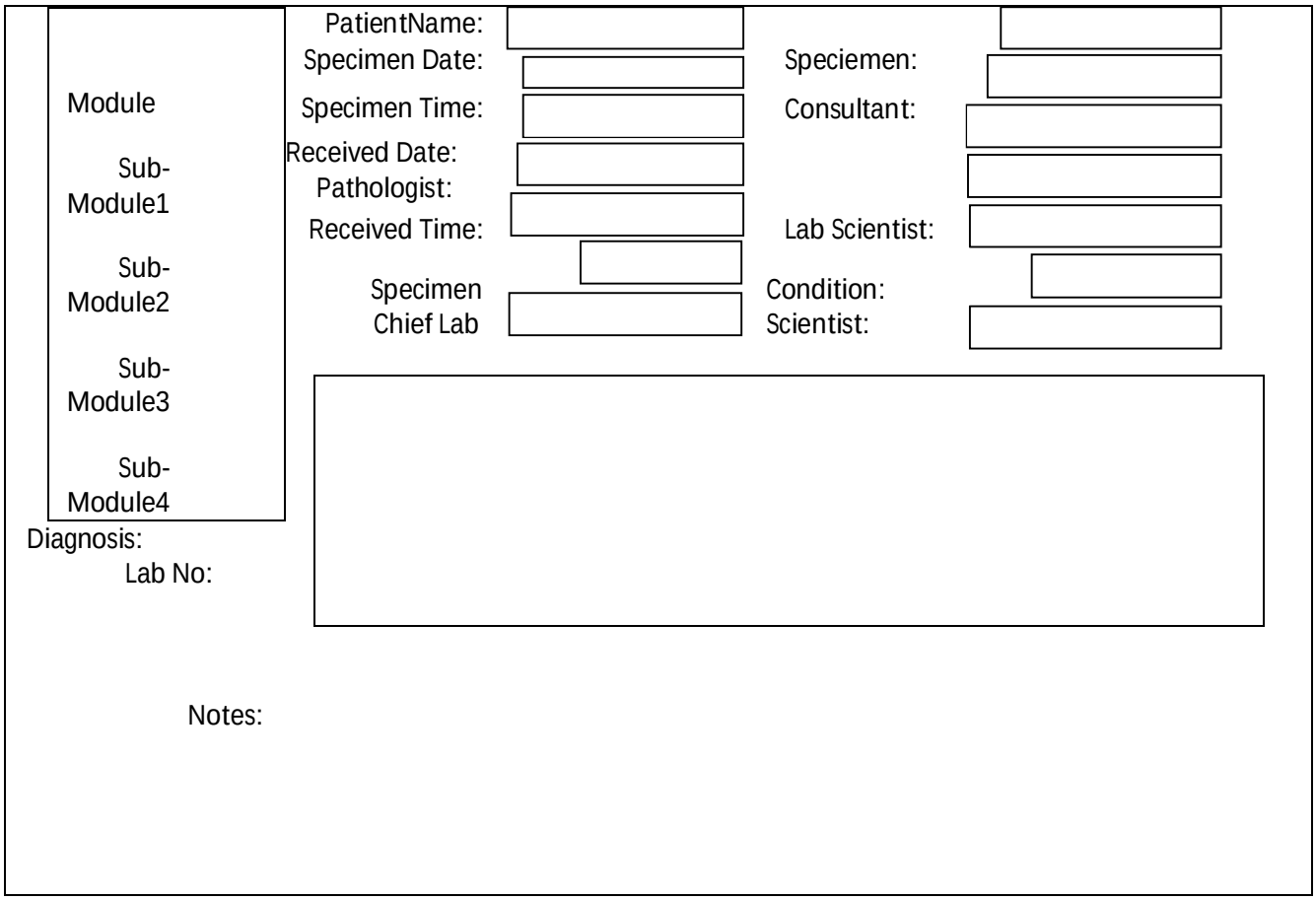
Date: Time:

PCV_____% Hb _____ G/100ml
MCHC _____% WBC _____ /cmm
Retics _____% ESR _____ mm/hr
Retics Index _____% Eosinophils_____/cmm
Platelets_____/cmm

FILM APPEARANCE		DIFFERENT COUNT (%)	
Aniscocytosis		Blast	
Poikilocytosis		Myel	
Polychromasia		Meta Myel	
Macrocytosis		Eosin Myel	
Hypochromia		Eosin	
Sickel Cells		Baso	
Target Cells		Neut	
Spherocytes		Prim Lymph	
Nucleated RBC		Lymph	
Microcytosis		Prim Mono	
Others		Mono	
		Other	

	MED. LAB. SCIENTIST:			
	CHIEF MED. LAB SCIENTIST			

Table 4.23: Chemical Pathology (Sample OutPut)



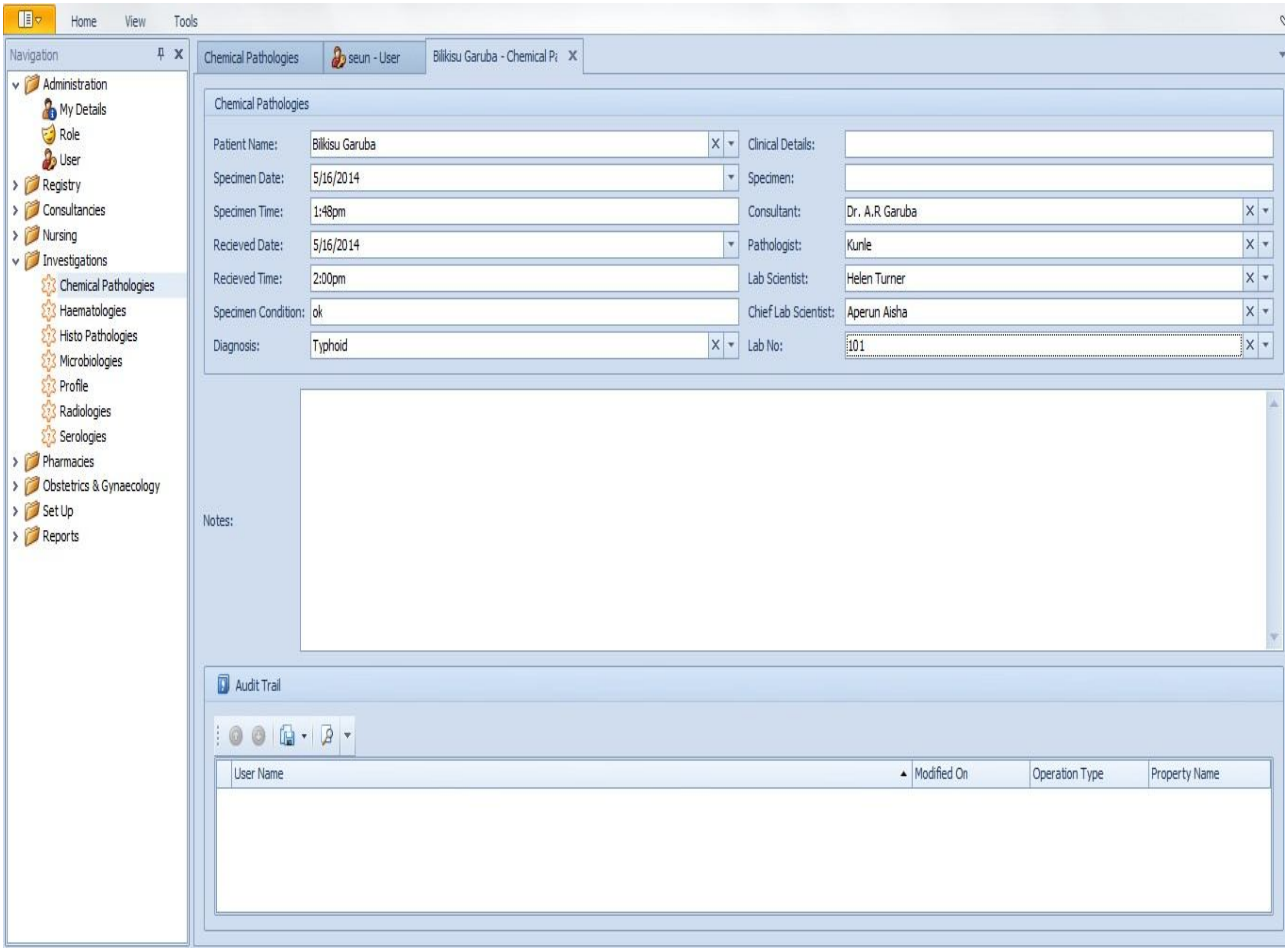
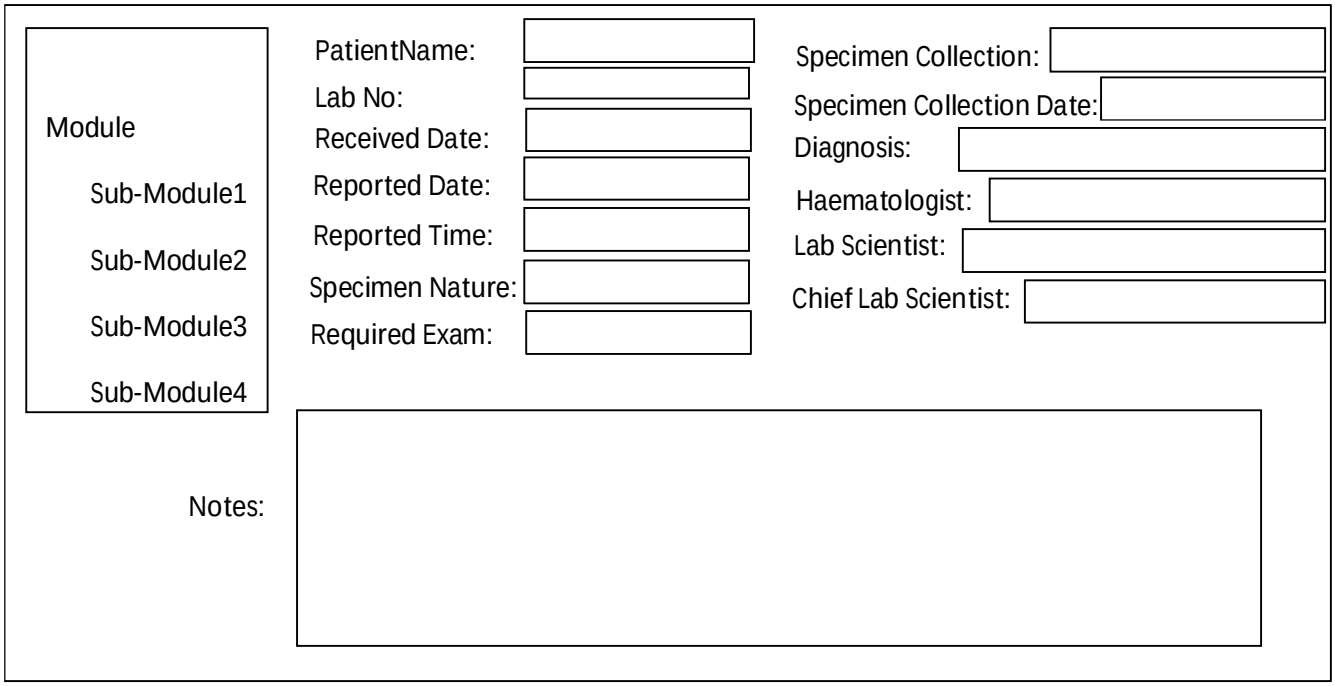


Fig 4.16:Proposed Automated Chemical Pathology Form

Fig 4.24: Hematology (Sample Output)



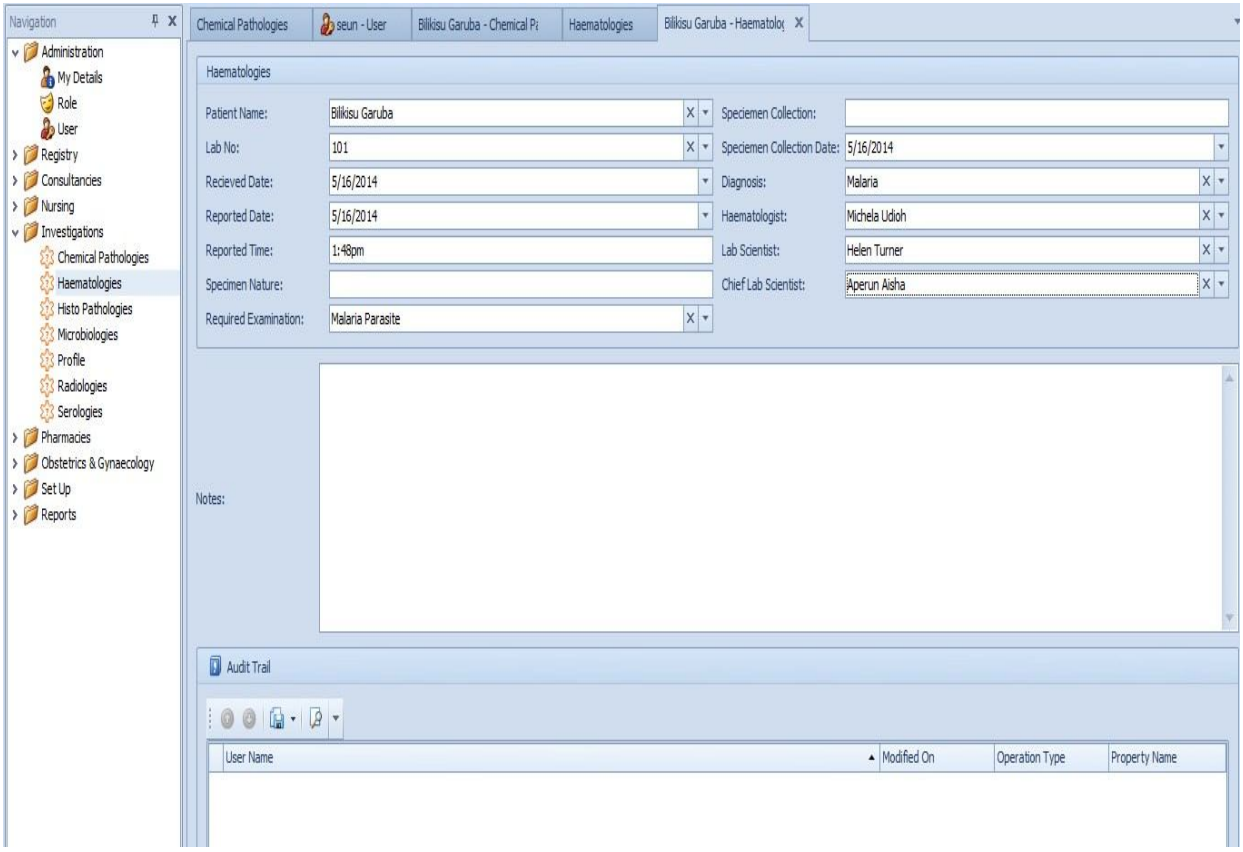


Fig 4.17: Proposed Automated Heamatology Form

Nursing Care/Treatment FORM (Existing Form)

NURSING PROCESS/CARE PLAN SHEET

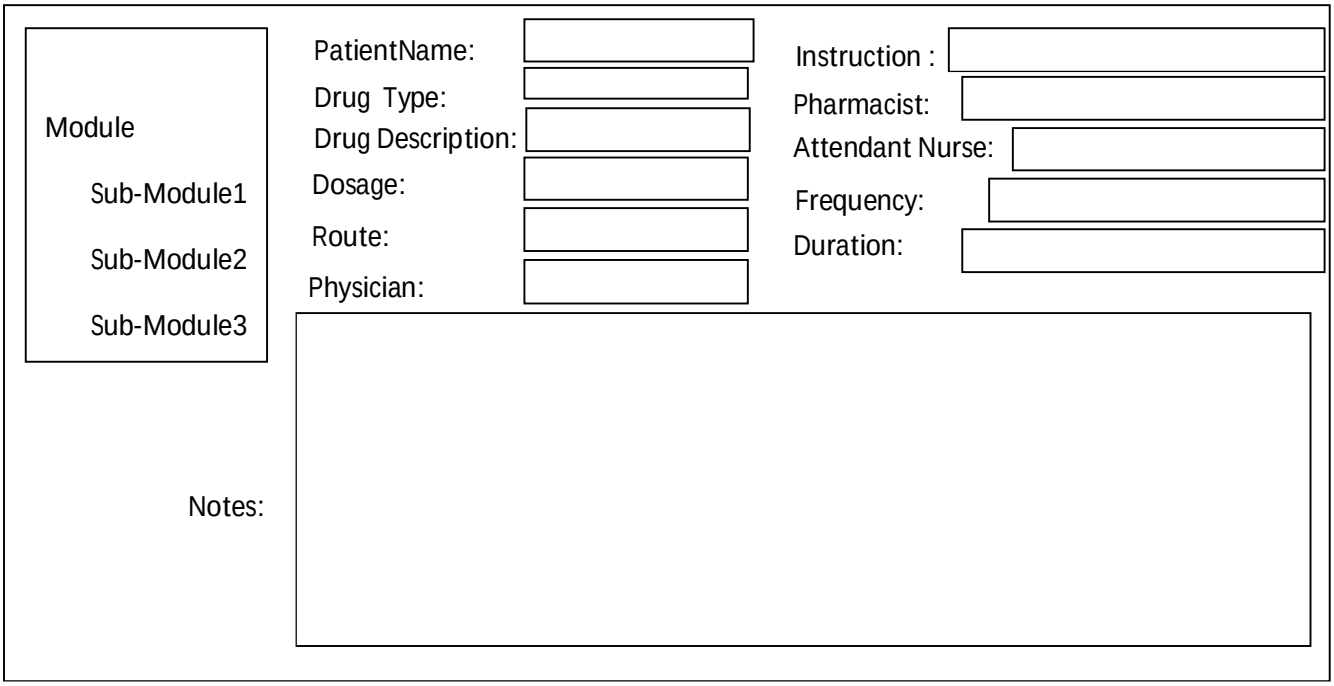
Name:……

Address:…… Ward:……

Age:……
Occupation:……
Medical Diagnosis:…… Case
Note No:……

DATE/TIME	NURSING DIAGNOSIS	NURSING OBJECTIVE	NURSING ACTIONS AND INTERVENTIONS	SCIENTIFIC RATIONALE	EVALUATION	SIGNATURE

Table 4.25: Nursing care/Treatment (Sample output)



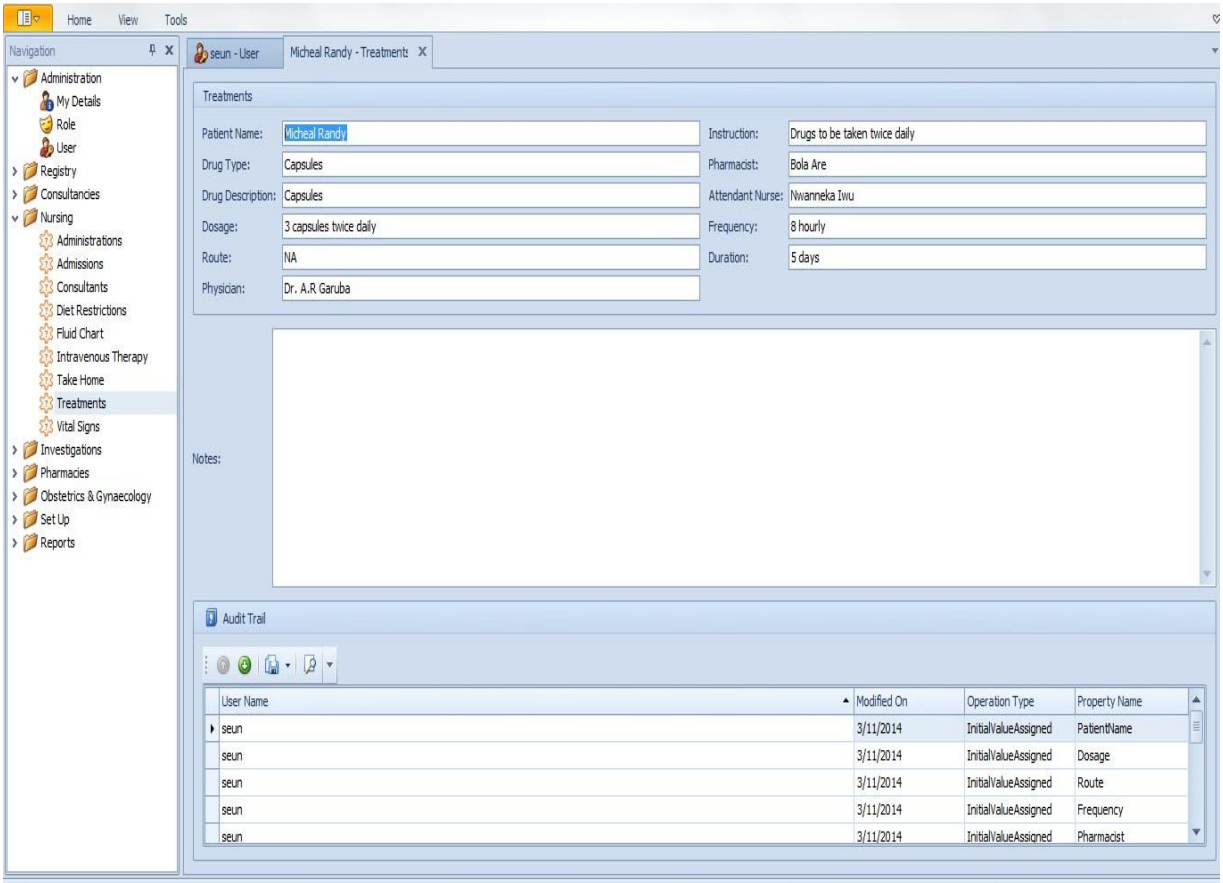


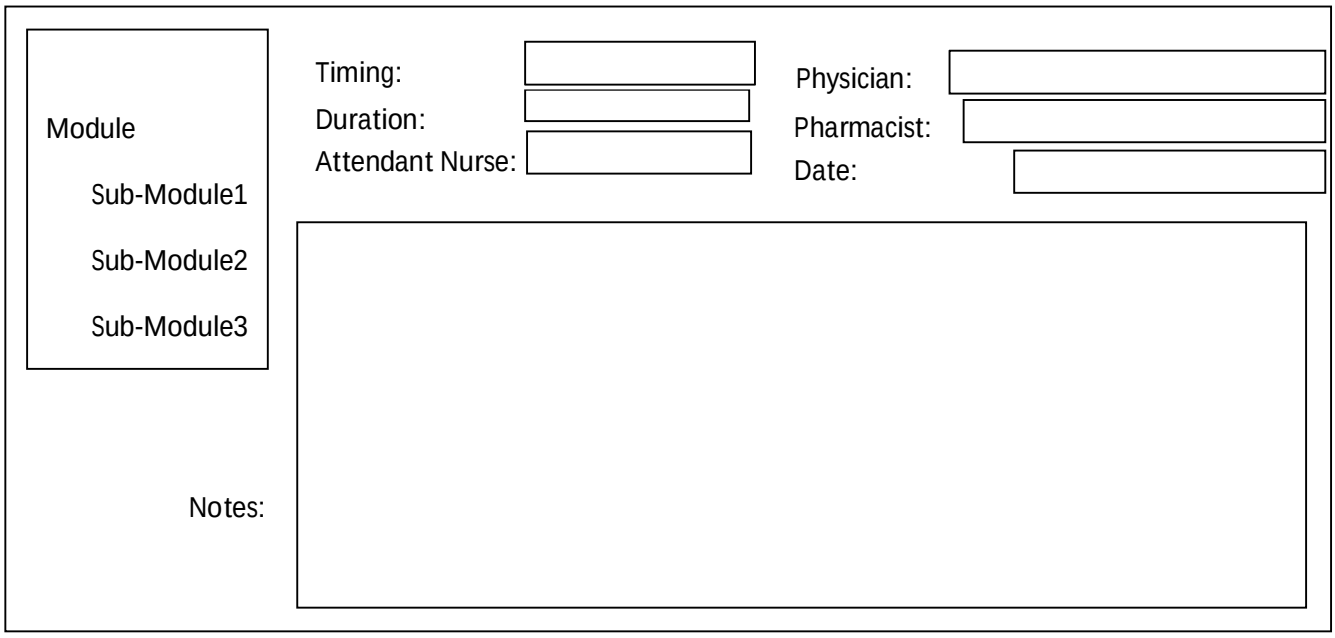
Fig 4.18: Proposed Automated Nursing Treatment Form
Nursing Administrations FORM (Existing Form)

NURSES NOTES

SURNAME	FIRSTNAME	ATTENDING PHYSICIAN/SURGEON	HOSPITAL NO

DATE	TIME	REMARKS-MEDICATIONS	NAME	SIGNATURE

Table 4.26: Nursing Administration (Sample Output)



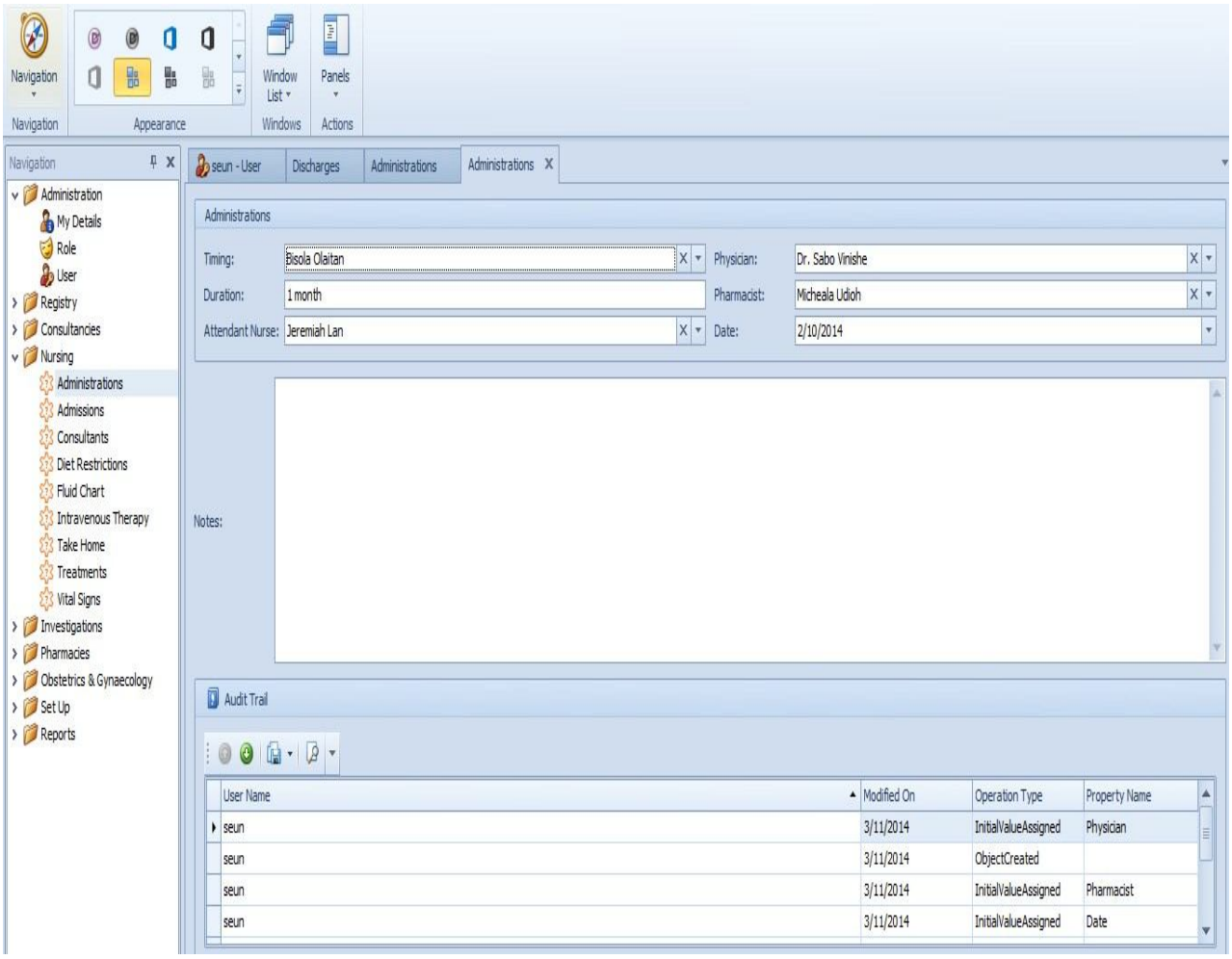


Fig 4.19: Proposed Automated Nursing Administration Form.

Pharmacy Stocks FORM (Existing Form)

PHARMACY STOCKS FORM

SIV NO:……DATE:……

No.	*Description*	*Unit of Issue*	*Quantity*		*Price per unit*	*Total amount*	*Ledger Amount*	*Remarks*
			Required	*Supplied*				

Signature of Pharmacist(Collector):……...…Approved by:…….

Table 4.27: Pharmacy Stocks (Sample Output)

Module

Sub-Module1

Sub-Module2

Drug Name:

Drug Category:

Cost Price:

Quantity:

Selling Price:

Discounted Amount:

Discount Percent:

Discount Computed:

Date Of Collection:

Batch No:

NAFDACReg No:

Pharmacy Code :

Expiry Date:

Collected By:

Manufacturer Date:

Dosage:

Direction Of Use:

Status:

Constituents:

Stock Date:

Notes:

Rank:…….....Rank:…….

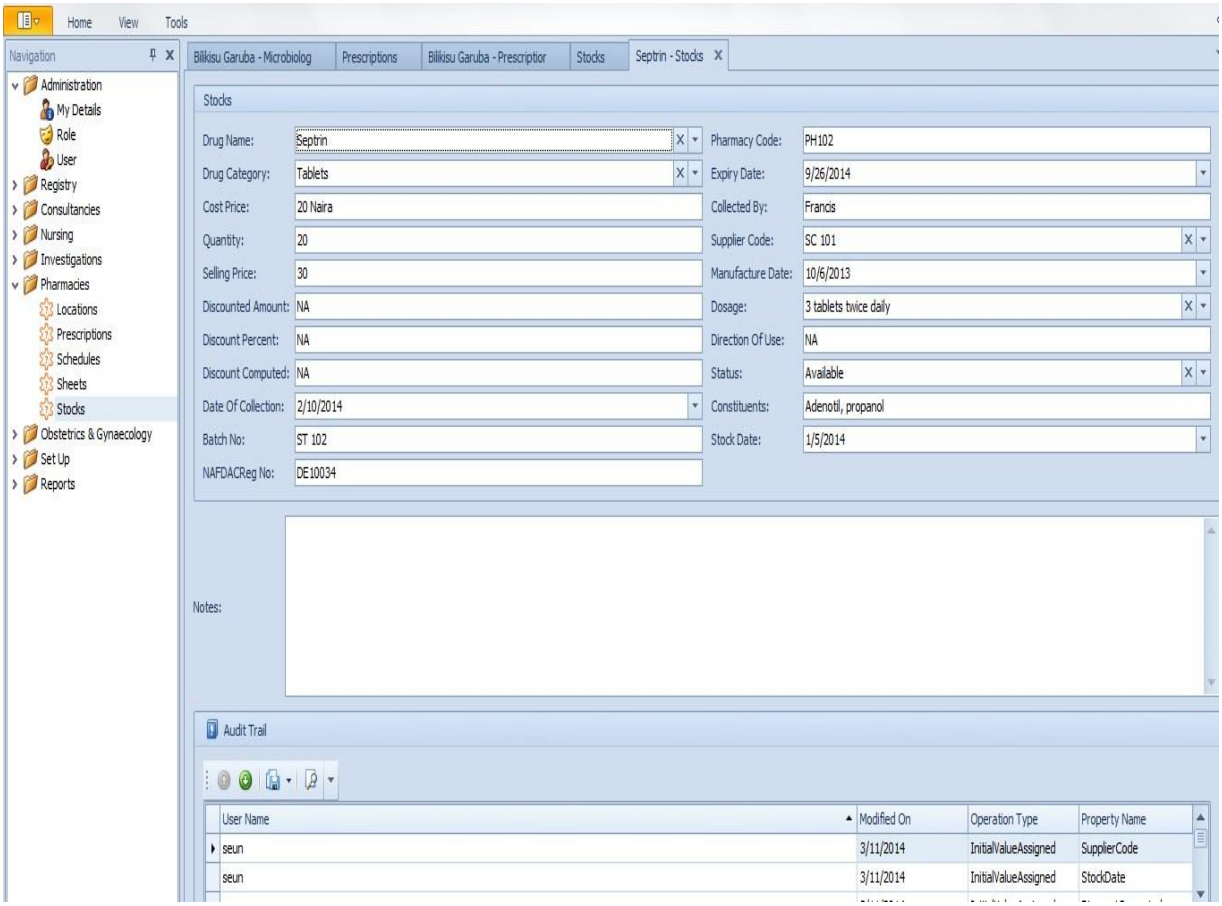
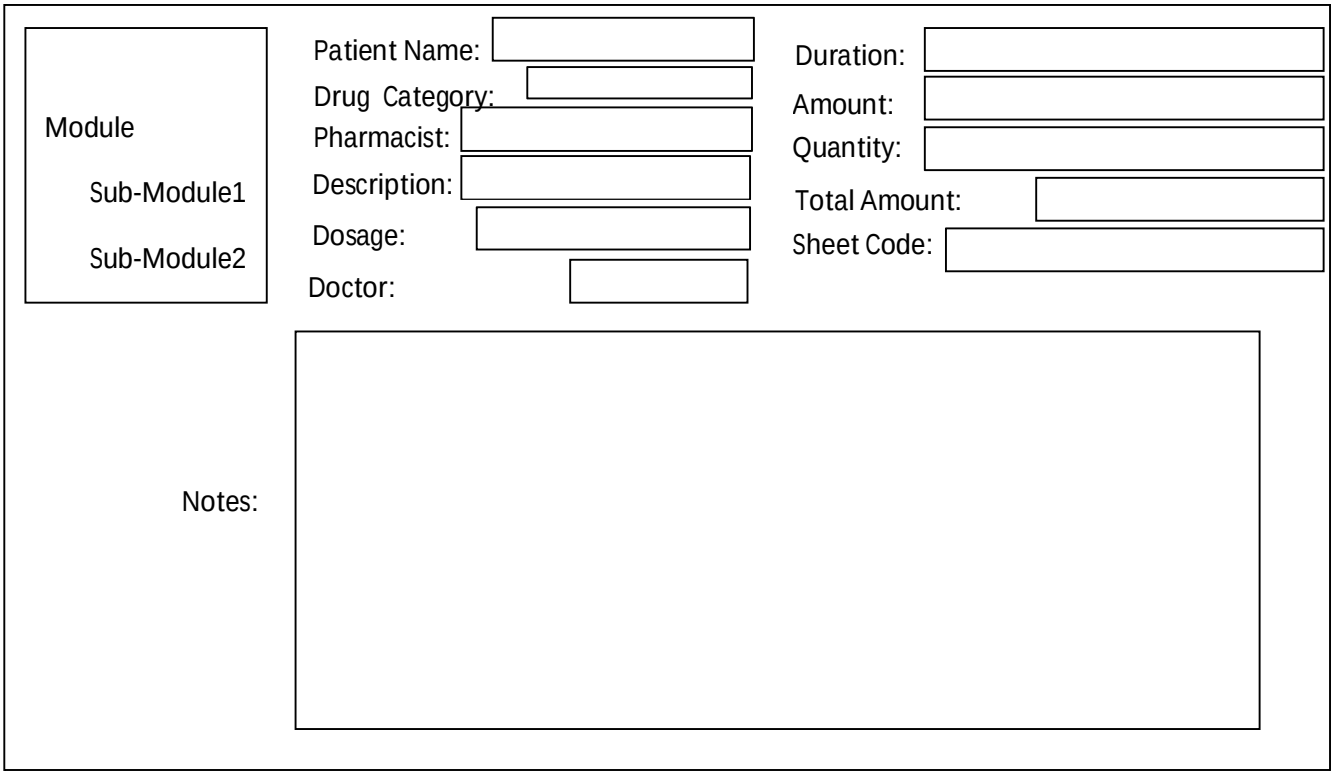


Fig 4.20: Proposed Automated Pharmacy Stocks Form

Table 4.28: Pharmacy Prescription (Sample Output)



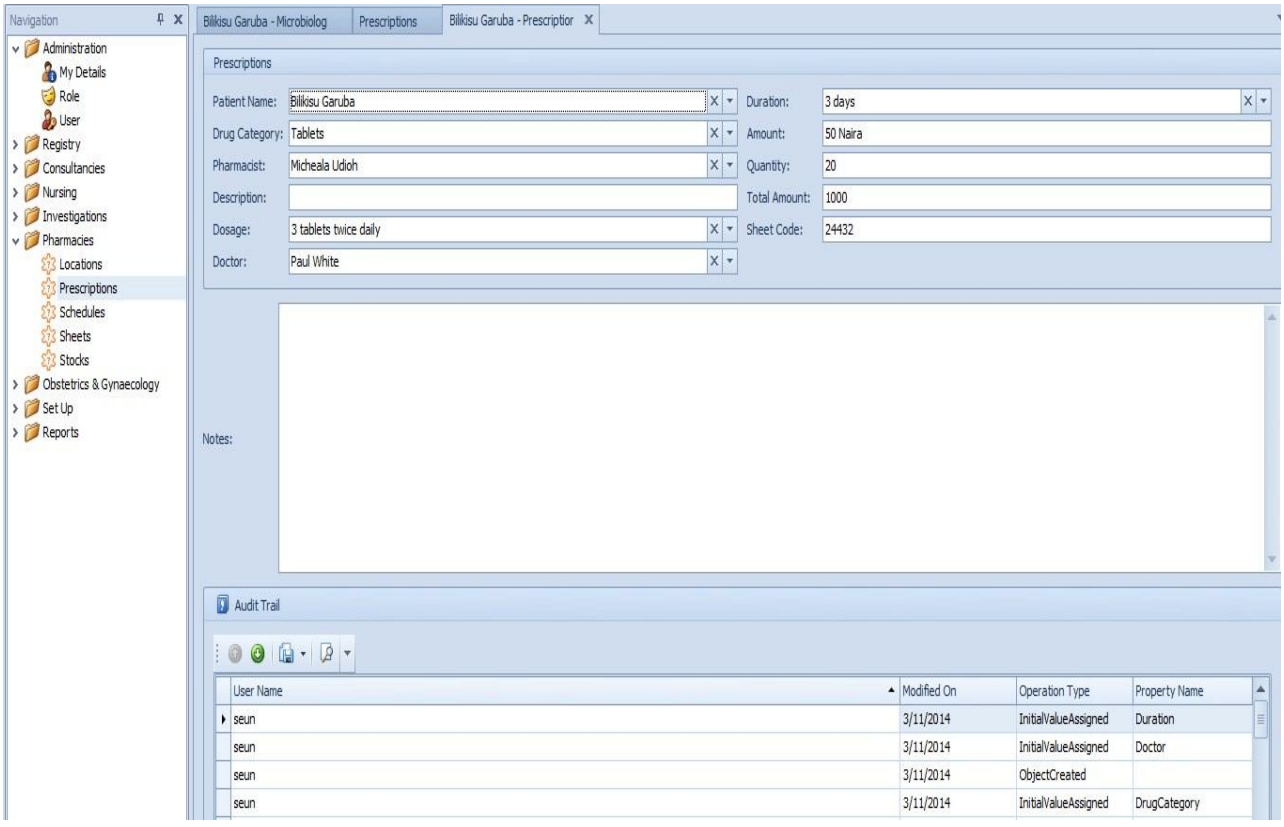
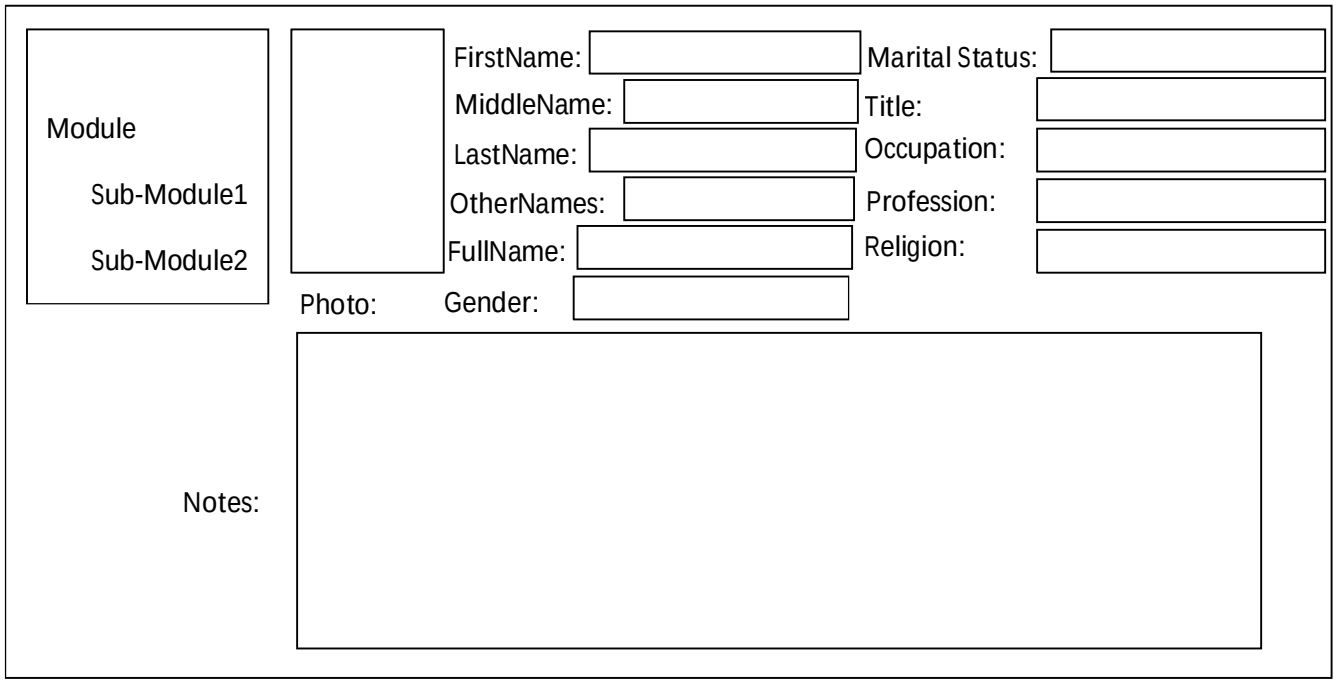


Fig 4.21: Proposed Automated Pharmacy Prescription Form

Table 4.28: Registry/Records(Sample Output)



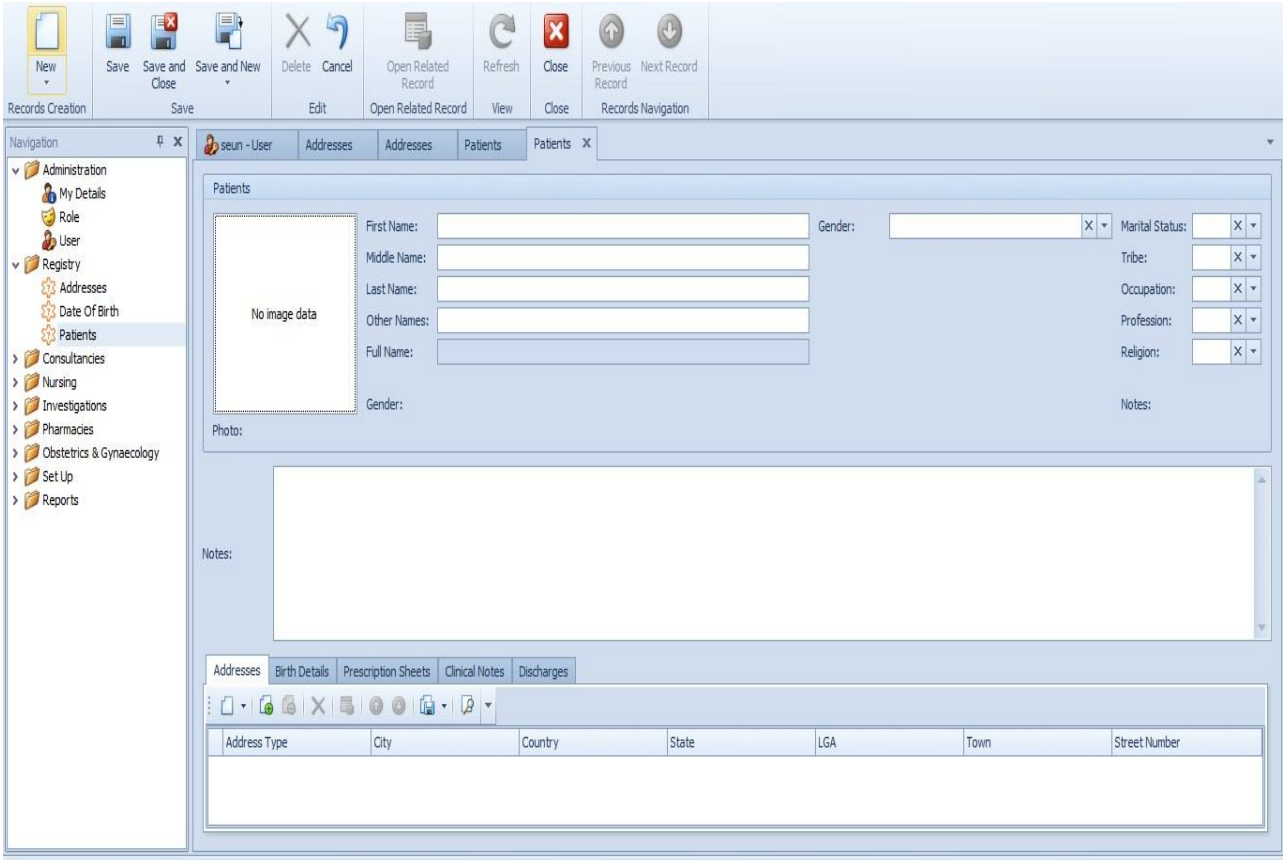
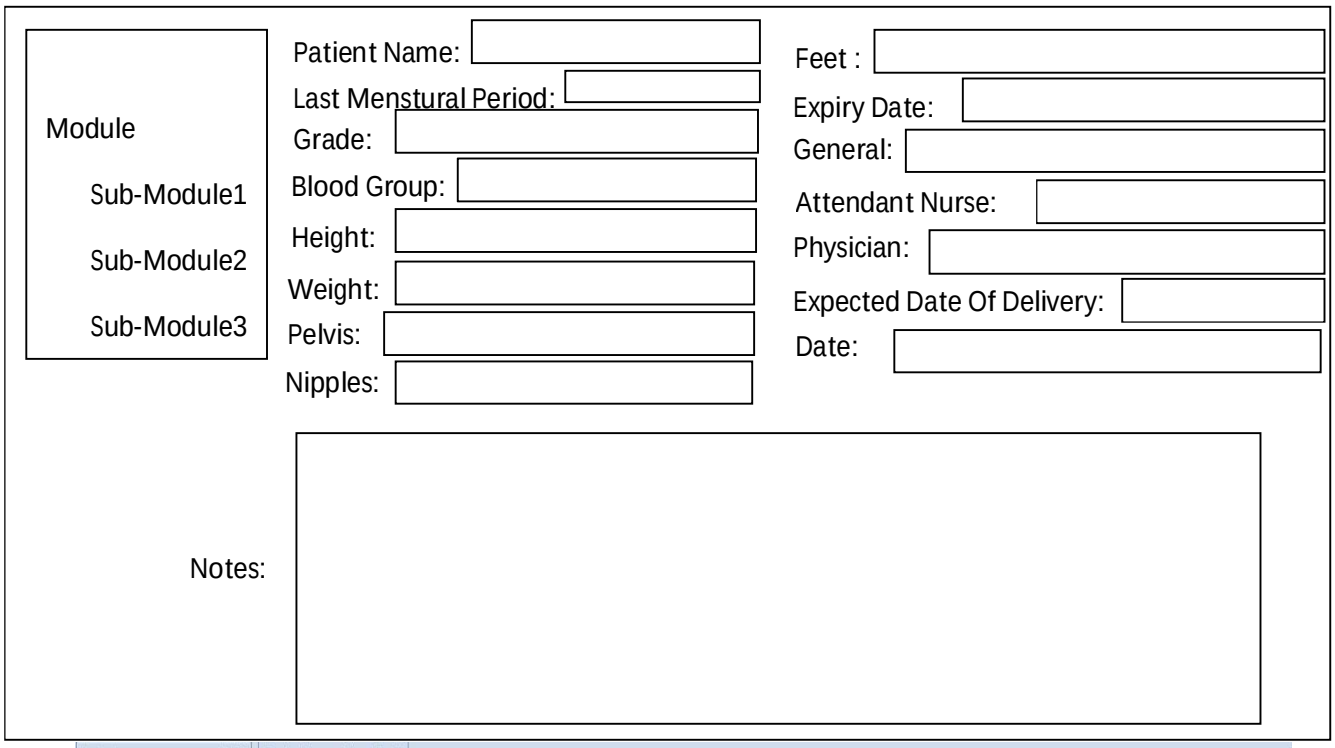


Fig 4.22: Proposed Automated Registry/Records Form

Table 4.29: Antenatals (Sample Output)


Module
Sub-Module1
Sub-Module2
Sub-Module3
Patient Name:
Last Menstural Period:
Grade:
Blood Group:
Height:
Weight:
Pelvis:
Nipples:
Feet :
Expiry Date:
General:
Attendant Nurse:
Physician:
Expected Date Of Delivery:
Date:
Notes:

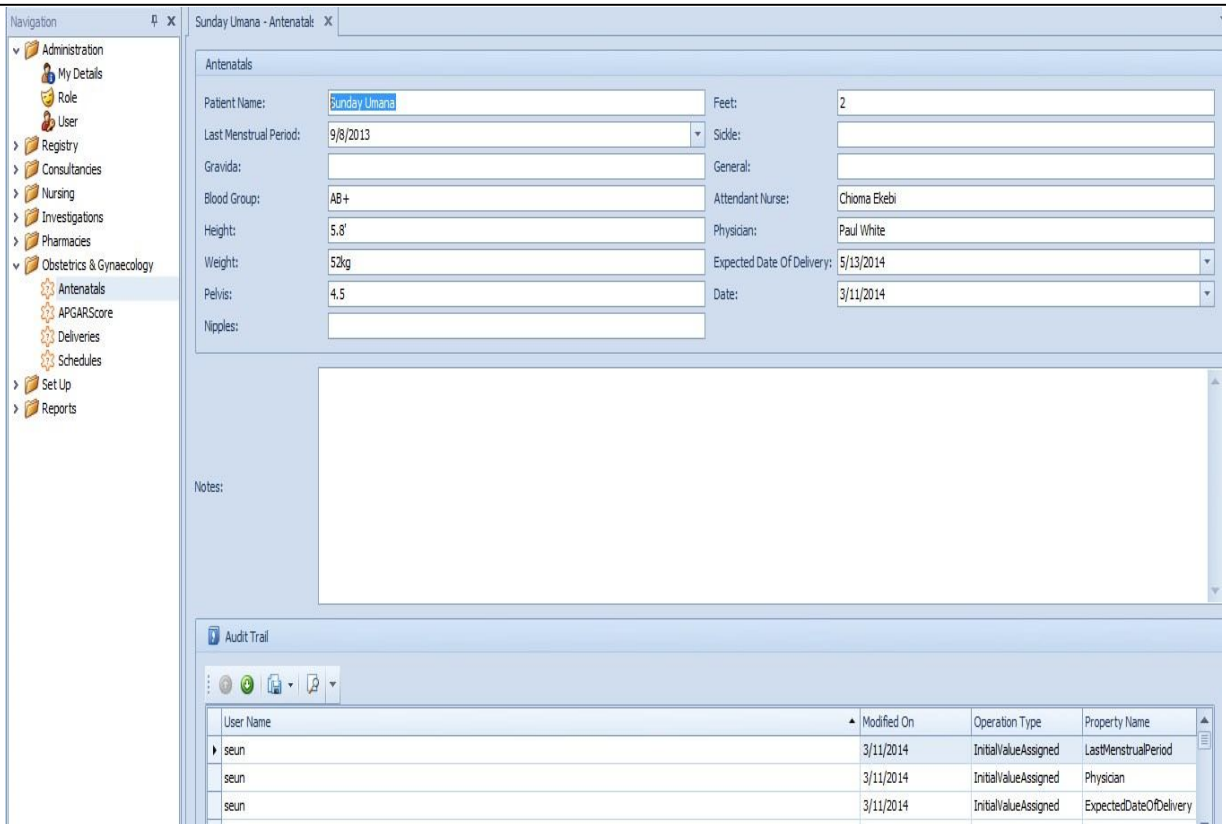
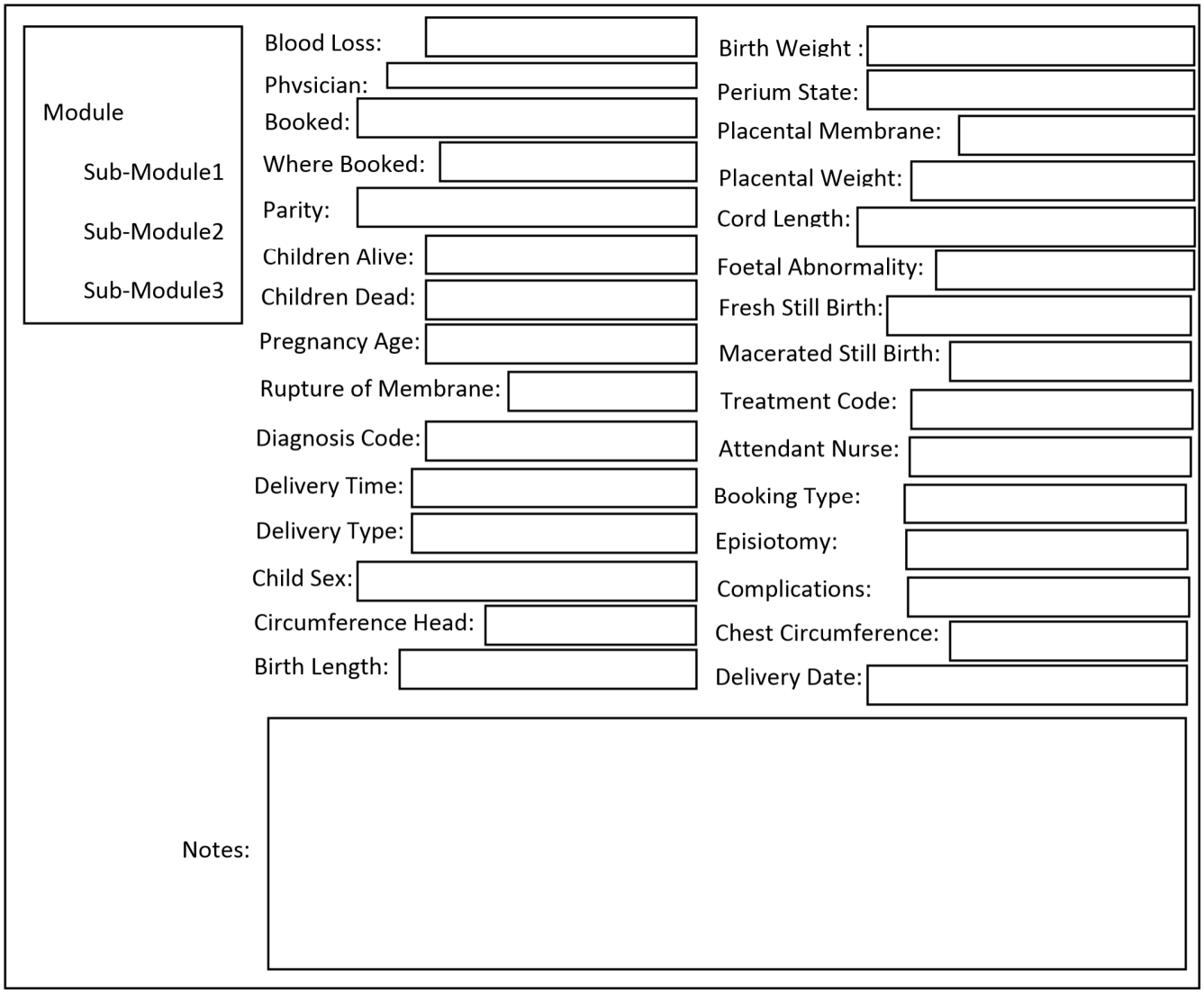

Navigation
Administration
My Details
Role
User
Registry
Consultancies
Nursing
Investigations
Pharmacies
Obstetrics & Gynaecology
Antenatals
APGARScore
Deliveries
Schedules
Set Up
Reports
Sunday Umana - Antenatals
Antenatals
Patient Name: Sunday Umana
Last Menstrual Period: 9/8/2013
Gravida:
Blood Group: AB+
Height: 5.8'
Weight: 52kg
Pelvis: 4.5
Nipples:
Feet: 2
Sickle:
General:
Attendant Nurse: Chioma Ekebi
Physician: Paul White
Expected Date Of Delivery: 5/13/2014
Date: 3/11/2014
Notes:
Audit Trail
User Name
Modified On
Operation Type
Property Name
seun 3/11/2014 InitialValueAssigned LastMenstrualPeriod
seun 3/11/2014 InitialValueAssigned Physician
seun 3/11/2014 InitialValueAssigned ExpectedDateOfDelivery

Fig 4.23: Proposed Automated Antenatals Form Table 4.29: Deliveries (Sample Output)



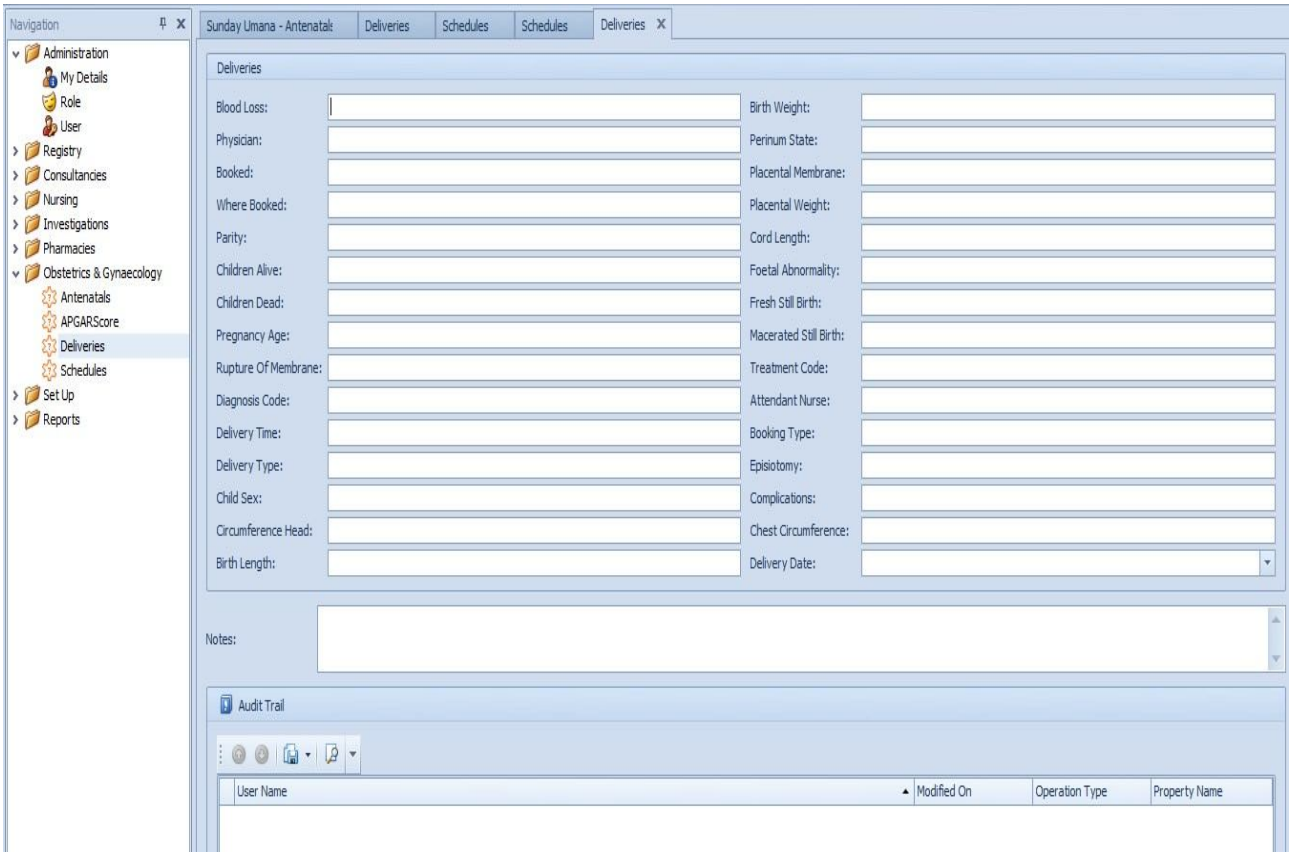


Fig 4.24: Proposed Automated Deliveries Form

Table 4.30: Schedules (Sample Output)

Module

Sub-Module1

Sub-Module2

Sub-Module3

Patient Name:

Fondus Height:

Presentation And Positioning:

Foetal Heart:

Oedema:

Urine:

Weight :

Haemoglobin:

Blood Pressure:

Treatment Code:

Date Visited:

Notes:

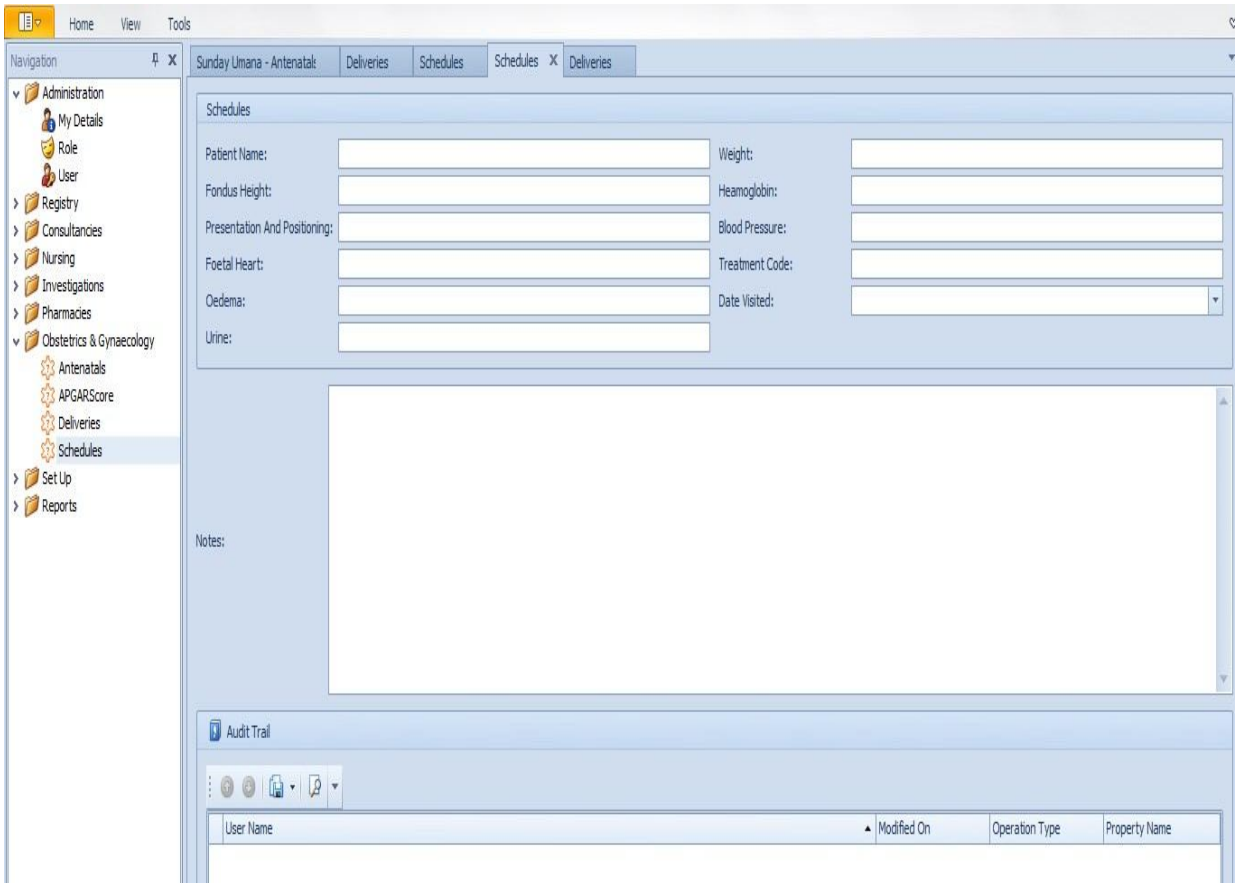


Fig 4.25:Proposed Automated Schedules Form

Table 4.31: Consultancy(Clinical Notes) Sample Output

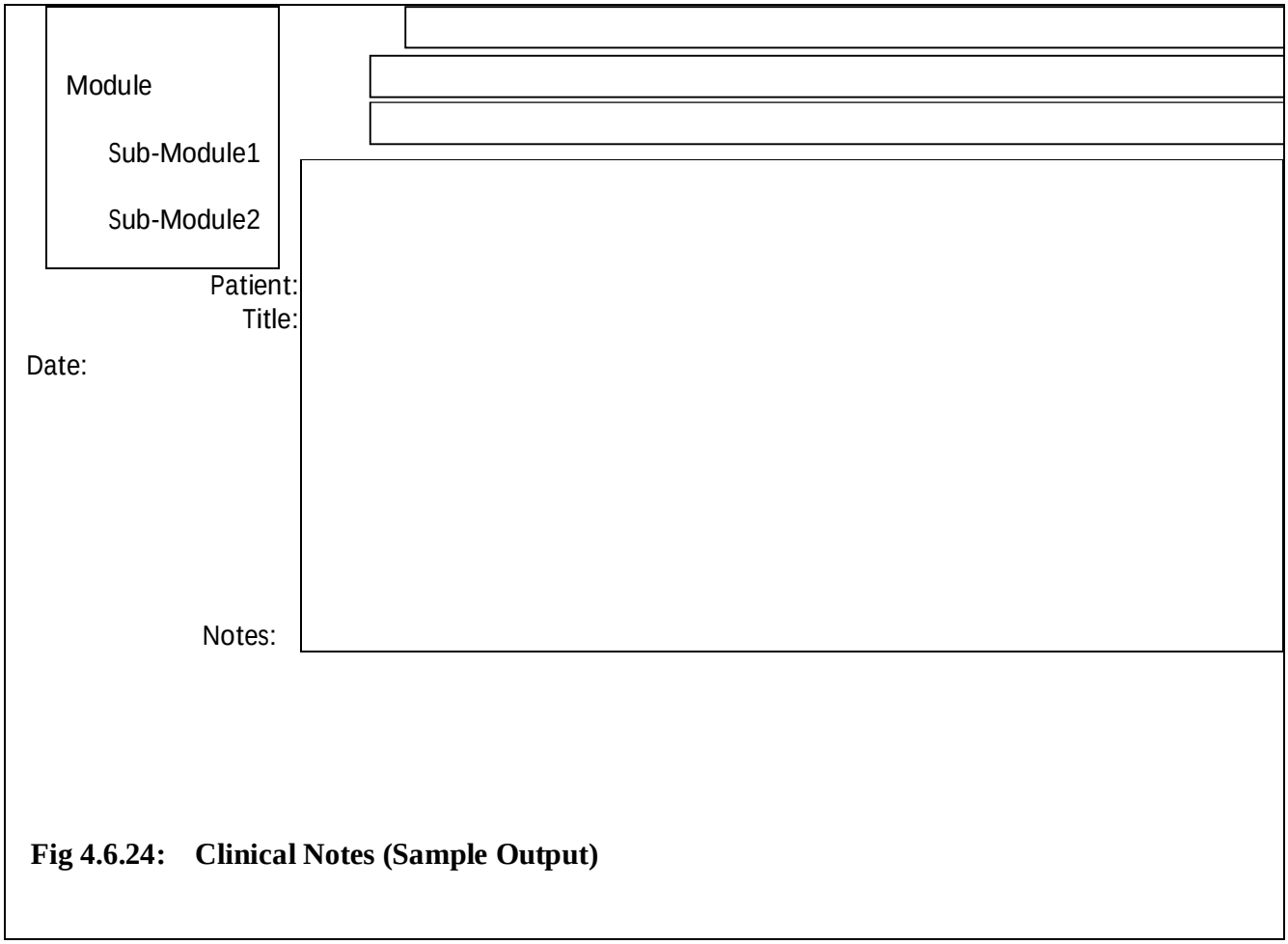


Fig 4.6.24: Clinical Notes (Sample Output)

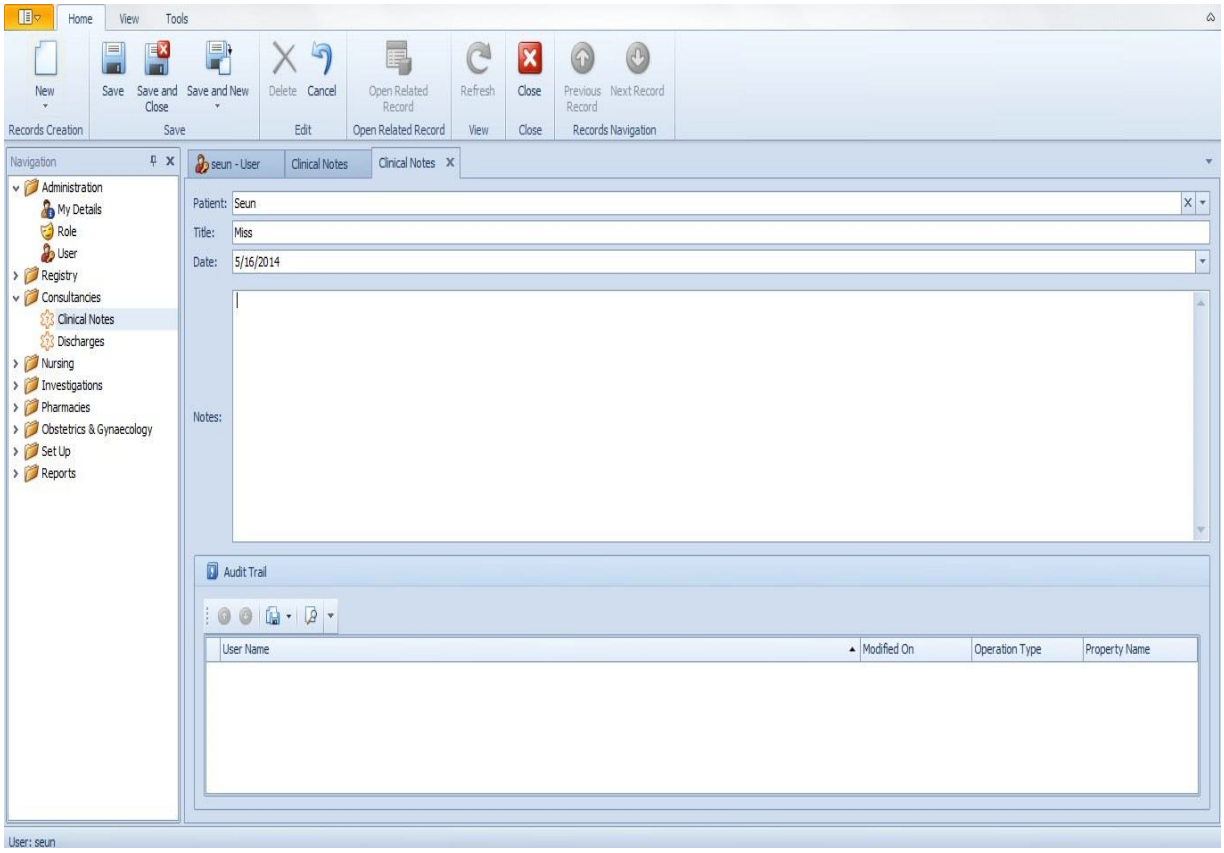


Fig 4.26: Proposed Automated Clinical Notes Form

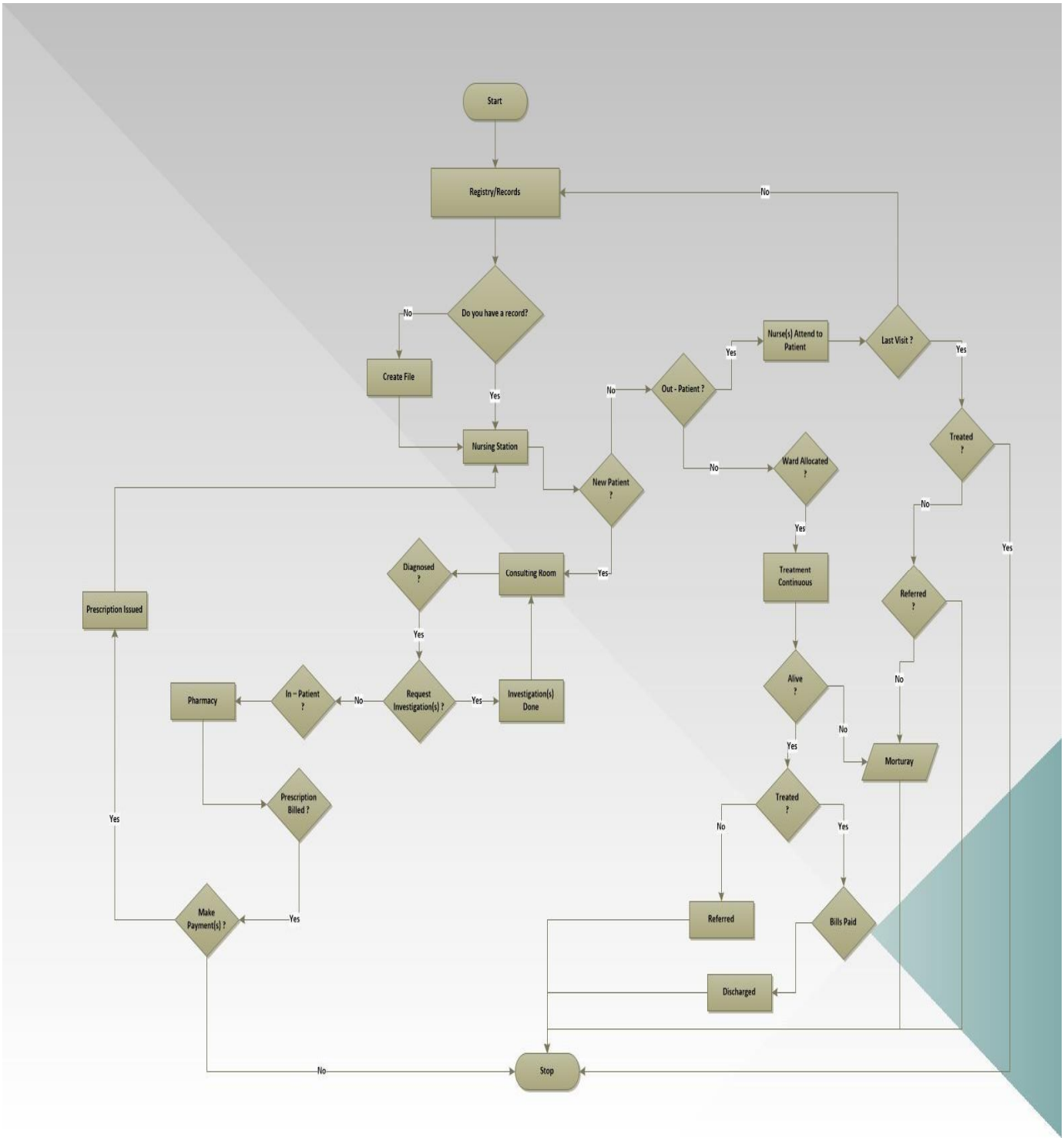


Fig 4.27 Overall Data Flow Diagram

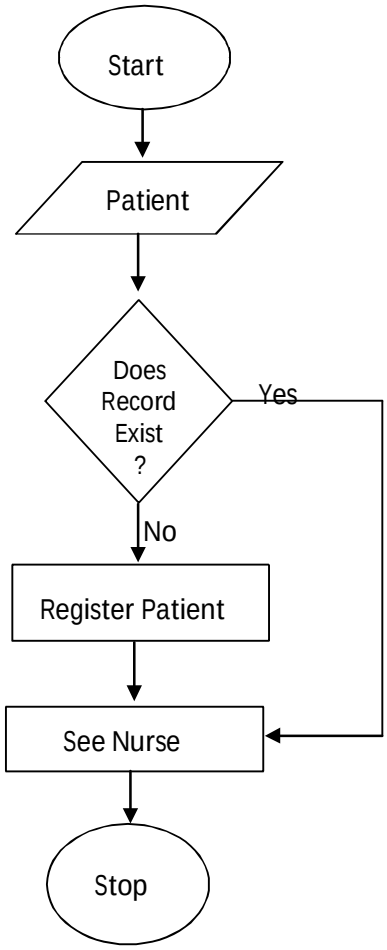


Fig 4.28: Pre-Iterative Data Flow

Pre-Iterative Flow

The pre-iterative flow simply shows diagrammatically a sequence of the first steps that occurs in a health care system in the software. In this case, the patient is first referred to the registry and records and a check is made to know if his/her record exist in the system. If the record doesn‘t exist, the patient is made to register. Then (s)he gets to see the nurse and the clinical process starts. But if the record exists the patient‘s record is taken directly to the nurse and the pre-iterative process flows ends at this point.

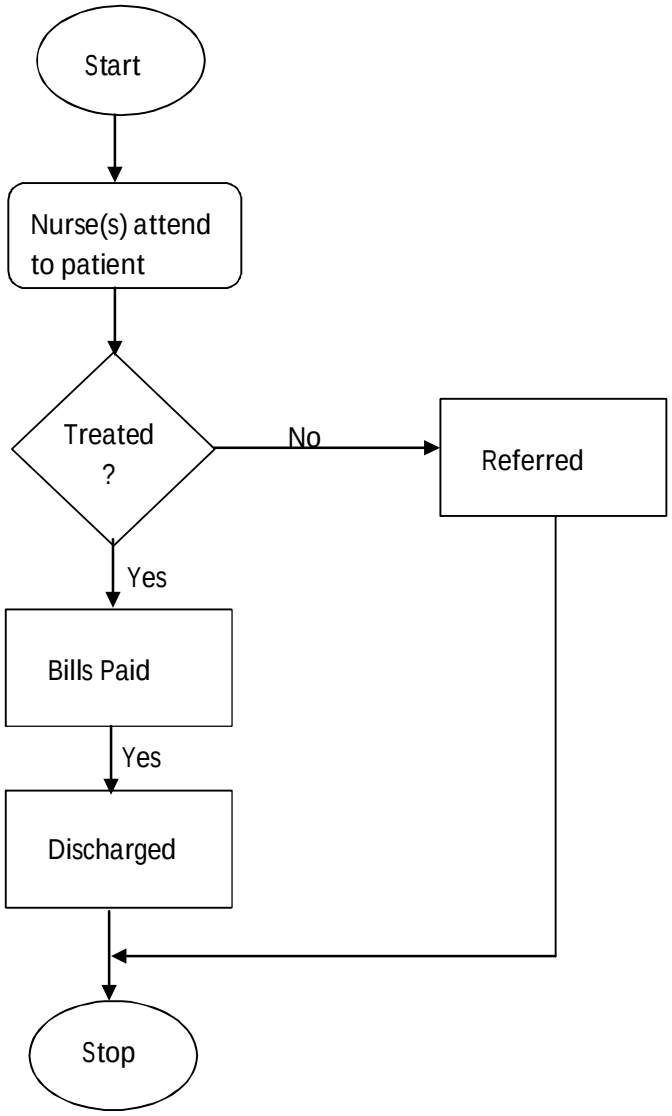


Fig 4.29: Iterative Data Flow

Iterative Data Flow

The Iterative flow is the sequence of events in the software that occurs repeatedly. In the health care environment, the iterative sequence of events starts with the nursing station. The patient is either treated or not.If (s)he is treated a rundown of the treatment is taken and (s)he is referred to the billing section for the bills to be paid and only after the bills are paid will the patient be discharged. If the patient is not treated then (s)he is referred to another hospital and the sequence of events at this stage comes to a halt.

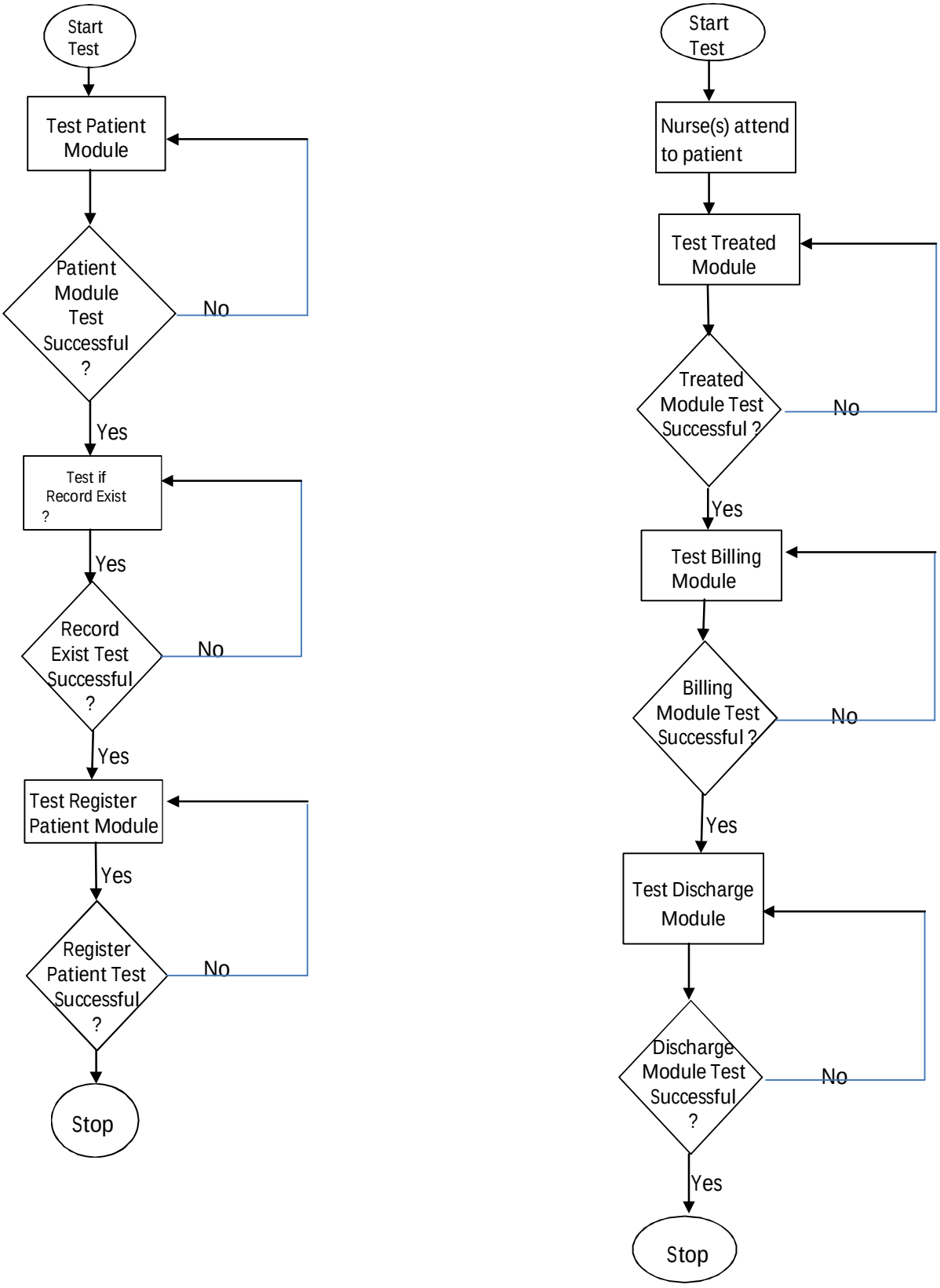


Fig 4.30: Post Iterative Flow Post-Iterative Flow

The post iterative-flow is the sequence of events that occurs before the software as a whole is produced. At this stage testing(debugging i.e checking to ensure that there are no errors) in all the various steps in the stages are done.

4.7Choice of Programming Language

The programming language used for the front end or client side of this thesis is the C sharp programming language version 4.0 (also known as C#).

C# Overview [Visual Studio 2012]

C# is a unified development model that includes the services necessary for you to build enterprise-class applications with minimum amount of coding. C# is part of the .NET Framework, and when coding C# applications you have access to classes in the .NET Framework. You can code your applications in any language compatible with the common language runtime (CLR), including Microsoft Visual Basic e.t.c. These languages enable you to develop C# applications that benefit from the common language runtime, type safety, inheritance, and so on.

It was based on the mentioned reason that C# 4.0 (Version 2012) has been chosen as the programming language in developing the Hospital Information Management System application.

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network.

Why choose SQL Server

Microsoft SQL Server is a comprehensive database server and information platform offering a complete set of enterprise-ready technologies and tools that help people derive the most value from information at the lowest total-cost-of-ownership. Enjoy high levels of performance, availability, and security; employ more productive management and development tools; and deliver pervasive insight with self-service business intelligence (BI).

A complete and integrated platform, Microsoft SQL Server brings it all together to get more value out of existing IT skills and assets, increase the productivity and agility of IT departments, and quickly build flexible, innovative applications.

4.8SYSTEM DOCUMENTATION AND IMPLEMENTATION

4.8.1How the Program Will Be Installed

Before we start the installation of the HIMS software there are a few things we need to do. One of them is to get the Visual Studio 2012 application software installed on the system. The development environment on which this application is built as at the time this article is written already has the Visual Studio 2012 installed. The steps taken to install the HIMS software are on the ground that the Visual Studio 2012 has been installed.

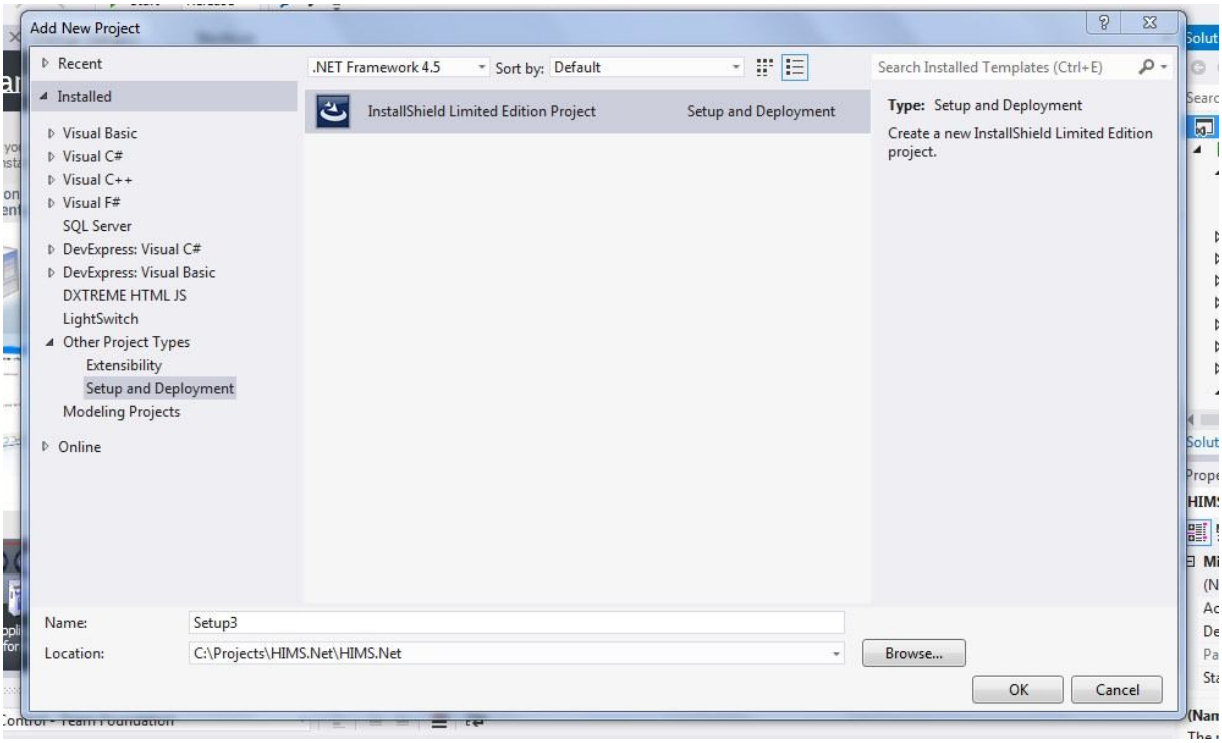
4.8.2 Pre-Installation

Step 1

Launch the Visual Studio Application

Step 2

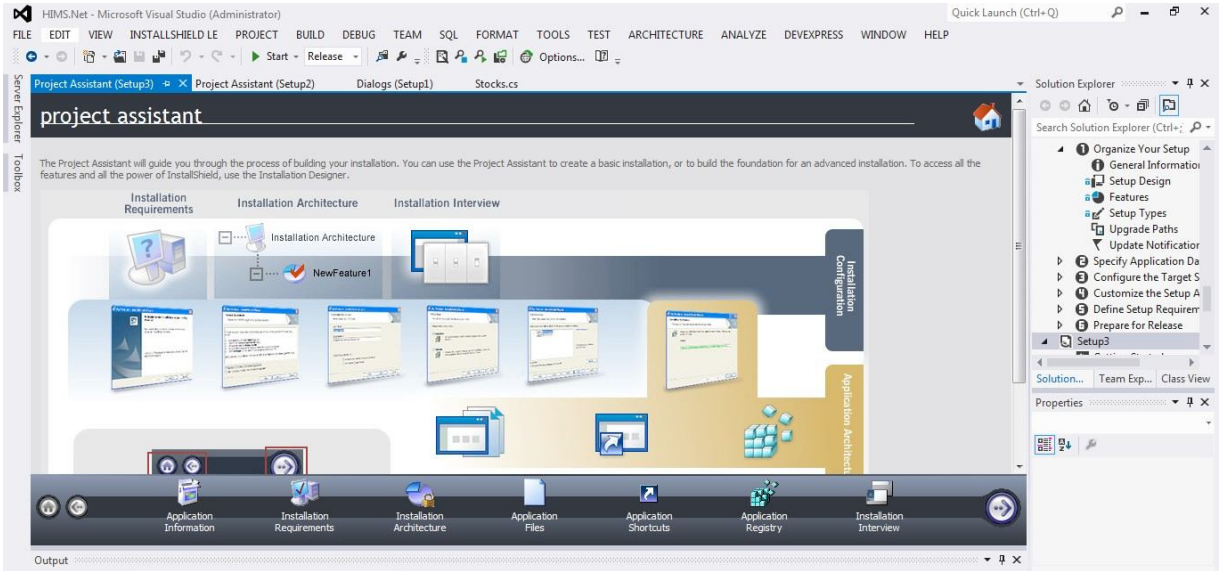
Navigate to the Solution Explorer pane and right click on Solution HIMS**.** Go to **—Add New**‖, Click on **NewProject.** A screen such as the one below appears.



Step 3

Click on **Setup and Deployment**(on the above screen) and provide name in the **Name** textbox and click **Ok**.

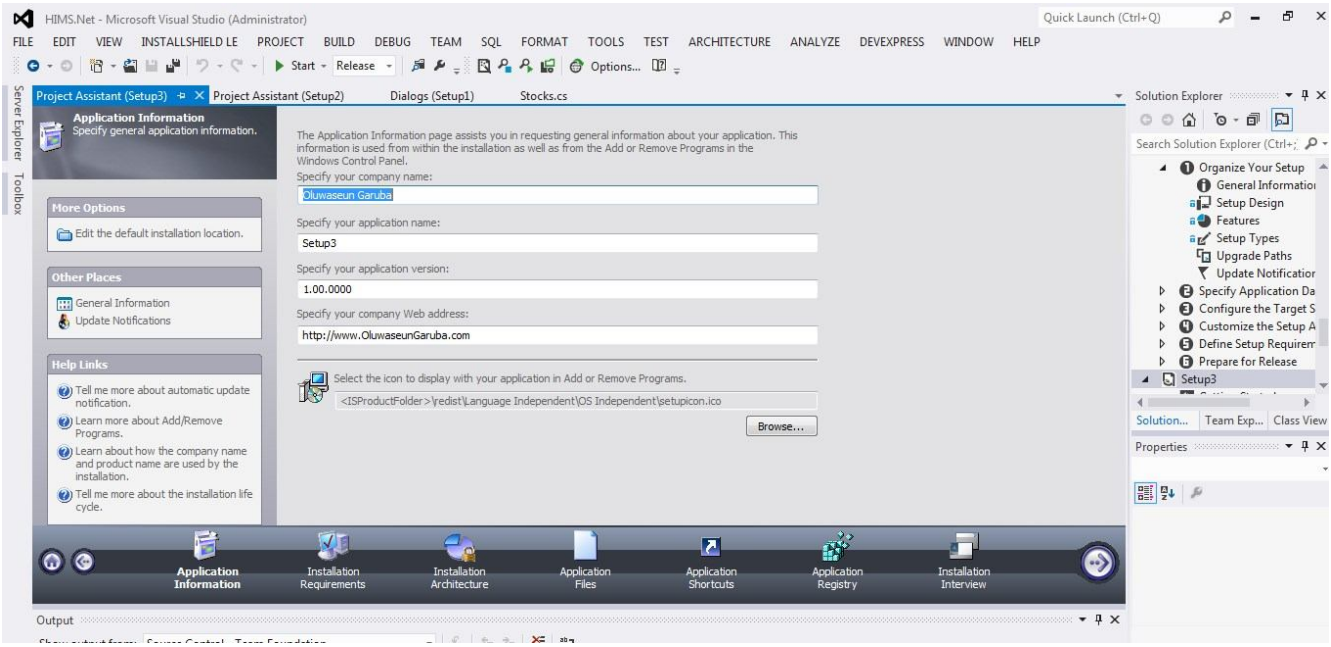
Step 4



On the screen that appears as the one above, click on **Application Information.**

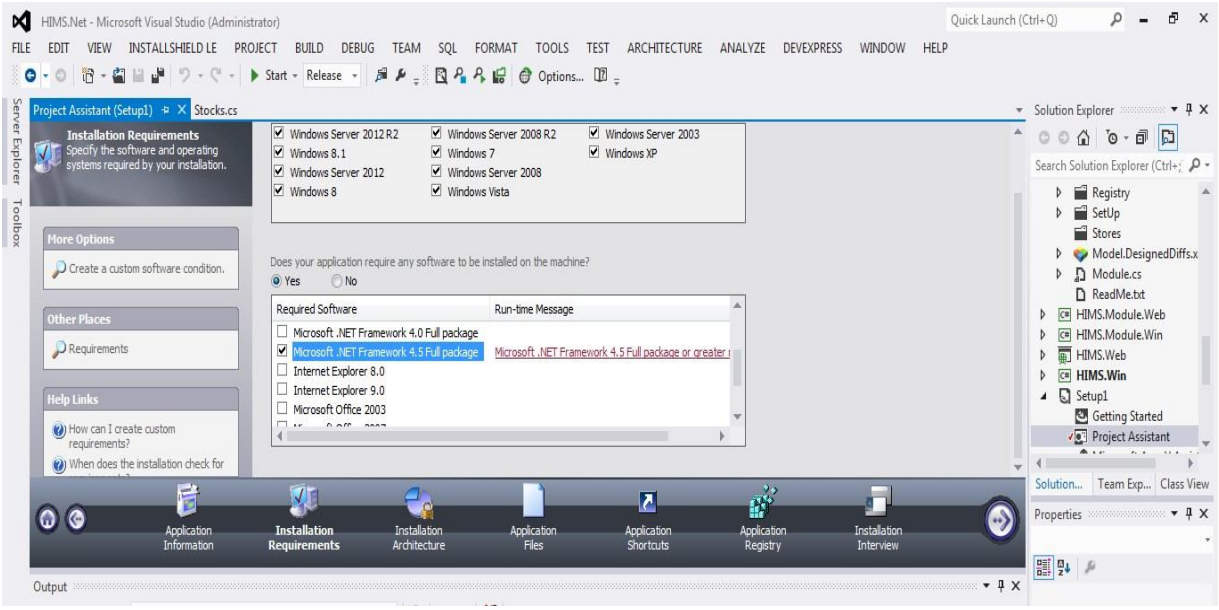
Step 5

On the **Application Information** page, specify your **Company Name**, **Application Name**, **Application Version** and **Company Web address** in the spaces provided. The screen that captures this is as shown below.



Step 6

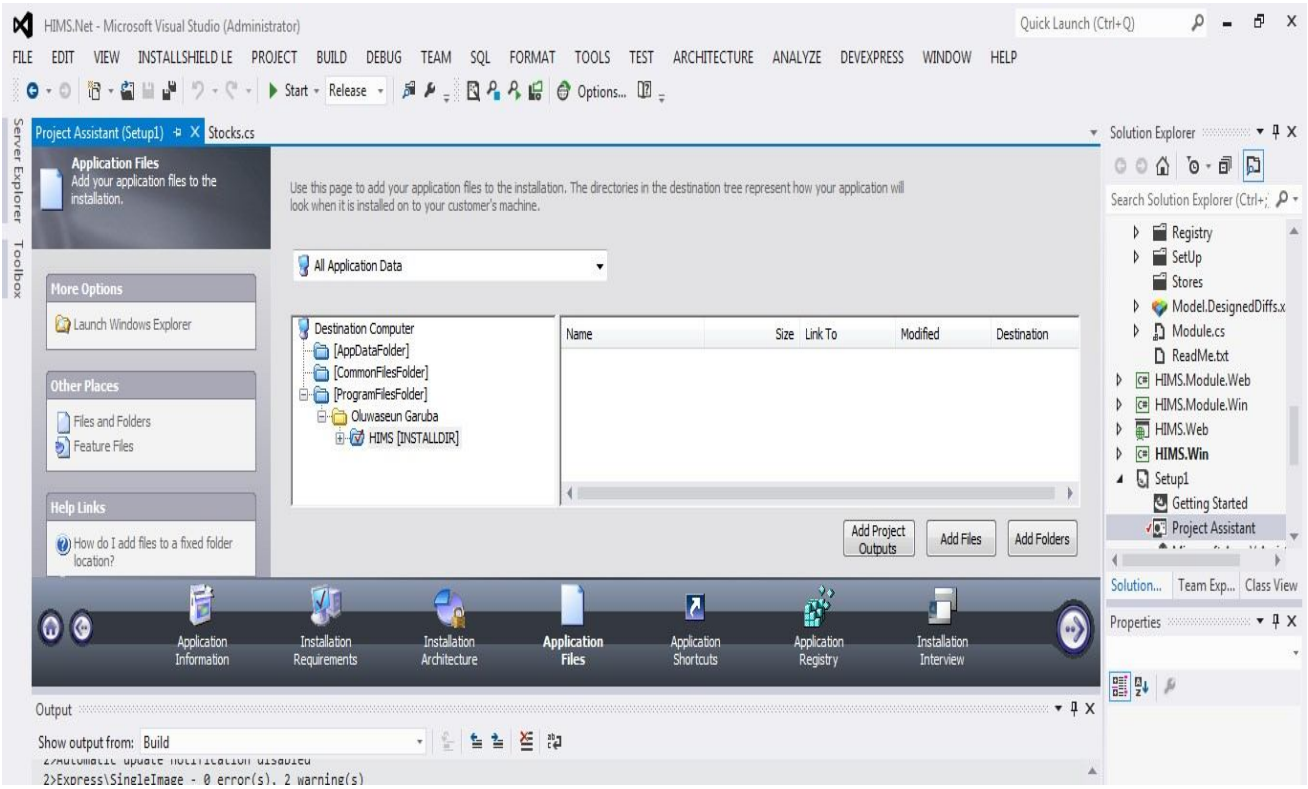
Click on **Installation Requirement.** On the **Software Requirement** area, click **Yes** radio button and tick the **Microsoft.Net Framework 4.5 Full Package** checkbox. An example is shown on the screen below



Step 7

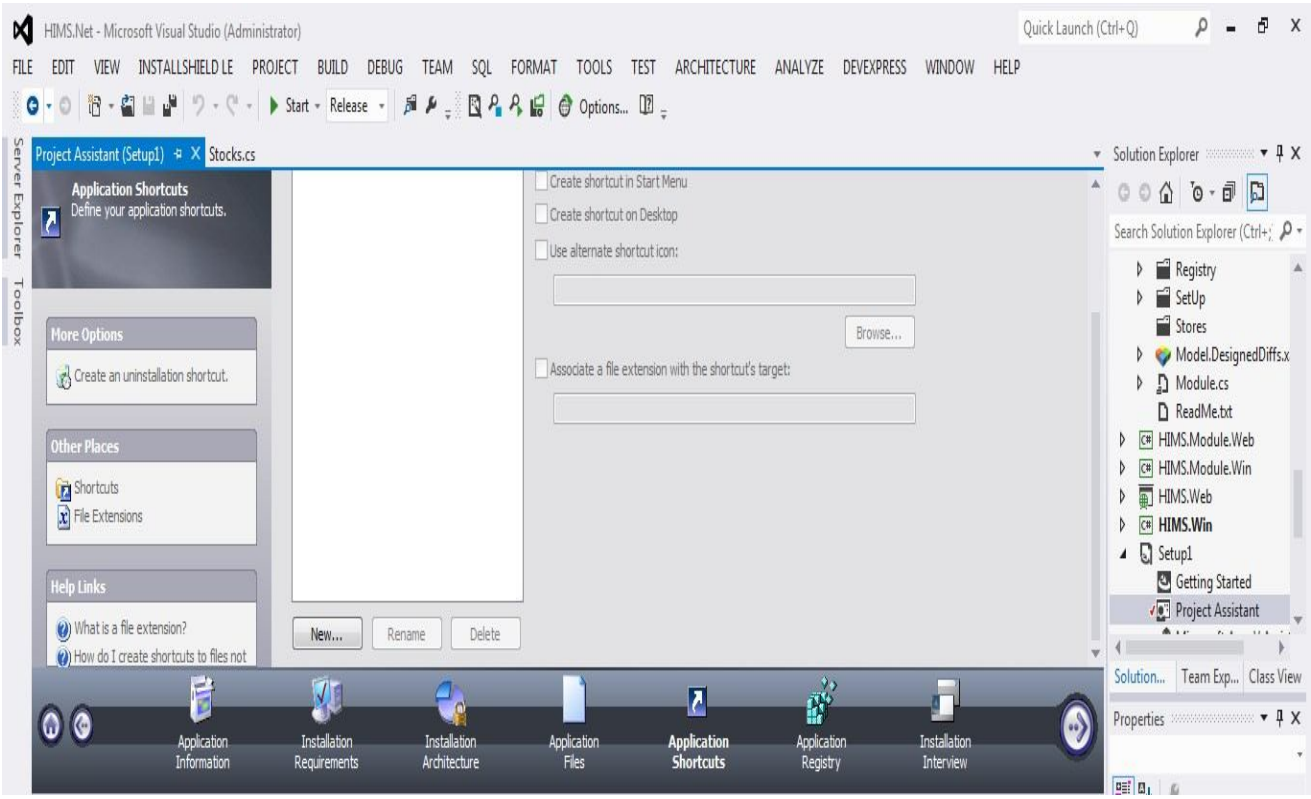
Move to **ApplicationFiles,** click **Add Projects** to specify the folder location for your application.

The screen below shows a representation of what this means.



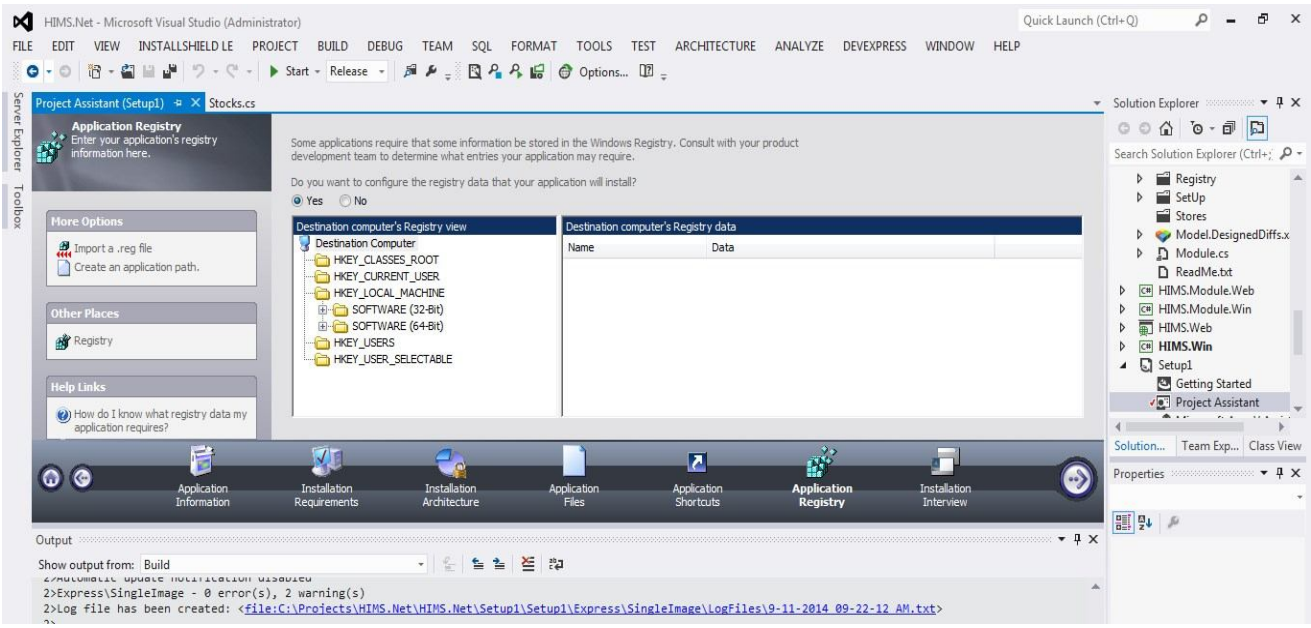
Step 8

Click **Application Shortcuts** to create a shortcut on the start menu or the desktop.



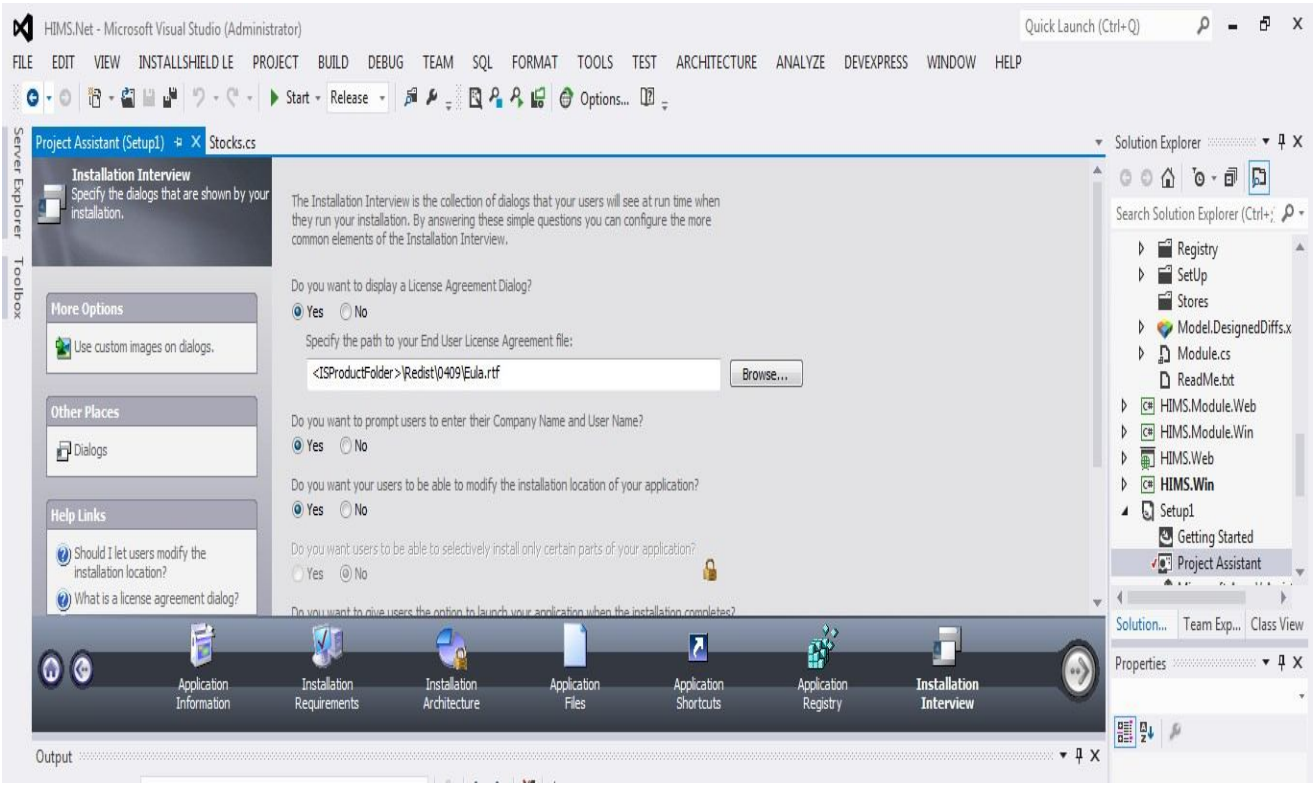
Step 9

Navigate to **Application Registry,** click on the **Yes** radio button to configure the registry data that your application will install. Below shows a screen explaining this.



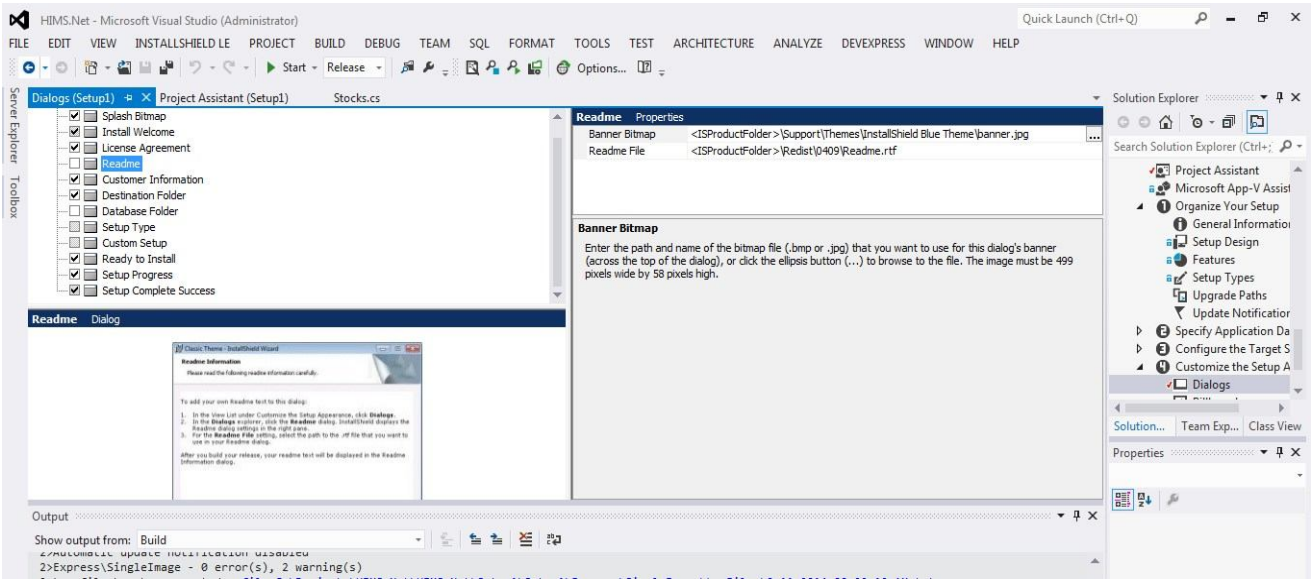
Step 10

Navigate to the **Interview** section, Click **Yes**to the License Agreement Dialog, prompt users to enter their company names and user name.



Step 11

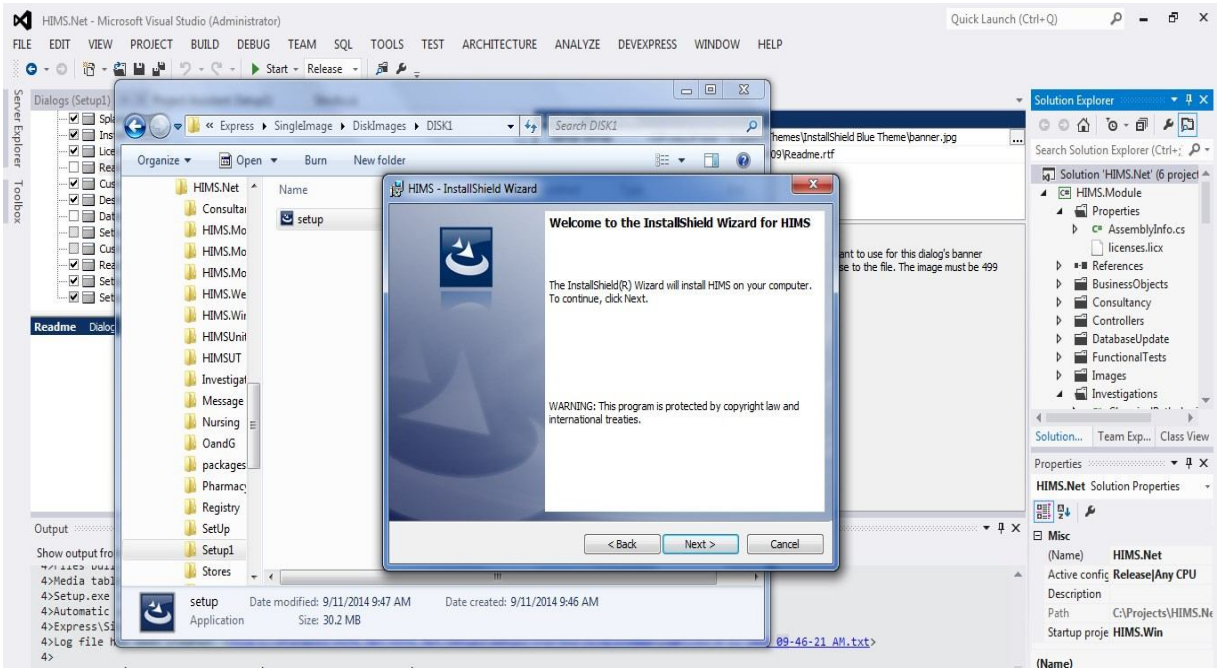
On the **Dialogs tab**, check the boxes of what you want to be a part of your program installation process.



4.8.3Program Installation

Step 1

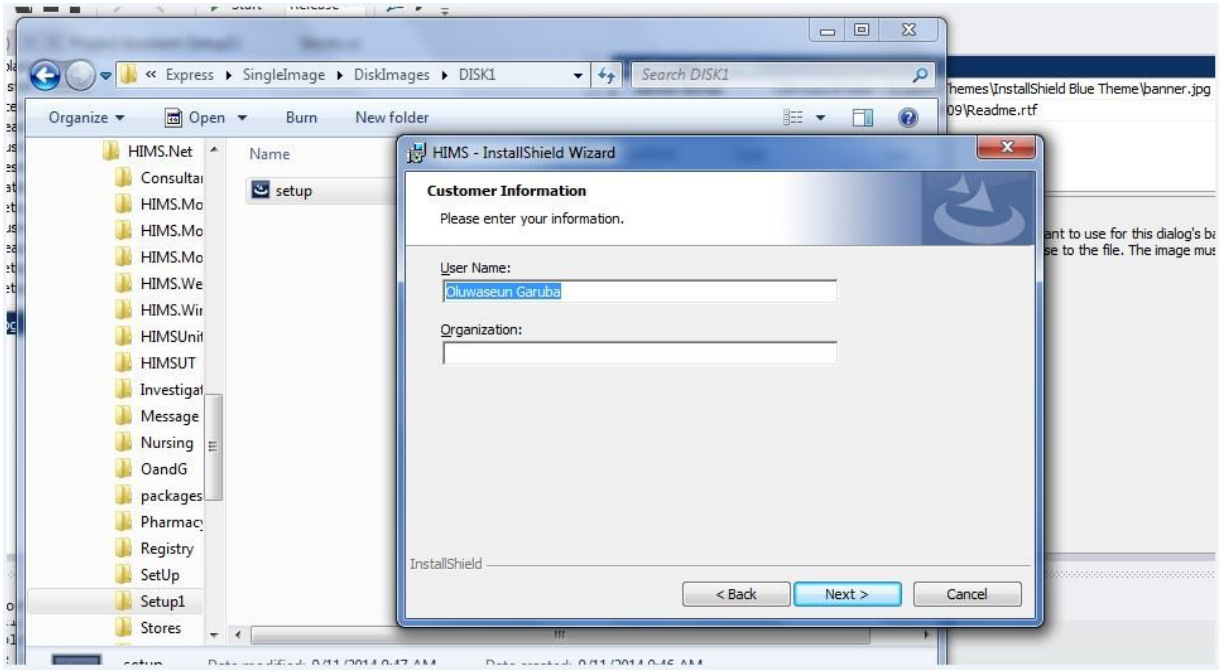
Locate the folder where the application resides and double click on the setup icon. The screen shot below explains this.



Click **Next**

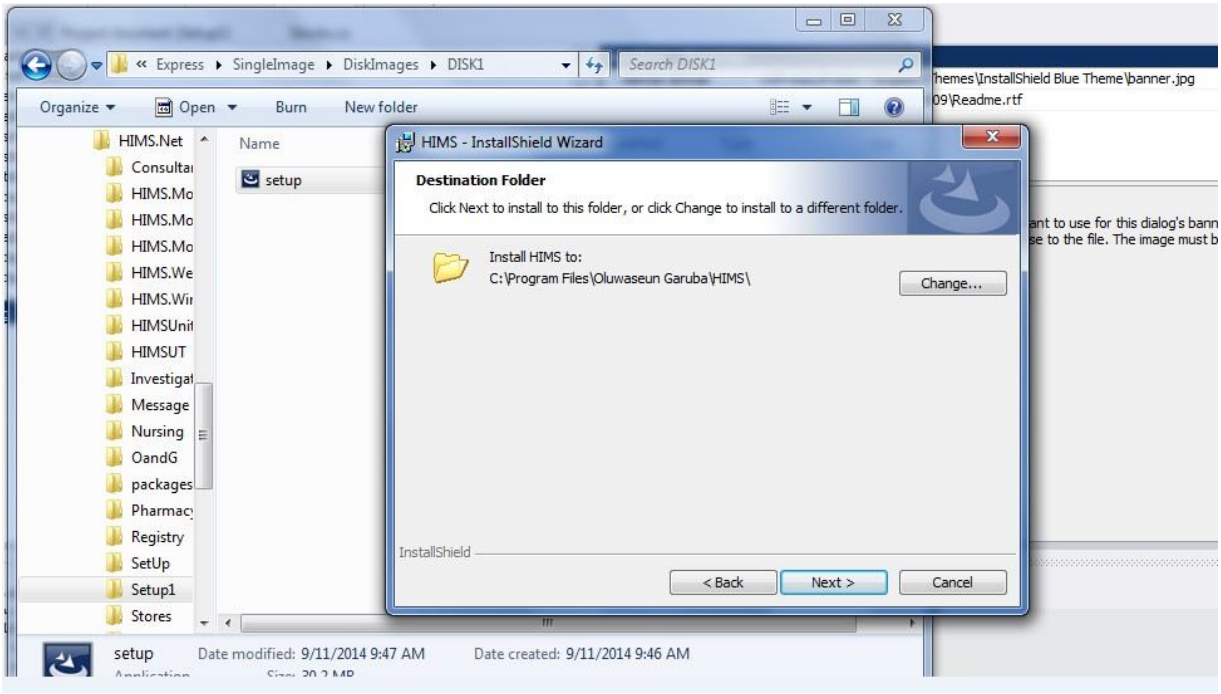
Step 2

Enter **User Name** and **Company Name**and click **Next**.



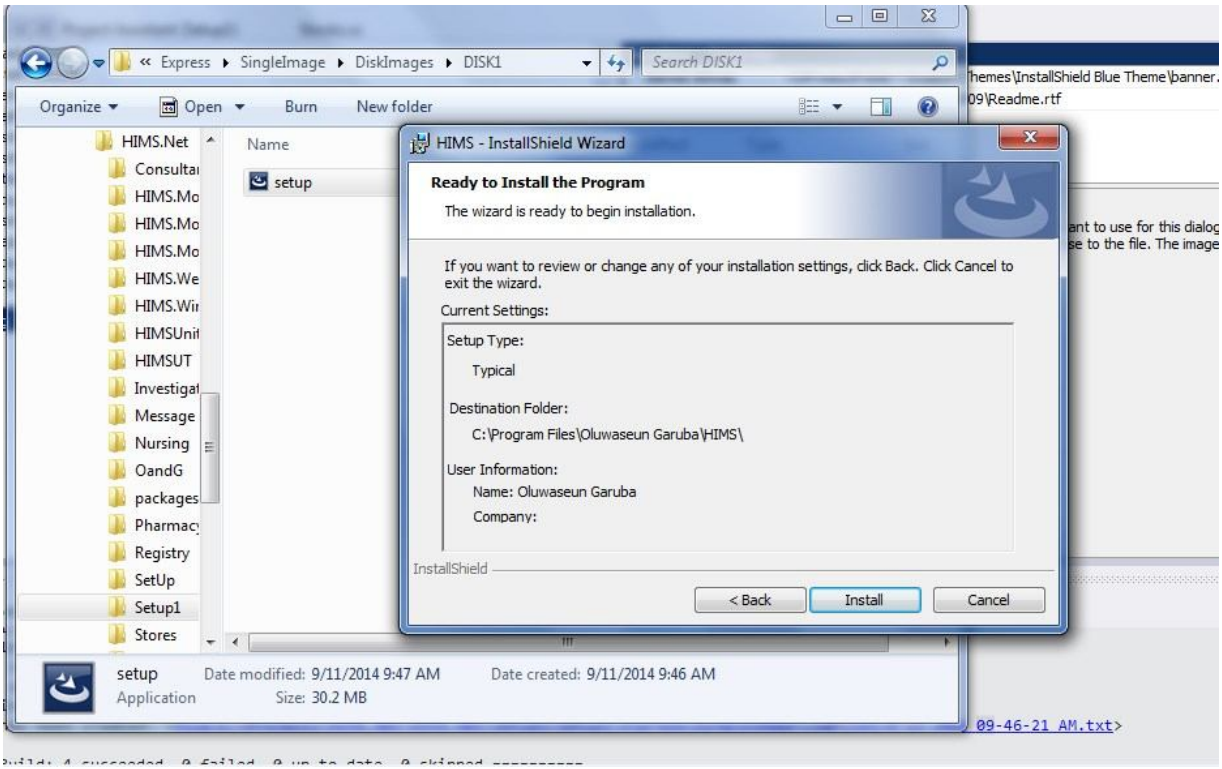
Step 3

Specify the program destination folder and click **Next.**



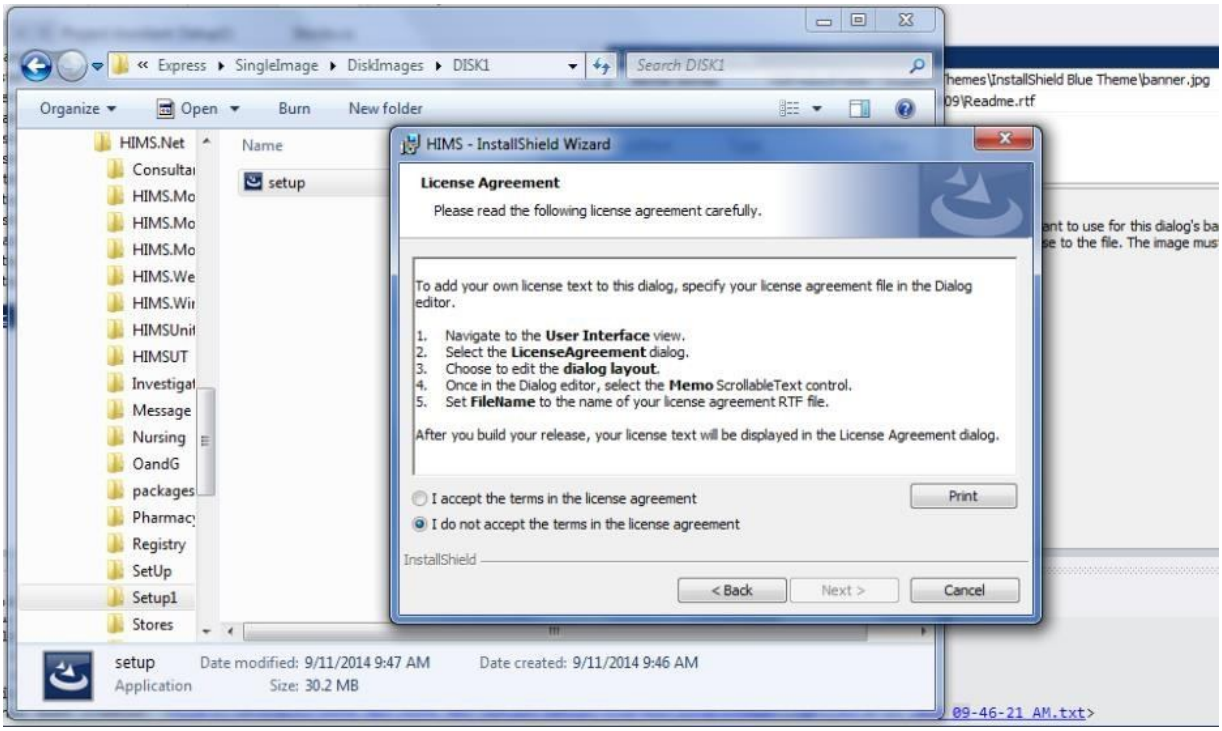
Step 4

Click on **Install** to install the HIMS program.



Step 5

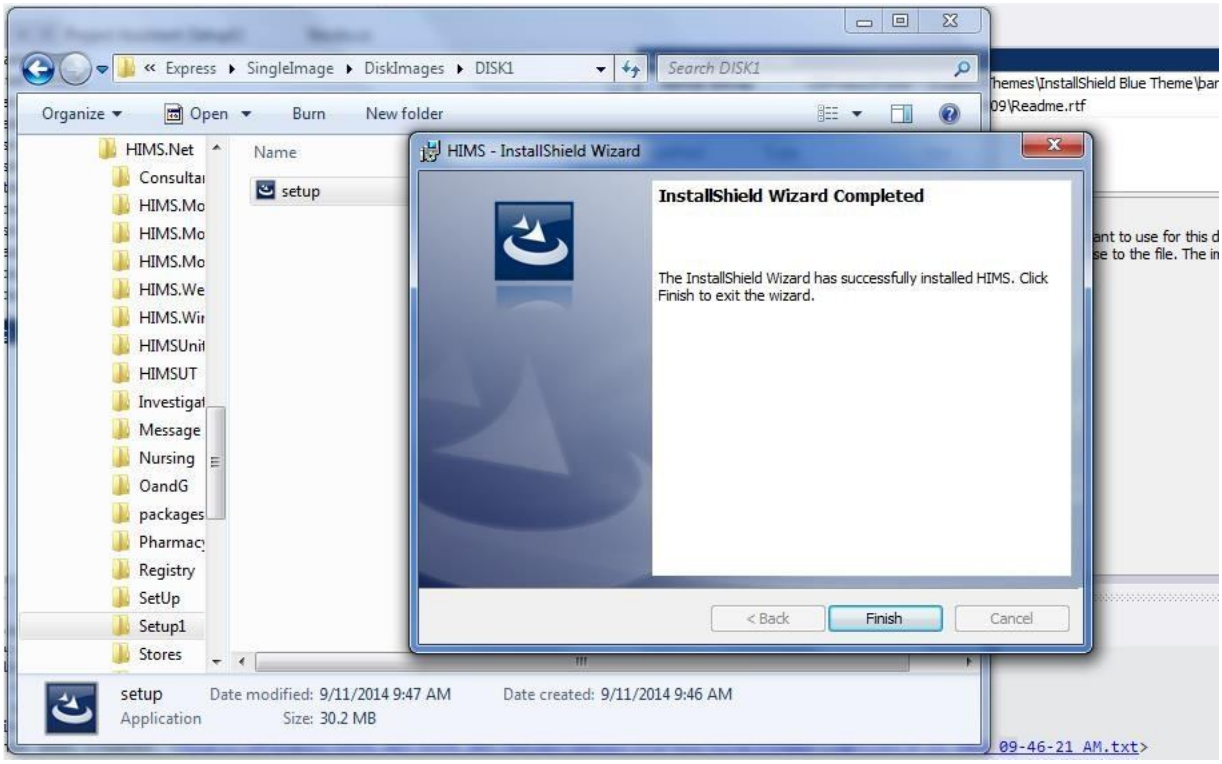
Click on the **I accept the terms in the License Agreement** radio button to accept the terms and conditions and click **Next**.



The installation begins

Step 6

Click **Finish** and the program is installed on the system.



Microsoft SQL Server 2012 Installation:

Step by Step Instructions

What are SQL Server 2012 OS Requirements?

As per Microsoft, SQL Server 2012 can be installed on Windows 7, Windows Server 2008 R2, Windows Server 2008 Service Pack 2, Windows Vista Service Pack 2.

Note: SQL Server 2012 installation on Windows 7 required me to install Windows 7 SP1. It is recommended to perform Windows update before beginning SQL Server 2012 installation.

Microsoft SQL Server 2012 on 32-bit and 64-bit processors

32-bit systems – Computer with Intel or compatible 1GHz or faster processor (2 GHz or faster is recommended.)

64-bit systems – 1.4 GHz or faster processor

What are memory requirements for SQL Server 2012?

Minimum of 1 GB of RAM (2 GB or more is recommended.) **What**

are hard disc requirements for SQL Server 2012?

2.2 GB of available hard disk space **SQL**

Server 2012 Installation:

Step 1

Insert SQL Server 2012 installation media into your DVD drive/virtual disc drive.

At this point, you will be prompted (based on your Windows settings) with AutoPlay dialog as shown in the image below:

Click on Run SETUP.EXE (Alternatively you can launch set up by locating and double clicking on SETUP.EXE file on the media drive)

At this time, SQL Server 2012 install will run background processes and you should see the following dialog.

STEP 2:

If your OS is compatible, at this point the install should show you the SQL Server Installation Center.

Click on _Installation' link (2nd in the list)

STEP 3:

Click on _New SQL Server stand-alone installation or add features to an existing installation' (See the image below). All the options on this screen are self-explanatory and I recommend you to explore and read through the options for your learning.

At this point, the install wizard will check for various setup rules and present you with any issues that needs to be resolved for set up to continue (See image below).

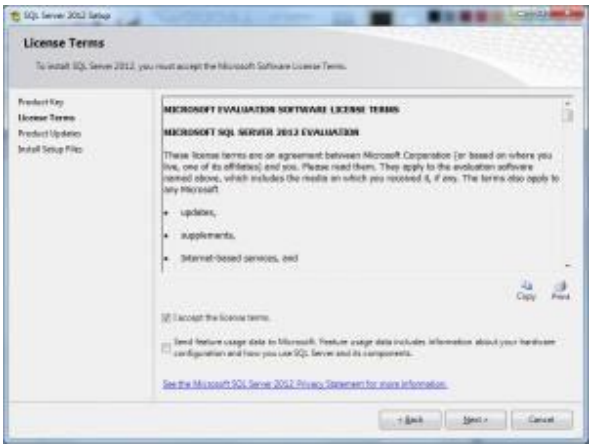
STEP 4:

In this step the install will ask for product key. A default option of _Specify a free edition‘ should be already selected and _Evaluation‘ is selected from the drop-down list (See image below).

STEP 5:

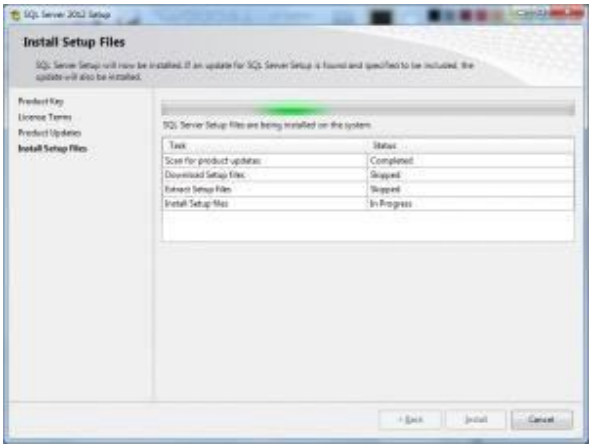
Accept SQL Server 2012 Evaluation license terms and conditions (See image below).

Click Next.

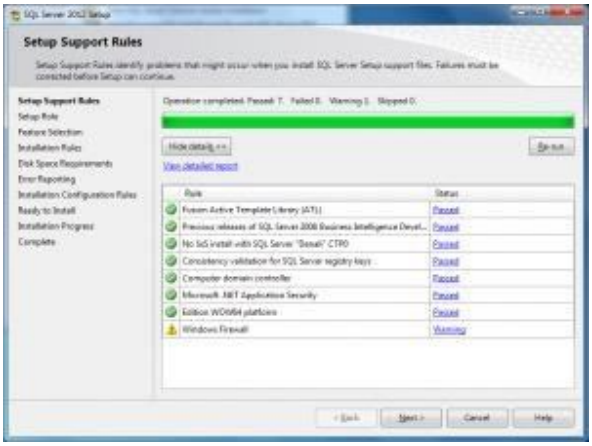


STEP 6:

In the next screen, SQL Server 2012 install wizard will ask you to install setup files needed. Click on _Install' button (see image below).

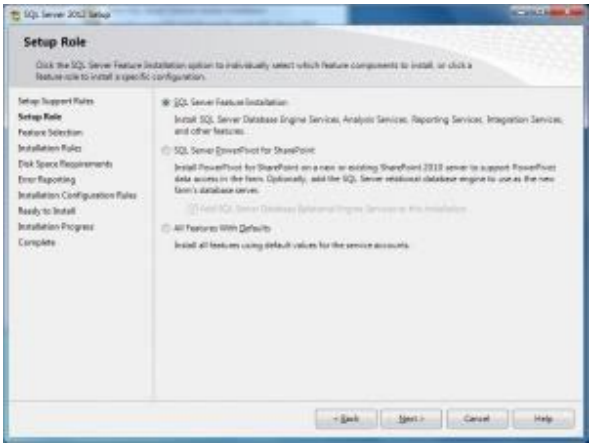


Once the setup files are installed, you should see the install complete screen (see image below).Click Next.



STEP 7:

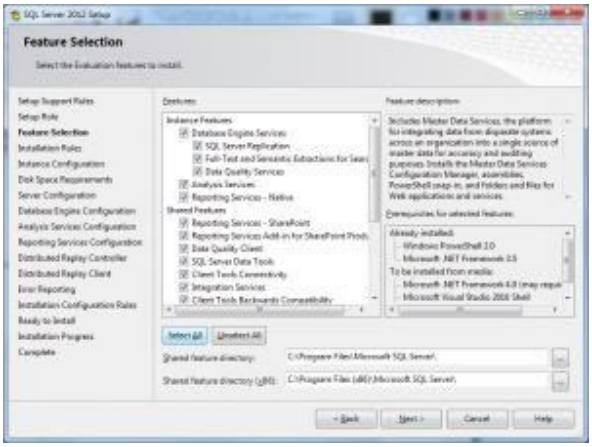
In this step make sure _SQL Server Feature Installation‘ is selected and click Next to continue (see image below).



Step 8:

At this time you should see _Feature Selection‘ screen. I recommend you to click on _Select All‘ if you are new to installing SQL Server (which is why I guess you are using these instructions).

See image below.

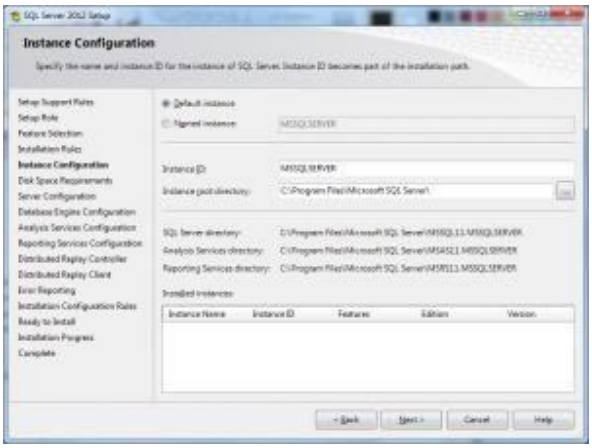


At this point, the wizard will again check your system for any blocking issues (see image below).

Upon successful completion click _Next‘ **STEP 9:**

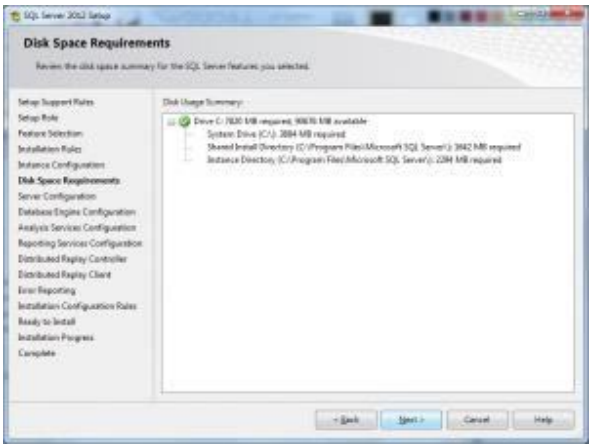
In this step you will be able to create a new SQL Server Instance. However, I recommend to go with default settings i.e, default instance (see image below).

Click _Next‘ to continue.



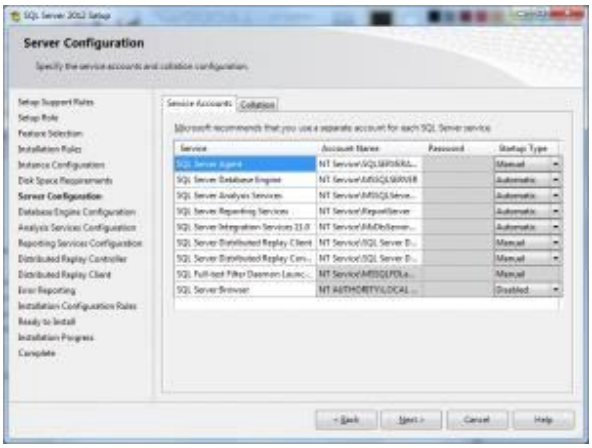
STEP 10:

Installer will check your system for required disk space and provide you a summary (see image below). Click Next.

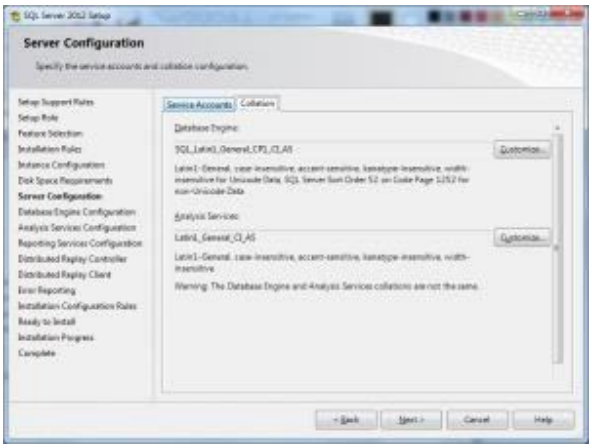


STEP 11:

Unlike previous versions of SQL Server, SQL Server 2012 uses an already created individual and independent service accounts for running various SQL Server services. Review and leave default options as-is (See image below).



Before clicking on Next to continue, click on collation tab and read through descriptions for your learning. Leave default selections as-is (see image below).



Click Next to continue.

STEP 12:

In this step, please follow the instructions carefully!!!

As you can see in the image below, you can have either windows authentication mode or mixed mode. Windows authentication mode lets you add only windows users while mixed mode lets you create users with a unique login and password.

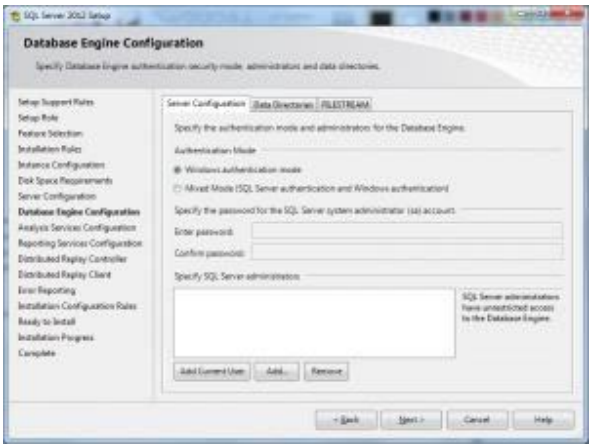
Make sure you select Mixed Mode (converse to what is shown in the image below).

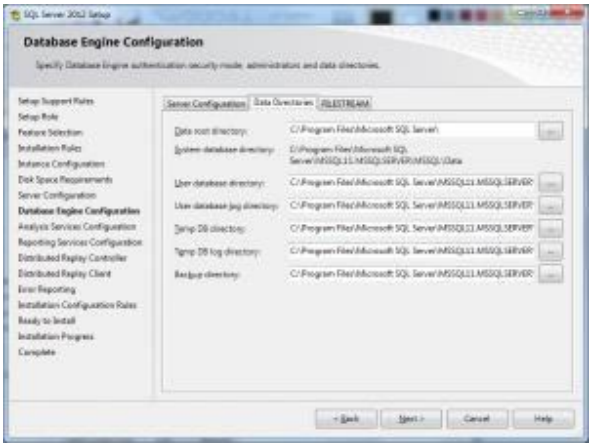
Enter a password for _sa‘ account.

Also make sure to click on _Add Current User _ which will add the account with which you are currently logged in into windows as a admin account to SQL Server.

Note: Image below does not reflect the instructions above. Follow the instructions above carefully.

Click on Data Directories and FILESTREAM tabs and review for your learning (See images below).



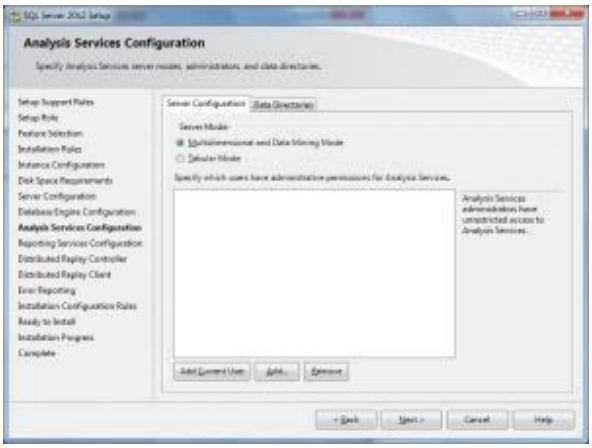


Click Next to continue.

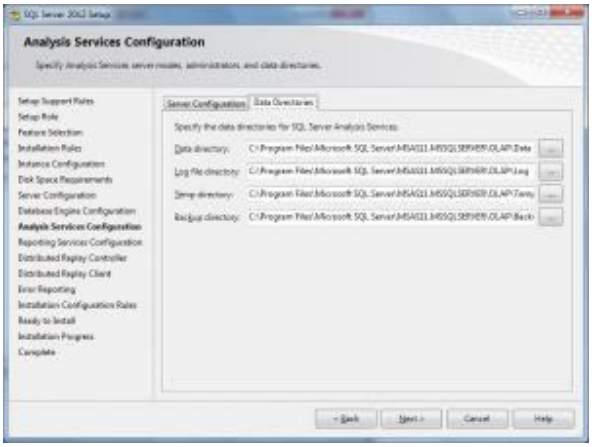
STEP 13:

Leave default selections as-is.

Make sure to click on _Add Current User' which will add the account with which you are currently logged in into windows as an administrator account for Analysis Service Engine.



Click on Data Directories tab to review (see image below).

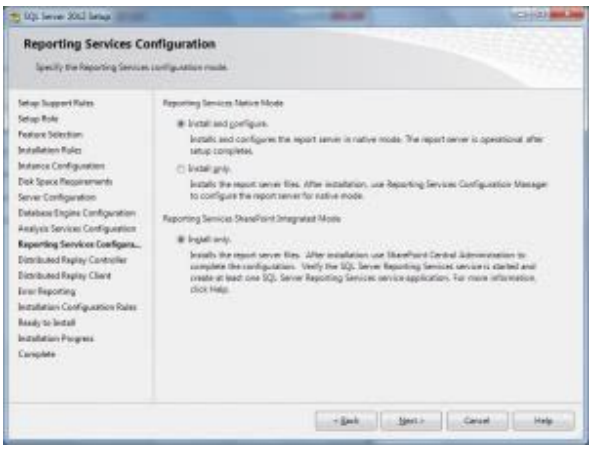


Click Next to continue.

STEP 14:

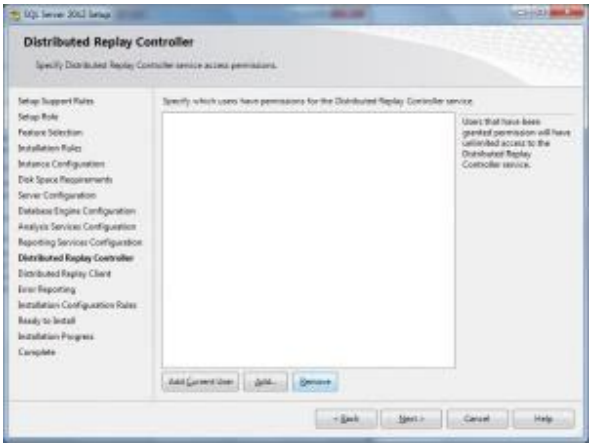
In this step leave the default options as-is.

Click Next to continue.



STEP 15:

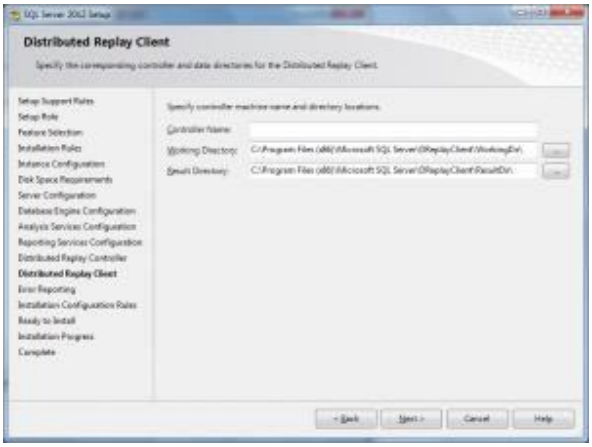
In this step Click Add Current User to add your windows account as an administrator account for Distributed Replay Controller.



Click Next to continue.

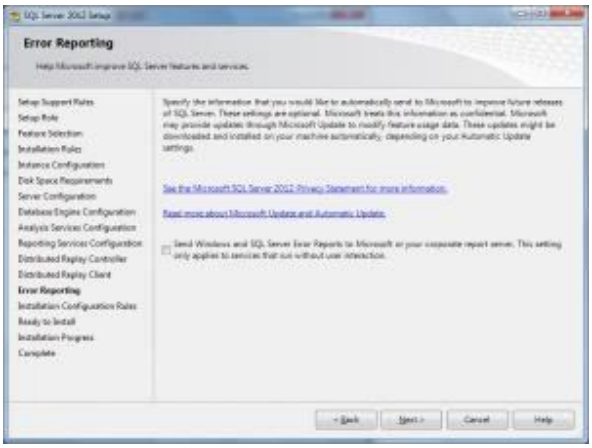
STEP 16:

Leave the Controller Name blank and Click Next to Continue (see image below).



STEP 17:

Check box if you would like to send error reports to Microsoft. I have left it as-is (see image below).



Click Next to Continue.

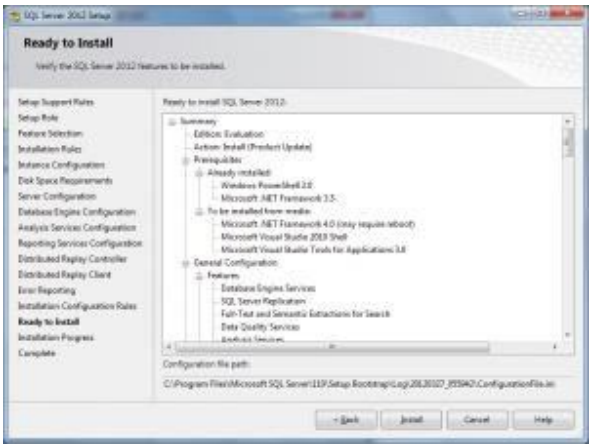
At this point, set up will again verify your system for any potential blocking issues.

Once the operation is completed, click Next to continue (see image below).

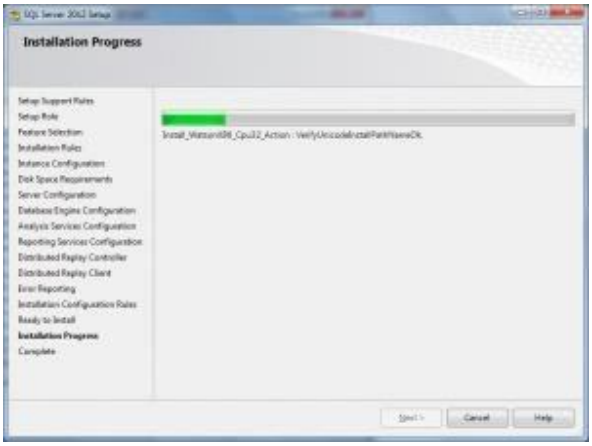
STEP 18:

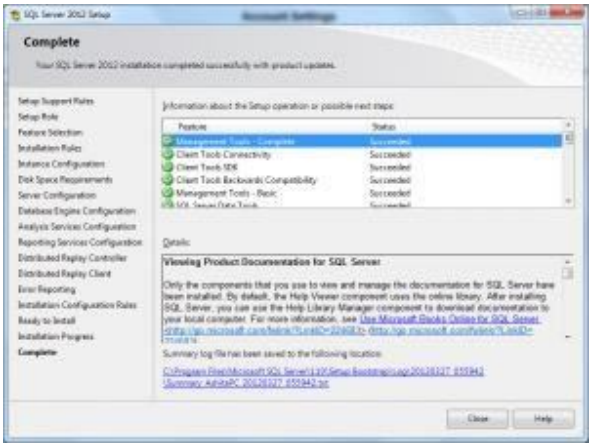
Review the options you selected for the install and verify the path to which configuration file will be stored. You can take advantage of this configuration file for future installs.

Click _Install‘ to begin Installation.



Installer will take a while in this step before the installation is complete.





4.8.4:Hardware and Operating Systems Requirements

Hardware Requirement

SERVER

Power 740 Express server

Business use- The Power® 740 Express® server offers large memory capacity, outstanding performance, and industrial-strength virtualization for smallto-midsize database servers and server consolidation.

UPS (UNINTERRUPTIBLE POWER SYSTEM)

HP T750J Uninterruptible Power System (AF456A)

Electrical input connection:	NEMA 5-15P
Electrical output connections:	(6) NEMA 5-15R
Communications serial ports:	Standard DB-9 port
Load segments:	1
Battery type:	Maintenance-free, sealed, valve-regulated lead acid (VRLA)
Hot-swap batteries:	Yes
Communications software:	HP Power Protector
Operating voltage:	100V, 110V

Software Requirement

Processor (Max) -	Sixteen 64-bit 3.55 GHz POWER7 cores
Memory (Min/Max) -	8 GB to 512 GB DDR3
Internal storage (Max) -	Up to 4.8 TB

4.8.5:How the Software will be used

The software will be installed on the hospital **server**and individual clients systems (User computers) will be connected to it. This method is adopted for centralized management, security reasons and for the sharing of common resources.

4.8.6:Training Requirements

The training requirements of the staff for the use of the software include:

1. A basic knowledge of Ms Office.
2. Knowledge of Ms Server 2012 administration for the safe installation and running of the software on the server.

3. Must have worked with the hospital for a minimum of 3 years to be sure the staff has basic knowledge of the workflow in the hospital environment.

4.8.7: Conversion Procedure

There are basically 3 types of conversion procedure (implementation). Below is a list of them:

1. Direct Changeover
2. Parallel running
3. Phased Implementation

Direct Changeover

This is one of the methods to change from an existing system to a new one. Although this is a simple method, it takes a lot of planning. All the files have to be ready to load all the files have to be ready to load, all the users of the system need to have been trained in advance and the system must have been properly tested. If the system doesn‘t work when switched on, the system will have to be shutdown the users can‘t go back to the old system.

Advantages

- Takes the minimal time and effort
- The new system is up and running immediately

Disadvantages

- If the new system fails, there is no back-up system, so data can be lost

Parallel Running

In this method of implementation, a new system and an existing system run side by side. To input the same data and perform the same processes, compare their output and prove the reliability of the new system if the new system is accepted, the existing system will stop running and will be replaced by the new one.

Advantages

- If anything goes wrong with the new system, the old system will act as a back-up.
- The outputs from the old and new systems can be compared to check that the new system is running correctly

Disadvantages

 Entering data into two systems, and running two systems together, takes a lot of extra time and effort

Phased Implementation

The phased implementation is a changeover that takes place in stages.

Advantages

- Allows users to gradually get used to the new system
- Staff training can be done in stages

Disadvantage

 If a part of the new system fails, there is no back-up system, so data can be lost

This project adopts the Phased Implementation for conversion form the manual to the new software. The phased implementation is done such that the each module in the new system is run and if it works well, the next module is run and so on. In the case where one of the module doesn‘t run properly, the module is taken out and the needed change effected and re-introduced to the system. This is done until the entire module of the new system is implemented.

4.8.8:Adoption Date

The new software will be fully adopted and commissioned, when the programmed is installed, the new system is tested module by module and all modules are verified to have worked properly the

staff who will use the system are all trained, and then the new software will be adopted by the hospital.

CHAPTER FIVE

5.0 SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

The project work was embarked upon with the major aim of reducing the administrative difficulties and difficulties staff and patients face in the General Hospital, Asokoro. The current manual system which involved a lot of stress, a labor intensive and time consuming process were a patient moves from one office to another for registration and where staffs(depending on the one handling an issue) will have to sit all day attending to patients one after the other was analyzed carefully and the problems were highlighted.

Working with the result obtained a new system was designed.

REFERENCES

Hern-Underwood M. J. and Workman(1993) *Group climate: a significant retention factor for pediatric nurse managers.*

O'Brien, J (1999). *Management Information Systems – Managing Information Technology in the Internetworked Enterprise*. Boston: Irwin McGraw-Hill.

Cholewka, P. A. (2006). *Implementation of a health care information system in Lithuania*

Brailer, D. J. (2005). *Economic perspective on health information technology*.

Wilcke, B. (2008). *Finding the knowledge in Information.*

Paul R. Vegoda(1987). *Introduction to Hospital Information Systems.*

Dujat C, Haux R, Schmucker P, Winter A.(1996). *Digital Optical Archiving of Medical Records in Hospital Information Systems- a practical approach towards the computer-based patient record*

Onwujekwe, O. (2005). *Inequities in healthcare seeking in treatment of communicable endemic diseases in Southeast Nigeria.*

Ouma, S., & Herselman, M. E. (2008). *E-health in rural areas: Case of developing countries.*

Pant, S., Hsu, C., (1995), *Strategic Information Systems Planning: A Review, Information*

Resources Management Association International Conference, May 21–24, Atlanta.

Serena, (2007), *An Introduction to Agile Software Development.*

Keenan, C. R., Nguyen, H. H., & Srinivasan, M. (2006). Electronic medical records and their impact on residents and medical student education.

R. Goe and M. Kenney. 1990. *"Should Information Be a Social or Economic Good in the*

Emerging Information Society." In Sven Lundstedt (ed.) Telecommunications, Values, and the Public Interest

Namita, K. Mansi, C., Avinash, C. *Project Report on Hospital Management System.*

Williams Laurie, (2007), *A Survey of Agile Development Methodologies.*

Nabimara Charles, (2007), *Computerized Health Records Management System*

Idowu, B., Adagunodo, R., & Adedoyin, R. (2006), *Information technology infusion model for health sector in developing country: Nigeria as a case.*

Grimm, N., & Shaw, N. (2007), *Using participatory action research to evaluate and improve change: Pilot implementation of electronic health records in rural Tanzania.*

Krishna, R., Kelleher, K., & Stahlberg, E. (2007). *Patient confidentiality in the research use of clinical medical databases.*

Gillies, A., & Howard, J. (2005). An international comparison of information in adverse events

Sisniega, L. C. (2009). *Barriers to electronic government use as perceived by citizens at the municipal level in Mexico.*

Svensson, P. (2002). *E-health applications in health care management.*

Stone, T., Patrick, T. B., & Brown, G. D. (2005). *Strategic Management of Information Systems in Healthcare*.

Morath, J. M., & Turnbull, J. E. (2005). *To do no harm: Ensuring patient safety in health care organizations*.

Cohan, P. S. (2005). *CFO's to tech: I'll spend for the right technology*.

Furukawa, M. F., Raghu, T. S., Spaulding, T. J., & Vinze, A. (2008). *Adoption of health information technology for medication safety in U.S. hospitals*.

Fuji, K. T., & Galt, K. A. (2008). *Pharmacists and health information technology: Emerging issues in patient safety*.

Harrison, J., & McDowell, G. (2008). *The role of laboratory information systems in healthcare quality improvement.*

Woodside, J. M. (2007). *EDI and ERP: A real-time framework for healthcare data exchange*.

Phillips, J. T. (2009). *Selecting software for managing physical & electronic records*.

Crane, J., & Crane, F. G. (2006). *Preventing medication errors in hospitals through a systems approach and technological innovation: A prescription for 2010*.

Rennolls, K., & AL-Shawabkeh, A. (2008). *Formal structures for data mining, knowledge discovery and communication in a knowledge management environment.*

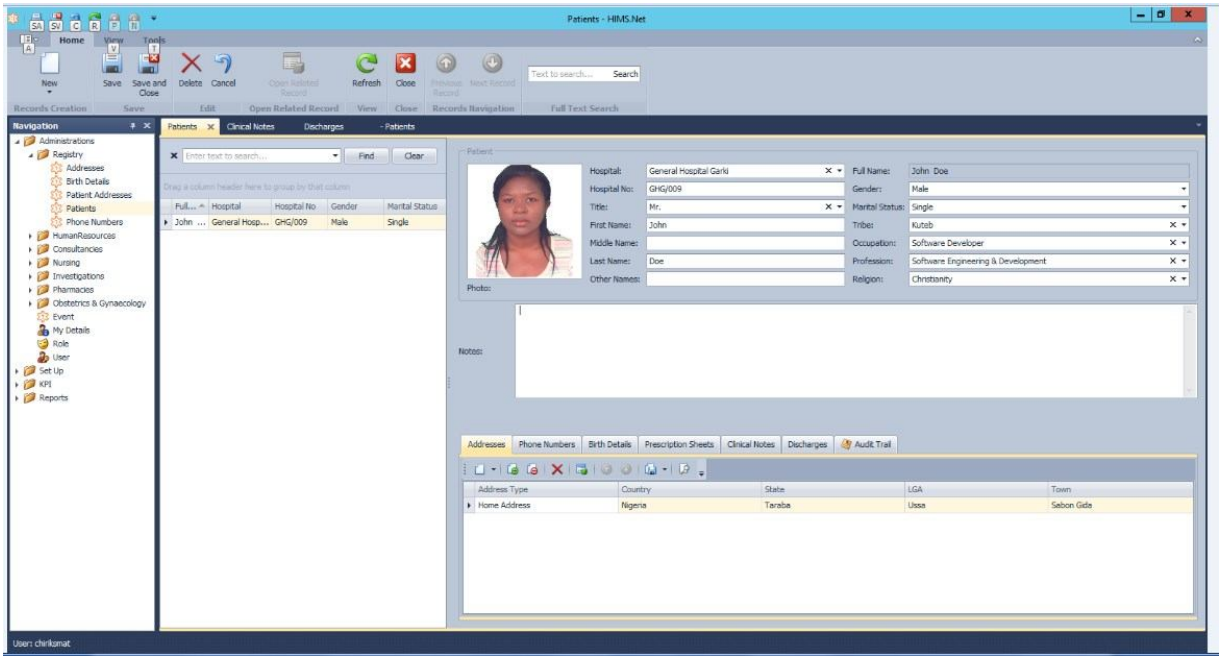
Linz, A. J., & Fallon L.F Jr.(2008). *Public perception regading the impact of electronic medical records on health care quality and medical errors. Journal of Controversial Medical Claims*.

Pineault et al.(1989). *Health Counselling in a Hospital.*

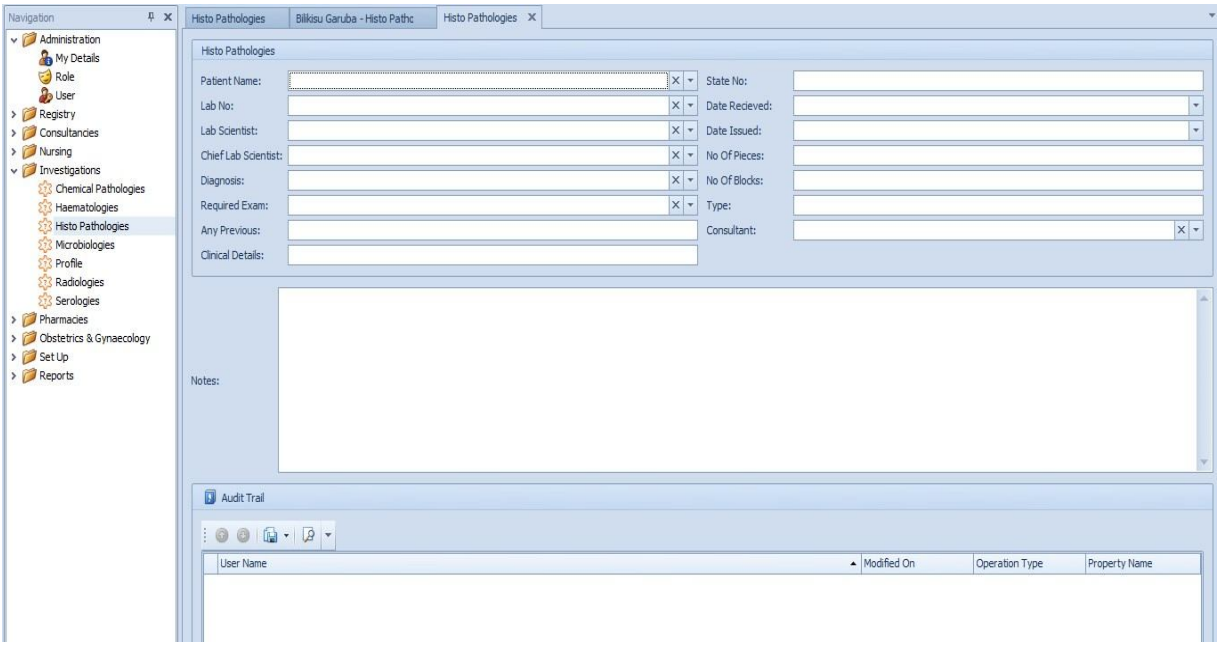
Appendix A

Sample Outputs Login Screen

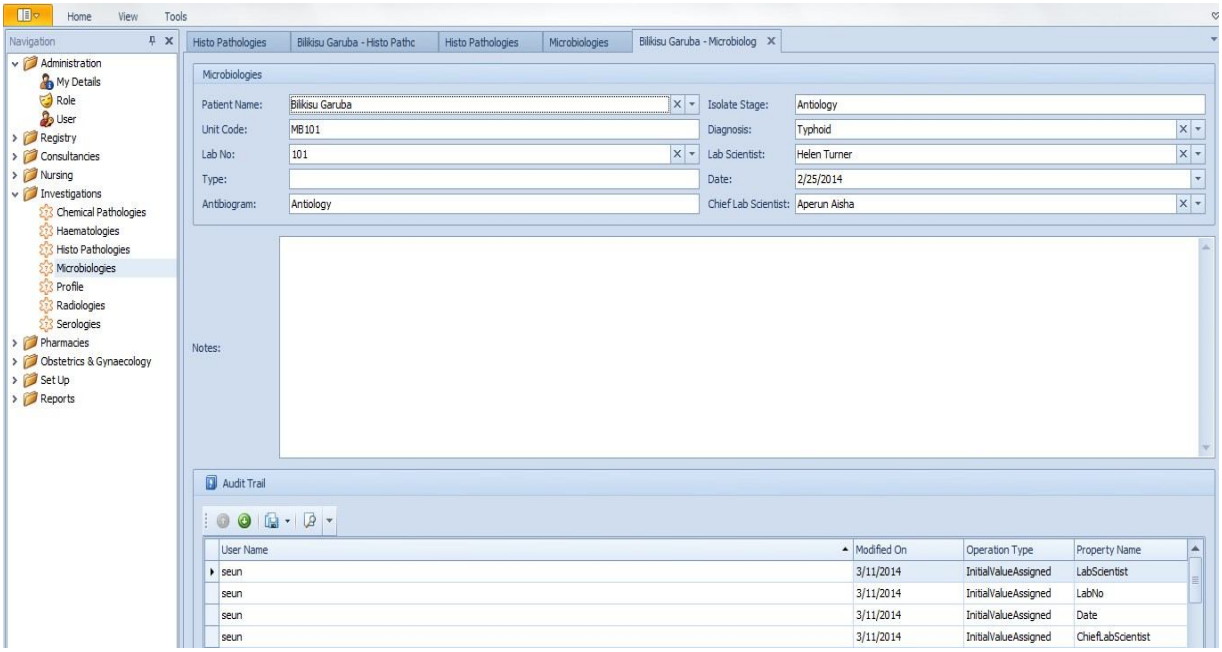
Registry/Records Screen



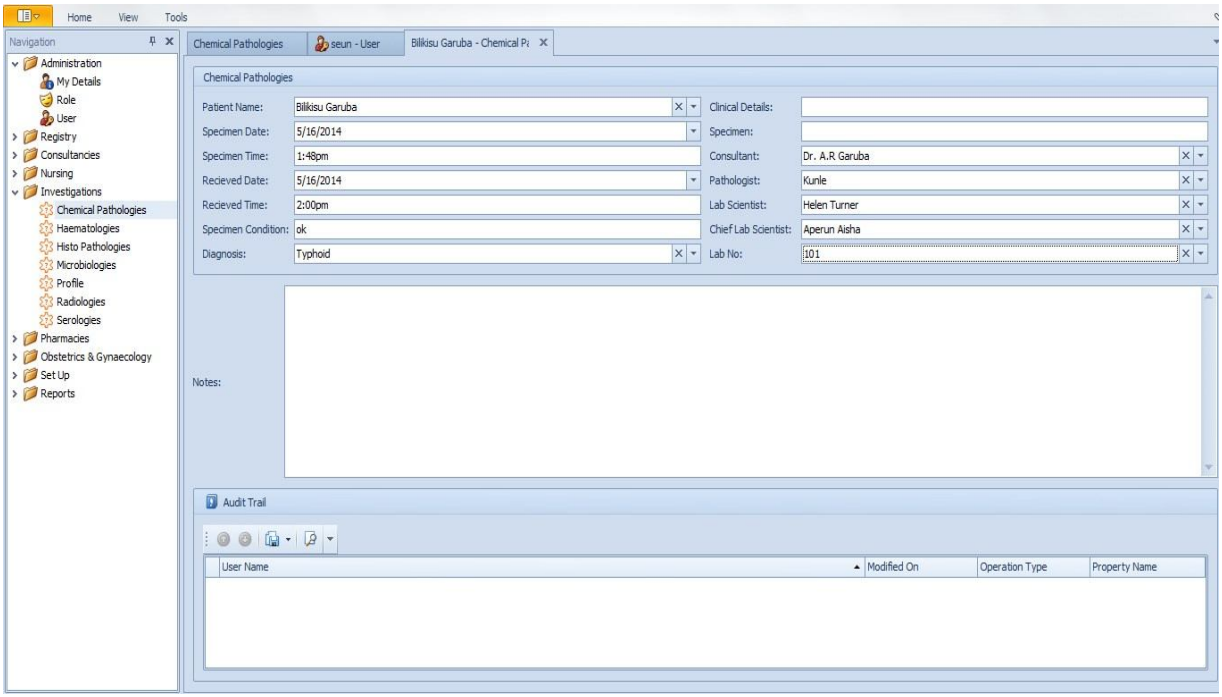
Investigations Histopathology Screen



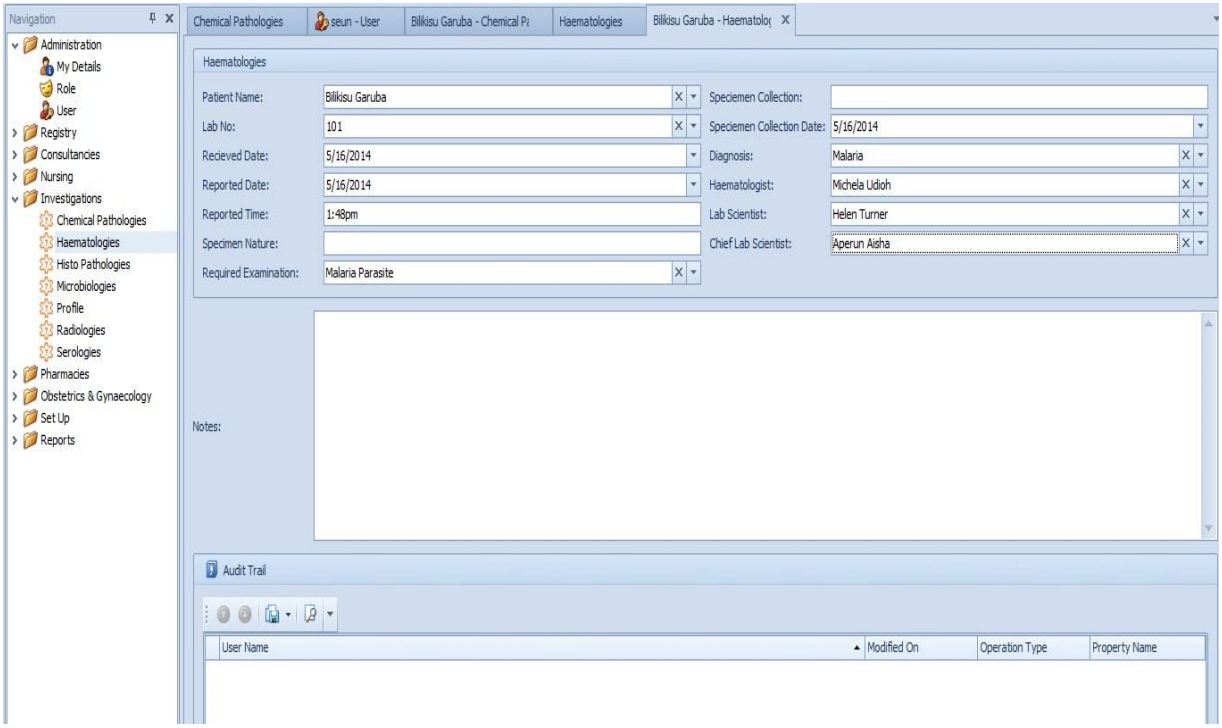
Microbiology



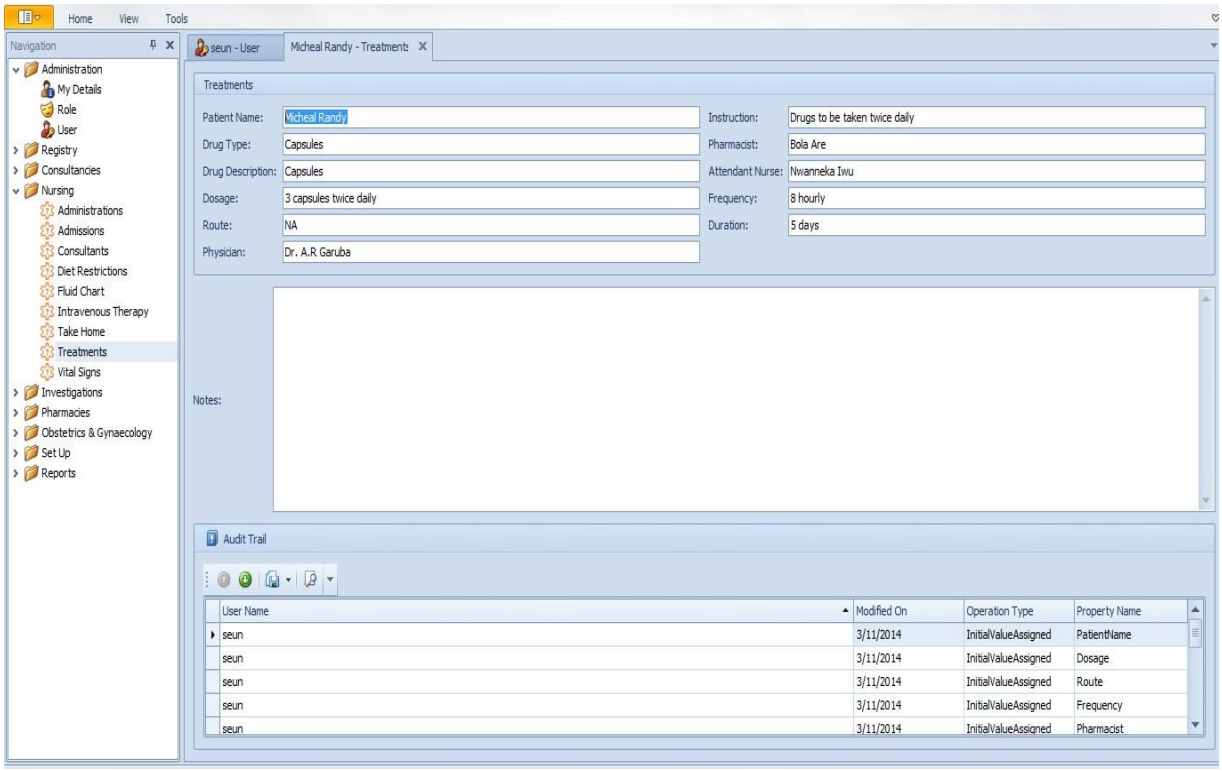
Chemical Pathology



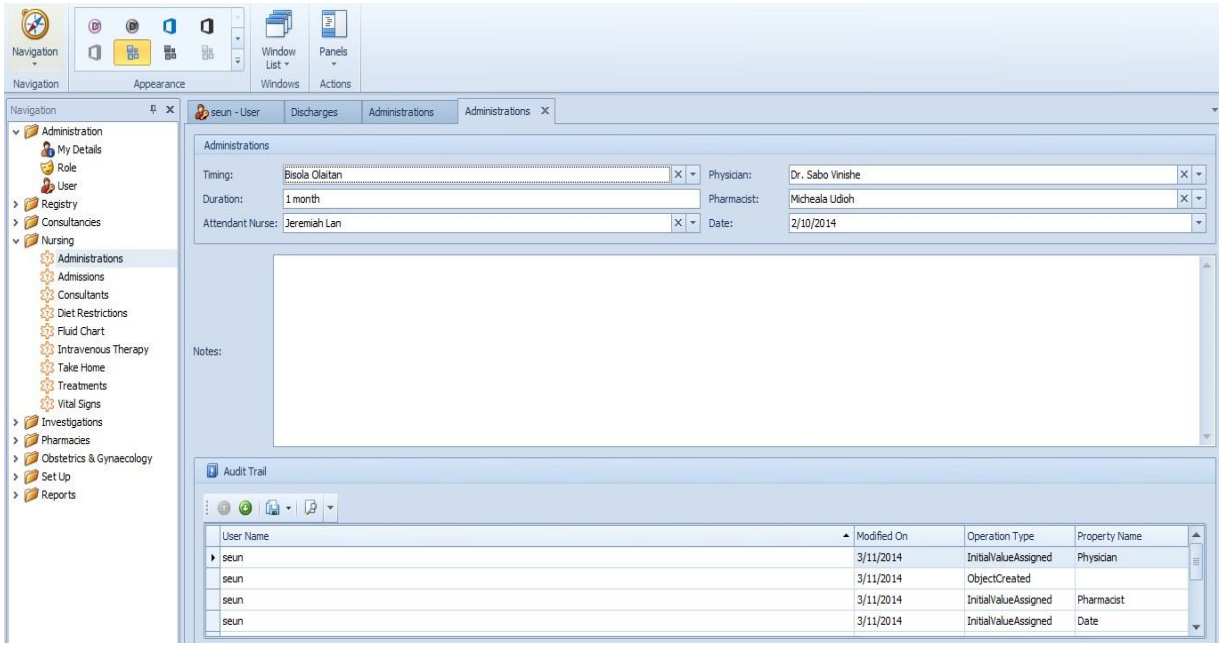
Haematology



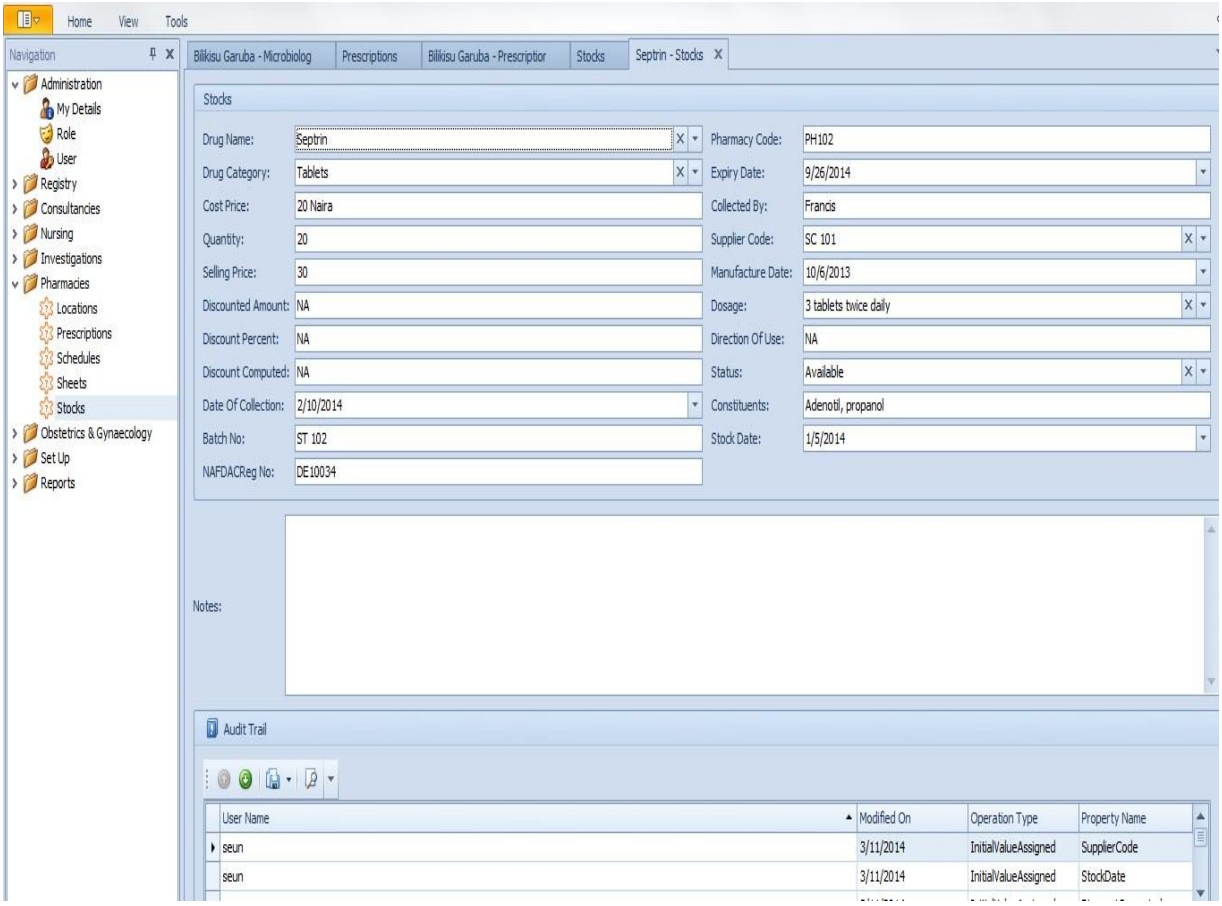
Nursing Treatments



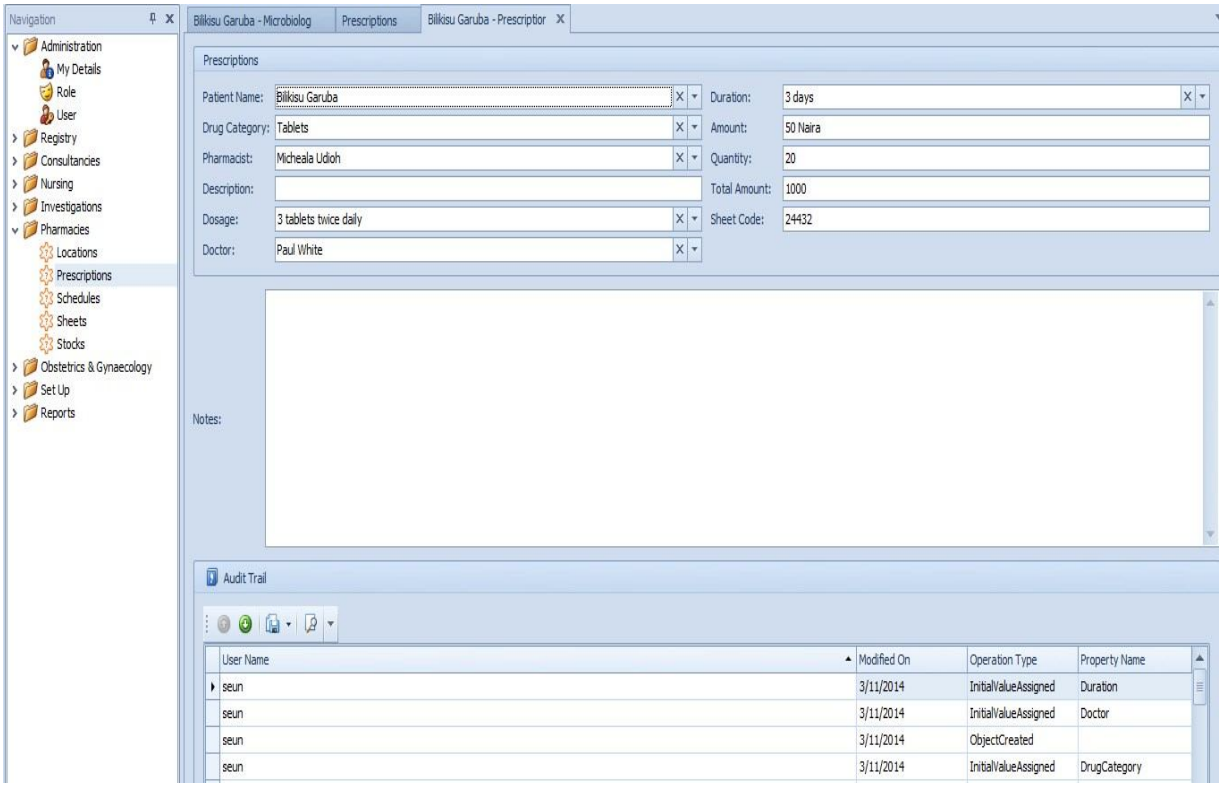
Administrations



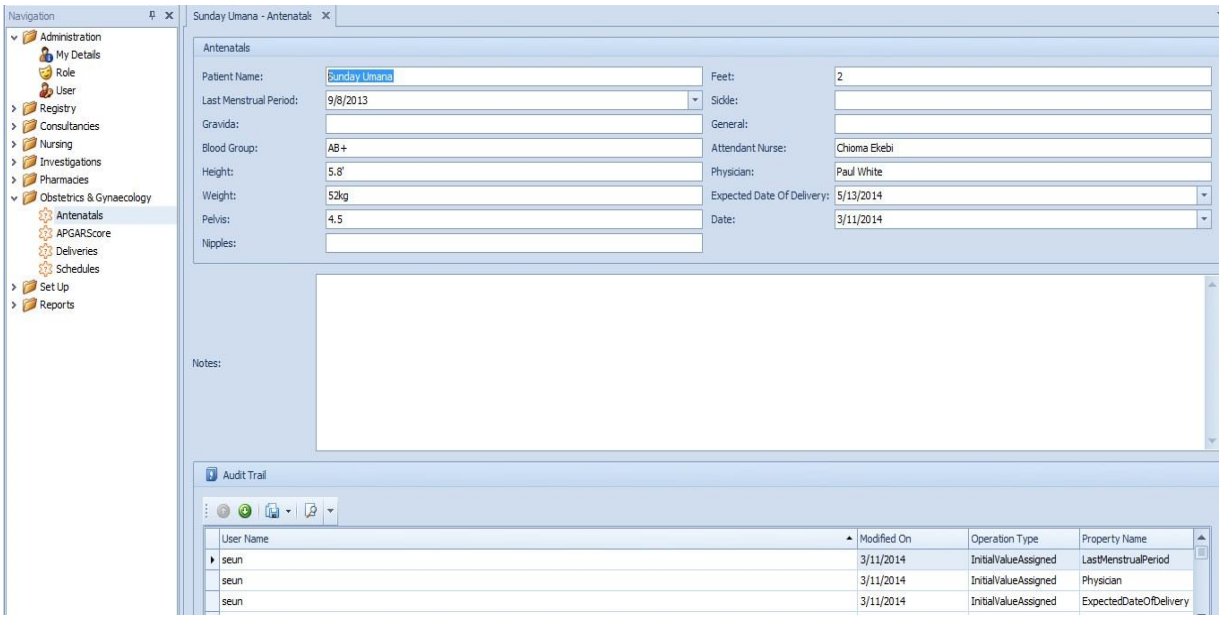
Pharmacy Stocks



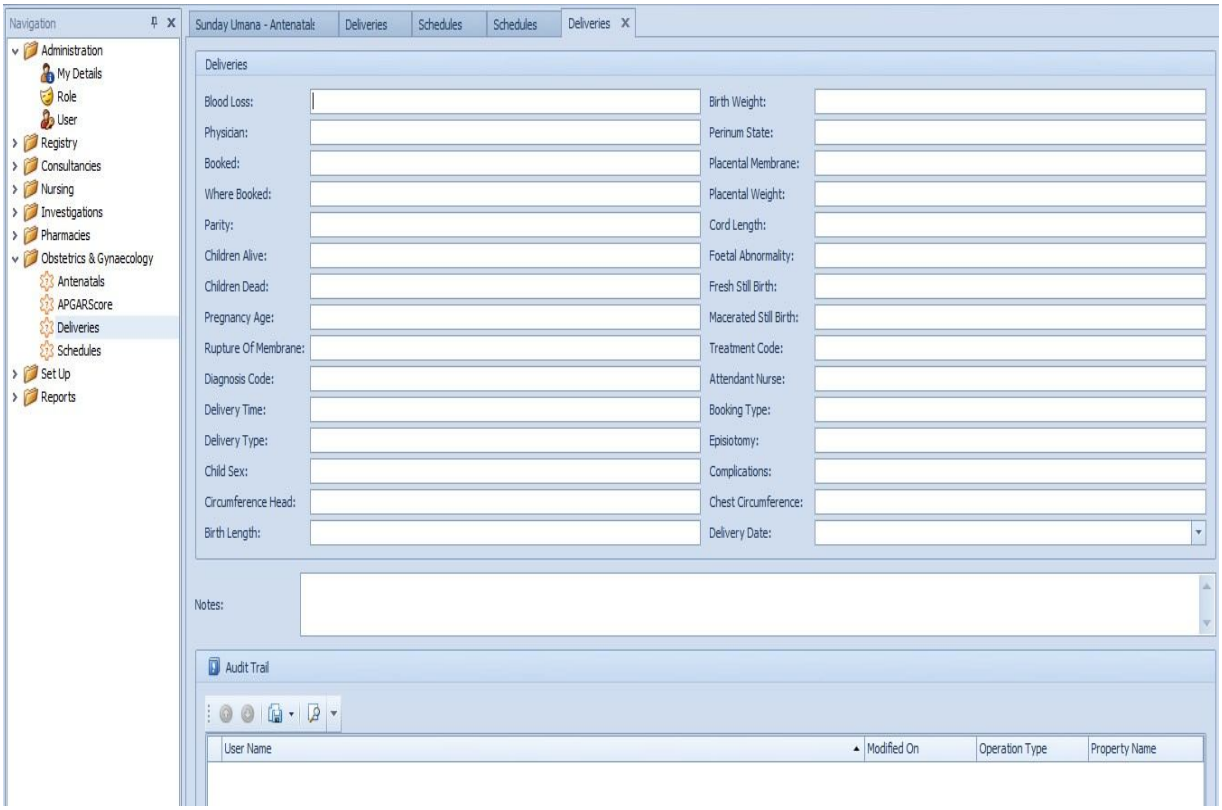
Prescriptions



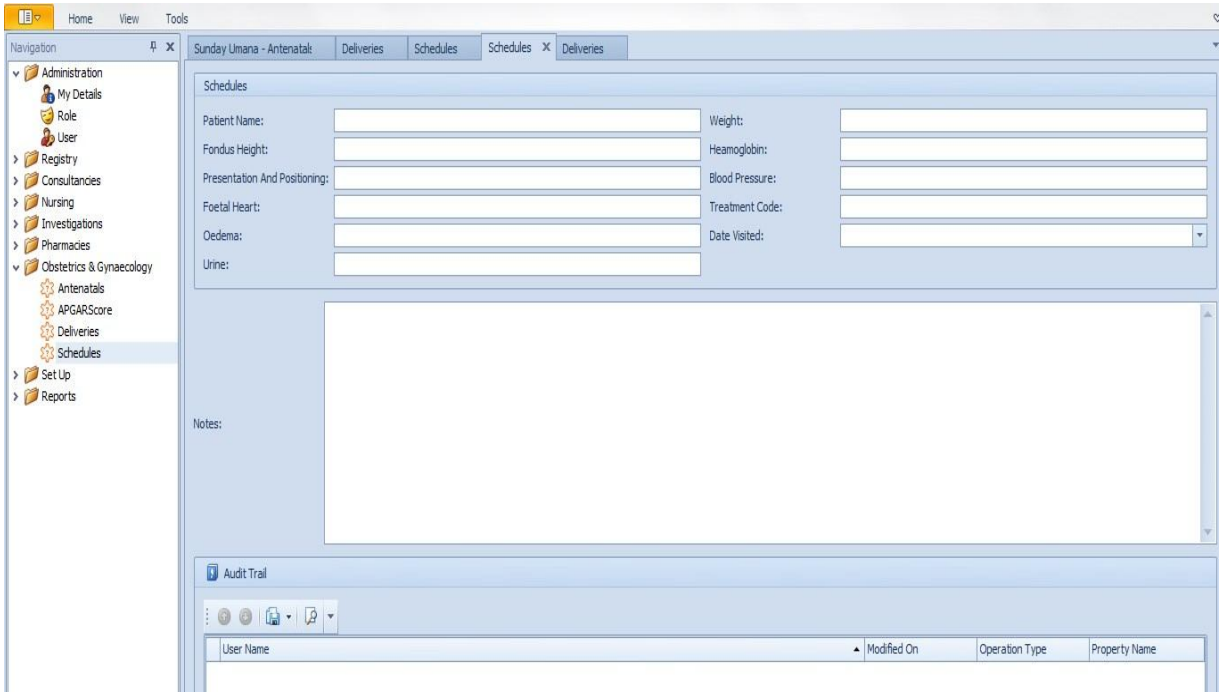
Obstetrics and Gynecology Antenatals



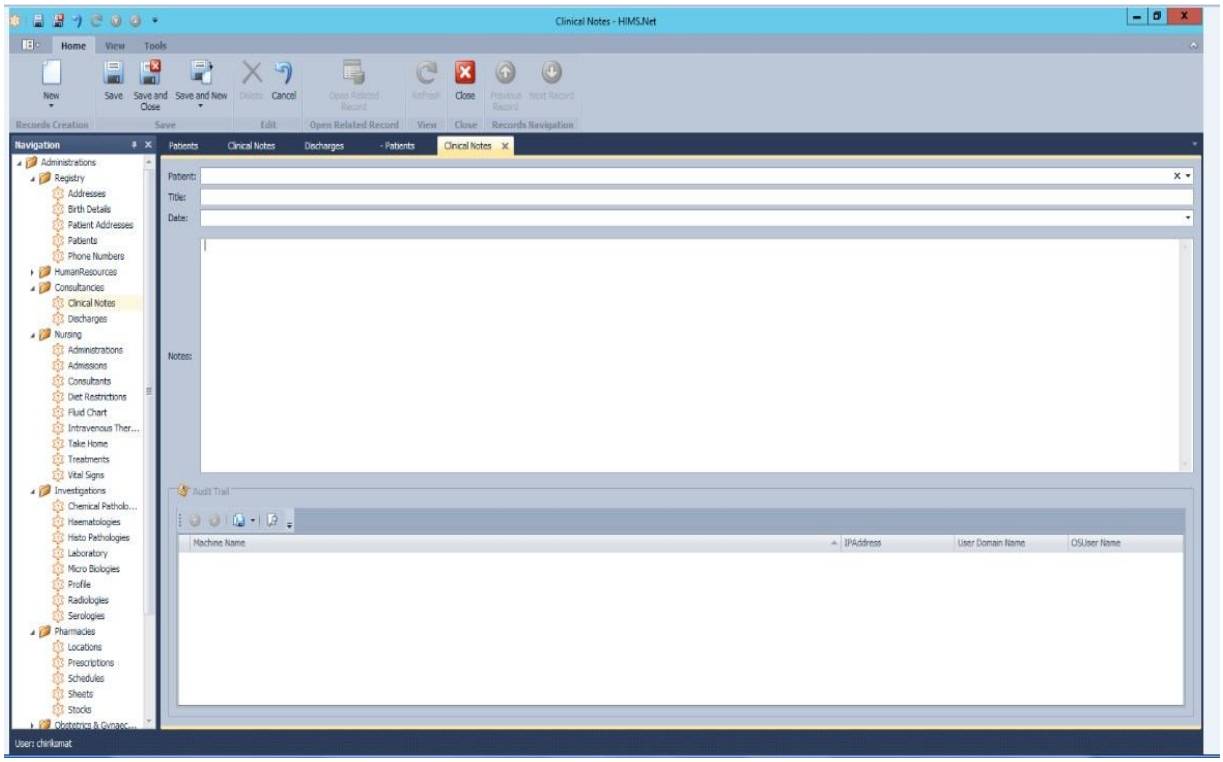
Deliveries



Schedules Screen



Consultancy Clinical Notes



Appendix B

Program Codes

Consultancy Clinical Notes

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp; using
HIMS.Module.Registry;

namespace HIMS.Module.Consultancy
{

    [DefaultClassOptions]
```

```
    [NavigationItem("Consultancies")]
public class ClinicalNotes : BaseObject
    {
        public ClinicalNotes(Session session)
            : base(session)
        {
        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();
        }

        // Fields...
        private Patients _Patient;
private DateTime _Date;         private
string _Title;                 private
string _Notes;
         private XPCollection<AuditDataItemPersistent> _auditTrail =
null;
        [Association("Patients-ClinicalNotes")]
public Patients  Patient
        {
get
{
                return _Patient;
            }
            set
            {
                SetPropertyValue("Patient", ref _Patient, value);
            }
}
        public string  Title
        {
get
{
                return _Title;
            }
set
            {
                SetPropertyValue("Title", ref _Title, value);
            }
        }
        [Size(4026)]
public string Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
        }
        public DateTime Date
        {
get
{
                return _Date;
            }
set
```

```
            {
                SetPropertyValue("Date", ref _Date, value);
            }
}
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
            if (_auditTrail == null)
                _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
            return _auditTrail;
            }

        }
    } }
```

Discharges

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp; using
HIMS.Module.Registry;

namespace HIMS.Module.Consultancy
{
    [DefaultClassOptions]
    [NavigationItem("Consultancies")]
public class Discharges : BaseObject
    {
        public Discharges(Session session)
            : base(session)
        {
}
        public override void AfterConstruction()
        {
            base.AfterConstruction();
        }         // Fields...         private
DischargeType _DischargeType;         private
Patients _Patient;         private DateTime
_Date;         private Doctor _DischargeDoctor;
private string _Witness;         private
ConsultantNurse _DischargeNurse;         private
string _Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;
        [Association("Patients-Discharges")]
public Patients Patient
        {
get
{
```

```
                return _Patient;
            }
set
            {
                SetPropertyValue("Patient", ref _Patient, value);
}
        }


        public DischargeType DischargeType
        {
            get
            {
                return _DischargeType;
            }
set
            {
                SetPropertyValue("DischargeType", ref _DischargeType, value);
            }
}
        public ConsultantNurse  DischargeNurse
        {
get
{
                return _DischargeNurse;
            }
set
            {
                SetPropertyValue("DischargeNurse", ref _DischargeNurse, value);
            }
        }
        public string  Witness
        {
get
{
                return _Witness;
            }
set
            {
                SetPropertyValue("Witness", ref _Witness, value);
            }
        }
        public Doctor  DischargeDoctor
        {
get
{
                return _DischargeDoctor;
            }
set
            {
                SetPropertyValue("DischargeDoctor", ref _DischargeDoctor, value);
}
        }

        public DateTime Date
        {
get
{
                return _Date;
            }
set
```

```
            {
                SetPropertyValue("Date", ref _Date, value);
            }
        }
        [Size(500)]
        public string Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
}
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }
        }
    } }
```

Investigations

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp;

namespace HIMS.Module.Investigations
{
    [NavigationItem("Investigations")]
    [DefaultClassOptions]
    public class ChemicalPathologies : BaseObject
    {
        public ChemicalPathologies(Session session)
            : base(session)
        {

        }
 public override void AfterConstruction() {
            base.AfterConstruction();

        }
```

```
        // Fields...         private
Diagnosis _Diagnosis;         private
PatientName _PatientName;         private
DateTime _SpecimenDate;         private
DateTime _RecievedDate;         private
LabNo _LabNo;
        private ChiefLabScientist _ChiefLabScientist;
private LabScientist _LabScientist;         private
Pathologist _Pathologist;         private Doctor
_Consultant;         private string _Specimen;
private string _ClinicalDetails;         private
string _SpecimenCondition;         private string
_RecievedTime;         private string _SpecimenTime;
private string _Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;

        public PatientName PatientName
        {
get
{
                return _PatientName;
            }
set
            {
                SetPropertyValue("PatientName", ref _PatientName, value);
            }
        }
        public DateTime SpecimenDate
        {
get
{
                return _SpecimenDate;
            }
set
            {
                SetPropertyValue("SpecimenDate", ref _SpecimenDate, value);
}
        }
        public string  SpecimenTime
        {
get
{
                return _SpecimenTime;
            }
set
            {
                SetPropertyValue("SpecimenTime", ref _SpecimenTime, value);
}
        }

        public DateTime  RecievedDate
        {
get
{
                return _RecievedDate;
            }
set
            {
                SetPropertyValue("RecievedDate", ref _RecievedDate, value);
            }
```

```
        }
        public string  RecievedTime
        {
get
{
                return _RecievedTime;
            }
set
            {
                SetPropertyValue("RecievedTime", ref _RecievedTime, value);
            }
        }
        public string  SpecimenCondition
        {
get
{
                return _SpecimenCondition;
            }
set
            {
                SetPropertyValue("SpecimenCondition", ref _SpecimenCondition, value);
            }
        }

        public Diagnosis Diagnosis
        {
get
{
                return _Diagnosis;
            }
set
            {
                SetPropertyValue("Diagnosis", ref _Diagnosis, value);
            }
        }          public string
ClinicalDetails
        {
get
{
                return _ClinicalDetails;
            }
set {
                SetPropertyValue("ClinicalDetails", ref _ClinicalDetails, value);
            }
        }
        public string  Specimen
        {
get
{
                return _Specimen;
            }
set
            {
                SetPropertyValue("Specimen", ref _Specimen, value);
            }
        }
        public Doctor Consultant
        {
get
{
                return _Consultant;
```

```
            }
set
            {
                SetPropertyValue("Consultant", ref _Consultant, value);
            }
        }
        public Pathologist Pathologist
        {
get
{
            return _Pathologist;
            }
set
            {
                SetPropertyValue("Pathologist", ref _Pathologist, value);
            }
        }          public LabScientist
LabScientist
        {
get
{
            return _LabScientist;
            }
set
            {
                SetPropertyValue("LabScientist", ref _LabScientist, value);
}
        }
        public ChiefLabScientist ChiefLabScientist
        {
get
{
            return _ChiefLabScientist;
            }
            set
            {
                SetPropertyValue("ChiefLabScientist", ref _ChiefLabScientist, value);
            }
        }
        public LabNo LabNo
        {
get
{
            return _LabNo;
            }
set
            {
                SetPropertyValue("LabNo", ref _LabNo, value);
            }
        }
[Size(500)]         public
string Notes
        {
get
{
            return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
```

```
        }
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }

        }

    } }
```

Heamatology

```
using System; using System.Linq;
using System.Text; using
DevExpress.Xpo; using
DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base;
using
System.Collections.Generic;
using
DevExpress.ExpressApp.Model;
using
DevExpress.Persistent.BaseImpl;
using
DevExpress.Persistent.Validation
; using HIMS.Module.SetUp;

namespace HIMS.Module.Investigations
{
    [NavigationItem("Investigations")]
    [DefaultClassOptions]
    public class Haematologies : BaseObject
    {
        public Haematologies(Session session)
            : base(session)
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...         private Diagnosis
_Diagnosis;         private PatientName _PatientName;
private DateTime _RecievedDate;         private DateTime
_ReportedDate;         private DateTime
_SpeciemenCollectionDate;         private
ChiefLabScientist _ChiefLabScientist;         private
LabScientist _LabScientist;         private Haematologist
_Haematologist;         private string
```

```
_SpeciemenCollection;         private RequiredExamination
_RequiredExamination;         private string
_SpecimenNature;         private string _ReportedTime;
private LabNo _LabNo;         private string _Notes;
         private XPCollection<AuditDataItemPersistent> _auditTrail =
null;

        public PatientName PatientName
        {
get
{
                return _PatientName;
            }
set
            {
                SetPropertyValue("PatientName", ref _PatientName, value);
}
        }          public
LabNo LabNo
        {
            get
            {
                return _LabNo;
            }
set
            {
                SetPropertyValue("LabNo", ref _LabNo, value);
            }
        }

        public DateTime RecievedDate
        {
get
{
                return _RecievedDate;
            }
set
            {
                SetPropertyValue("RecievedDate", ref _RecievedDate, value);
            }
        }

        public DateTime ReportedDate
        {
get
{
                return _ReportedDate;
            }
set
            {
                SetPropertyValue("ReportedDate", ref _ReportedDate, value);
            }
        }
        public string ReportedTime
        {
get
{
                return _ReportedTime;
            }
set
            {
```

```
                SetPropertyValue("ReportedTime", ref _ReportedTime, value);
            }
        }          public string
SpecimenNature
        {
get
{
                return _SpecimenNature;
            }
set
            {
                SetPropertyValue("SpecimenNature", ref _SpecimenNature, value);
}
        }
        public RequiredExamination RequiredExamination
        {
get
{
                return _RequiredExamination;
            }
set
            {
                SetPropertyValue("RequiredExamination", ref _RequiredExamination, value);
            }
        }
        public string SpeciemenCollection
        {
get
{
                return _SpeciemenCollection;
            }
set
            {
                SetPropertyValue("SpeciemenCollection", ref _SpeciemenCollection, value);
            }
        }

        public DateTime SpeciemenCollectionDate
        {
get
{
                return _SpeciemenCollectionDate;
            }
set
            {
                SetPropertyValue("SpeciemenCollectionDate", ref _SpeciemenCollectionDate,
value);            }
        }
          public Diagnosis
Diagnosis
        {
get
{
                return _Diagnosis;
            }
set
            {
                SetPropertyValue("Diagnosis", ref _Diagnosis, value);
            }
        }
        public Haematologist Haematologist
```

```
        {
get
{
                return _Haematologist;
            }
            set
            {
                SetPropertyValue("Haematologist", ref _Haematologist, value);
            }
        }
        public LabScientist LabScientist
        {
get
{
                return _LabScientist;
            }
set
            {
                SetPropertyValue("LabScientist", ref _LabScientist, value);
            }
        }
        public ChiefLabScientist ChiefLabScientist
        {
get
{
                return _ChiefLabScientist;
            }
set
            {
                SetPropertyValue("ChiefLabScientist", ref _ChiefLabScientist, value);
            }
        }
[Size(500)]         public
string Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
}
        }          public XPCollection<AuditDataItemPersistent>
AuditTrail
        {
get
{
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }

        }

    } }
```

Histopathology

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp;

namespace HIMS.Module.Investigations
{
    [NavigationItem("Investigations")]
    [DefaultClassOptions]
    public class HistoPathologies : BaseObject
    {
        public HistoPathologies(Session session)
            : base(session)
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...         private
ChiefLabScientist _ChiefLabScientist;         private
LabScientist _LabScientist;         private LabNo
_LabNo;         private PatientName _PatientName;
private DateTime _DateRecieved;         private
DateTime _DateIssued;         private Doctor
_Consultant;         private string _Type;
private string _NoOfBlocks;         private string
_NoOfPieces;         private string _StateNo;
private string _ClinicalDetails;         private
string _AnyPrevious;         private
RequiredExamination _RequiredExam;         private
Diagnosis _Diagnosis;         private string _Notes;
         private XPCollection<AuditDataItemPersistent> _auditTrail =
null;

        public PatientName PatientName
        {
get
{
                return _PatientName;
            }
set
            {
                SetPropertyValue("PatientName", ref _PatientName, value);
            }
        }
```

```
        public LabNo LabNo
        {
get
{
                return _LabNo;
            }
set
            {
                SetPropertyValue("LabNo", ref _LabNo, value);
            }
        }

        public LabScientist LabScientist
        {
get
{
                return _LabScientist;
            }
set
            {
                SetPropertyValue("LabScientist", ref _LabScientist, value);
            }
        }

        public ChiefLabScientist ChiefLabScientist
        {
get
{
                return _ChiefLabScientist;
            }
set
            {
                SetPropertyValue("ChiefLabScientist", ref _ChiefLabScientist, value);
            }
        }          public
Diagnosis Diagnosis
        {
get
{
                return _Diagnosis;
            }
set             {
SetPropertyValue(
"Diagnosis", ref
_Diagnosis,
value);
            }
        }
        public RequiredExamination RequiredExam
        {
get
{
                return _RequiredExam;
            }
set
            {
                SetPropertyValue("RequiredExam", ref _RequiredExam, value);
            }
        }
        public string AnyPrevious
```

```
        {
get
{
                return _AnyPrevious;
            }
set
            {
                SetPropertyValue("AnyPrevious", ref _AnyPrevious, value);
            }
        }
        public string ClinicalDetails
        {
get
{
                return _ClinicalDetails;
            }
set
            {
                SetPropertyValue("ClinicalDetails", ref _ClinicalDetails, value);
            }
        }
        public string StateNo
        {
get
{
                return _StateNo;
            }
set
            {
                SetPropertyValue("StateNo", ref _StateNo, value);
}
        }

        public DateTime DateRecieved
        {
get
{
                return _DateRecieved;
            }
set
            {
                SetPropertyValue("DateRecieved", ref _DateRecieved, value);
            }
        }

        public DateTime DateIssued
        {
get
{
                return _DateIssued;
            }
set
            {
                SetPropertyValue("DateIssued", ref _DateIssued, value);
            }
        }
        public string  NoOfPieces
        {
get
{
                return _NoOfPieces;
```

```
            }
set
            {
                SetPropertyValue("NoOfPieces", ref _NoOfPieces, value);
            }
        }
        public string  NoOfBlocks
        {
get
{
            return _NoOfBlocks;
            }
set
            {
                SetPropertyValue("NoOfBlocks", ref _NoOfBlocks, value);
}
        }
        public string  Type
        {
get
{
            return _Type;
            }
set
            {
                SetPropertyValue("Type", ref _Type, value);
            }
        }
        public Doctor  Consultant
        {
            get
{
            return _Consultant;
            }
set
            {
                SetPropertyValue("Consultant", ref _Consultant, value);
            }
        }
[Size(500)]         public
string Notes
        {
get
{
            return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
        }
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
            if (_auditTrail == null)
                _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
            return _auditTrail;
            }

        }
```

```
    } }
```

Microbiology

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp;

namespace HIMS.Module.Investigations {
    [NavigationItem("Investigations")]
    [DefaultClassOptions]
    public class Microbiologies : BaseObject
    {
        public Microbiologies(Session session)
            : base(session)
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...
        private Diagnosis _Diagnosis;
private PatientName _PatientName;
private DateTime _Date;
        private ChiefLabScientist _ChiefLabScientist;
private LabScientist _LabScientist;         private
string _IsolateStage;         private string
_Antibiogram;         private string _Type;
private LabNo _LabNo;         private string
_UnitCode;         private string _Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;

        public PatientName PatientName
        {
get
{
                return _PatientName;
            }
set
            {
                SetPropertyValue("PatientName", ref _PatientName, value);
}
        }
        public string  UnitCode
```

```
        {
get
{
                return _UnitCode;
            }
set
            {
                SetPropertyValue("UnitCode", ref _UnitCode, value);
            }
        }
        public LabNo LabNo
        {
get
{
                return _LabNo;
            }
set
            {
                SetPropertyValue("LabNo", ref _LabNo, value);
            }
        }
        public string  Type
        {
get
{
                return _Type;
            }
set
            {
                SetPropertyValue("Type", ref _Type, value);
            }
        }
        public string  Antibiogram
        {
get
{
                return _Antibiogram;
            }
set
            {
                SetPropertyValue("Antibiogram", ref _Antibiogram, value);
            }
        }
        public string  IsolateStage
        {
get
{
                return _IsolateStage;
            }
set
            {
                SetPropertyValue("IsolateStage", ref _IsolateStage, value);
}
        }

        public Diagnosis Diagnosis
        {
get
{
                return _Diagnosis;
```

```
            }
set
            {
                SetPropertyValue("Diagnosis", ref _Diagnosis, value);
            }
        }

        public LabScientist LabScientist
        {
get
{
            return _LabScientist;
            }
set
            {
                SetPropertyValue("LabScientist", ref _LabScientist, value);
            }
        }

        public DateTime  Date
        {
get
{
            return _Date;
            }
set
            {
                SetPropertyValue("Date", ref _Date, value);
            }
        }
        public ChiefLabScientist ChiefLabScientist
        {
get
{
            return _ChiefLabScientist;
            }
set
            {
                SetPropertyValue("ChiefLabScientist", ref _ChiefLabScientist, value);
            }
        }
[Size(500)]         public
string Notes
        {
get
{
            return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
}
        }          public XPCollection<AuditDataItemPersistent>
AuditTrail
        {
get
{
            if (_auditTrail == null)
                _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
            return _auditTrail;
            }
```

```
        }
    } }
```

Administrations

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp;

namespace HIMS.Module.Nursing
{
    [NavigationItem("Nursing")]
[DefaultClassOptions]
    public class Administrations : BaseObject
    {
        public Administrations(Session session)
            : base(session)
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...         private DateTime
_Date;         private Pharmacist _Pharmacist;
private Doctor _Physician;         private
ConsultantNurse _AttendantNurse;         private
string _Duration;         private Timing
_Timing;         private string _Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;
         public Timing
Timing
        {
get
{
return _Timing;
            }
set
            {
                SetPropertyValue("Timing", ref _Timing, value);
            }
        }
        public string  Duration
```

```
        {
get
{
                return _Duration;
            }
set
            {
                SetPropertyValue("Duration", ref _Duration, value);
            }
        }
        public ConsultantNurse AttendantNurse
        {
get
{
                return _AttendantNurse;
            }
set
            {
                SetPropertyValue("AttendantNurse", ref _AttendantNurse, value);
            }
        }
        public Doctor Physician
        {
get
{
                return _Physician;
            }
set
            {
                SetPropertyValue("Physician", ref _Physician, value);
            }
        }
        public Pharmacist Pharmacist
        {
get
{
                return _Pharmacist;
            }
set
            {
                SetPropertyValue("Pharmacist", ref _Pharmacist, value);
}
        }

        [Size(500)]
public string Notes
{             get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
        }

        public DateTime  Date
        {
get
{
                return _Date;
```

```
            }
set
            {
                SetPropertyValue("Date", ref _Date, value);
            }
        }
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
            if (_auditTrail == null)
                _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
            return _auditTrail;
        }

        }

    } }
```

Admissions

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp;


namespace HIMS.Module.Nursing
{

    [DefaultClassOptions]
[NavigationItem("Nursing")]     public
class Admissions : BaseObject
    {
        public Admissions(Session session)
            : base(session)
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }
        // Fields...
private DateTime _Date;
private string _Type;
private string _Notes;
private string _BedNo;
private Doctor _Physician;
```

```
        private ConsultantNurse _AttendantNurse;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;

        public ConsultantNurse AttendantNurse
        {
get
{
                return _AttendantNurse;
            }
set
            {
                SetPropertyValue("AttendantNurse", ref _AttendantNurse, value);
            }
        }          public Doctor
Physician
        {
get
{
                return _Physician;
            }
set
            {
                SetPropertyValue("Physician", ref _Physician, value);
            }
        }
        public string  BedNo
        {
get
{
                return _BedNo;
            }
set
            {
                SetPropertyValue("BedNo", ref _BedNo, value);
            }
        }
        public string Type
        {
get
{
                return _Type;
            }
set
            {
                SetPropertyValue("Type", ref _Type, value);
            }
        }
[Size(500)]         public
string  Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
        }
```

```
        public DateTime  Date
        {
get
{
                return _Date;
            }
set
            {
                SetPropertyValue("Date", ref _Date, value);
}
        }
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }

        }
    } }
```

Treatments

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;

namespace HIMS.Module.Nursing
{

    [DefaultClassOptions]
[NavigationItem("Nursing")]     public
class Treatments : BaseObject
    {
        public Treatments(Session session)
            : base(session)
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...         private
string _PatientName;         private
string _Duration;         private string
```

```
_Frequency;         private string
_AttendantNurse;         private string
_Pharmacist;         private string
_Instruction;         private string
_Physician;         private string
_Route;         private string _Dosage;
private string _DrugDescription;
private string _DrugType;
private string _Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;

        public string  PatientName
        {
get
{
                return _PatientName;
            }
set
            {
                SetPropertyValue("PatientName", ref _PatientName, value);
            }
        }
        public string  DrugType
        {
get
{
                return _DrugType;
            }
set
            {
                SetPropertyValue("DrugType", ref _DrugType, value);
            }
        }
        public string  DrugDescription
        {
get
{
                return _DrugDescription;
            }
set
            {
                SetPropertyValue("DrugDescription", ref _DrugDescription, value);
            }
        }
        public string  Dosage
        {
get
{
                return _Dosage;
            }
set
            {
                SetPropertyValue("Dosage", ref _Dosage, value);
}
        }          public
string  Route
        {
get
{
                return _Route;
```

```
            }
set
            {
                SetPropertyValue("Route", ref _Route, value);
            }
        }

        public string  Physician
        {
get
{
            return _Physician;
            }
set
            {
                SetPropertyValue("Physician", ref _Physician, value);
            }
        }
        public string  Instruction
        {
get
{
            return _Instruction;
            }
set
            {
                SetPropertyValue("Instruction", ref _Instruction, value);
            }
        }
        public string  Pharmacist
        {
get
{
            return _Pharmacist;
            }
set
            {
                SetPropertyValue("Pharmacist", ref _Pharmacist, value);
            }
        }
        public string  AttendantNurse
        {
get
{
            return _AttendantNurse;
            }
set
            {
                SetPropertyValue("AttendantNurse", ref _AttendantNurse, value);
}
        }          public string
Frequency
        {
get
{
            return _Frequency;
            }
set
            {
                SetPropertyValue("Frequency", ref _Frequency, value);
            }
```

```
        }
        public string  Duration
        {
get
{
                return _Duration;
            }
set
            {
                SetPropertyValue("Duration", ref _Duration, value);
            }
        }
[Size(500)]         public
string Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
}
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }

        }
    } }
```

Obstetrics and Gynecology Antenatals

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;

namespace HIMS.Module.OandG
{

    [DefaultClassOptions]
[NavigationItem("OandG")]     public
class Antenatals : BaseObject
    {
        public Antenatals(Session session)
            : base(session)
```

```
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...         private string
_PatientName;         private DateTime
_LastMenstrualPeriod;         private DateTime
_ExpectedDateOfDelivery;         private DateTime
_Date;         private string _Physician;
private string _AttendantNurse;         private
string _General;         private string _Sickle;
private string _Feet;         private string
_Nipples;         private string _Pelvis;
private string _Weight;         private string
_Height;         private string _BloodGroup;
private string _Gravida;         private string
_Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;

        public string  PatientName
        {
get
{
                return _PatientName;
            }
set
            {
                SetPropertyValue("PatientName", ref _PatientName, value);
}
        }          public DateTime
LastMenstrualPeriod
        {
get
{
                return _LastMenstrualPeriod;
            }
set
            {
                SetPropertyValue("LastMenstrualPeriod", ref _LastMenstrualPeriod, value);
            }
        }
        public string Gravida
        {
get
{
                return _Gravida;
            }
set
            {
                SetPropertyValue("Gravida", ref _Gravida, value);
            }
        }
        public string BloodGroup
```

```
        {
get
{
                return _BloodGroup;
            }
set
            {
                SetPropertyValue("BloodGroup", ref _BloodGroup, value);
            }
        }
        public string Height
        {
get
{
                return _Height;
            }
set
            {
                SetPropertyValue("Height", ref _Height, value);
            }
        }
        public string Weight
        {
get
{
                return _Weight;
            }
set
            {
                SetPropertyValue("Weight", ref _Weight, value);
}
        }          public
string Pelvis
        {
get
{
                return _Pelvis;
            }
            set
            {
                SetPropertyValue("Pelvis", ref _Pelvis, value);
            }
        }
        public string Nipples
        {
get
{
                return _Nipples;
            }
set
            {
                SetPropertyValue("Nipples", ref _Nipples, value);
            }
        }
        public string Feet
        {
get
{
                return _Feet;
            }
set
```

```
            {
                SetPropertyValue("Feet", ref _Feet, value);
            }
        }
        public string Sickle
        {
get
{
            return _Sickle;
        }
set
            {
                SetPropertyValue("Sickle", ref _Sickle, value);
            }
        }
        public string General
        {
get
{
            return _General;
        }
set
            {
                SetPropertyValue("General", ref _General, value);
            }
        }          public string
AttendantNurse
        {
get
            {
                return _AttendantNurse;
            }
set
            {
                SetPropertyValue("AttendantNurse", ref _AttendantNurse, value);
            }
        }
        public string Physician
        {
get
{
            return _Physician;
        }
set
            {
                SetPropertyValue("Physician", ref _Physician, value);
            }
        }

        public DateTime ExpectedDateOfDelivery
        {
get
{
            return _ExpectedDateOfDelivery;
        }
set
            {
                SetPropertyValue("ExpectedDateOfDelivery", ref _ExpectedDateOfDelivery,
value);             }
        }
```

```
        [Size(500)]
public string Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
}
        }
        public DateTime  Date
        {
get
{
                return _Date;
            }
set
            {
                SetPropertyValue("Date", ref _Date, value);
}
        }
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }

        }
    } }
```

Antenatals

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;

namespace HIMS.Module.OandG
{

    [DefaultClassOptions]
[NavigationItem("OandG")]
    public class Deliveries : BaseObject
    {
        public Deliveries(Session session)
            : base(session)
        {
```

```
        }          public override void
AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...
        private DateTime _DeliveryDate;
private string _ChestCircumference;
private string _Complications;
private string _Episiotomy;         private
string _BookingType;
        private string _AttendantNurse;
private string _TreatmentCode;
private string _MaceratedStillBirth;
private string _FreshStillBirth;
private string _FoetalAbnormality;
private string _CordLength;         private
string _PlacentalWeight;         private
string _PlacentalMembrane;         private
string _PerinumState;         private string
_BirthWeight;         private string
_BirthLength;         private string
_CircumferenceHead;         private string
_ChildSex;         private string
_DeliveryType;         private string
_DeliveryTime;         private string
_DiagnosisCode;         private string
_RuptureOfMembrane;         private string
_PregnancyAge;         private string
_ChildrenDead;         private string
_ChildrenAlive;         private string
_Parity;         private string
_WhereBooked;         private string
_Booked;         private string _Physician;
private string _BloodLoss;         private
string _Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;

        public string  BloodLoss
        {
get
{
                return _BloodLoss;
            }
set
            {
                SetPropertyValue("BloodLoss", ref _BloodLoss, value);
            }
        }          public string
Physician
        {
get
{
                return _Physician;
            }
set
```

```
            {
                SetPropertyValue("Physician", ref _Physician, value);
            }
        }
        public string  Booked
        {
get
{
            return _Booked;
        }
set
            {
                SetPropertyValue("Booked", ref _Booked, value);
            }
        }
        public string  WhereBooked
        {
get
{
            return _WhereBooked;
        }
set
            {
                SetPropertyValue("WhereBooked", ref _WhereBooked, value);
            }
        }
        public string  Parity
        {
get
{
            return _Parity;
        }
set
            {
                SetPropertyValue("Parity", ref _Parity, value);
            }
        }
        public string  ChildrenAlive
        {
get
{
            return _ChildrenAlive;
        }
set
            {
                SetPropertyValue("ChildrenAlive", ref _ChildrenAlive, value);
}
        }
        public string  ChildrenDead
        {
get
{
            return _ChildrenDead;
        }
set
            {
                SetPropertyValue("ChildrenDead", ref _ChildrenDead, value);
            }
        }
        public string  PregnancyAge
```

```
        {
get
{
                return _PregnancyAge;
            }
set
            {
                SetPropertyValue("PregnancyAge", ref _PregnancyAge, value);
            }
        }
        public string  RuptureOfMembrane
        {
get
{
                return _RuptureOfMembrane;
            }
set
            {
                SetPropertyValue("RuptureOfMembrane", ref _RuptureOfMembrane, value);
            }
        }
        public string  DiagnosisCode
        {
get
{
                return _DiagnosisCode;
            }
set
            {
                SetPropertyValue("DiagnosisCode", ref _DiagnosisCode, value);
            }
        }
        public string  DeliveryTime
        {
get
{
                return _DeliveryTime;
            }
set
            {
                SetPropertyValue("DeliveryTime", ref _DeliveryTime, value);
}
        }
        public string  DeliveryType
        {
get
{
                return _DeliveryType;
            }
set
            {
                SetPropertyValue("DeliveryType", ref _DeliveryType, value);
            }
        }
        public string  ChildSex
        {
get
{
                return _ChildSex;
            }
set
```

```
            {
                SetPropertyValue("ChildSex", ref _ChildSex, value);
            }
        }
        public string  CircumferenceHead
        {
get
{
            return _CircumferenceHead;
        }
set
            {
                SetPropertyValue("CircumferenceHead", ref _CircumferenceHead, value);
            }
        }
        public string  BirthLength
        {
get
{
            return _BirthLength;
        }
set
            {
                SetPropertyValue("BirthLength", ref _BirthLength, value);
            }
        }
        public string  BirthWeight
        {
get
{
            return _BirthWeight;
        }
set
            {
                SetPropertyValue("BirthWeight", ref _BirthWeight, value);
}
        }
        public string  PerinumState
        {
get
{
            return _PerinumState;
        }
set
            {
                SetPropertyValue("PerinumState", ref _PerinumState, value);
}
        }

        public string  PlacentalMembrane
        {
get
{
            return _PlacentalMembrane;
        }
set
            {
                SetPropertyValue("PlacentalMembrane", ref _PlacentalMembrane, value);
            }
        }
        public string  PlacentalWeight
```

```
        {
get
{
                return _PlacentalWeight;
            }
set
            {
                SetPropertyValue("PlacentalWeight", ref _PlacentalWeight, value);
            }
        }
        public string  CordLength
        {
get
{
                return _CordLength;
            }
set
            {
                SetPropertyValue("CordLength", ref _CordLength, value);
            }
        }
        public string  FoetalAbnormality
        {
get
{
                return _FoetalAbnormality;
            }
set
            {
                SetPropertyValue("FoetalAbnormality", ref _FoetalAbnormality, value);
            }
        }          public string
FreshStillBirth
        {
get
{
                return _FreshStillBirth;
            }
set
            {
                SetPropertyValue("FreshStillBirth", ref _FreshStillBirth, value);
            }
        }
        public string  MaceratedStillBirth
        {
get
{
                return _MaceratedStillBirth;
            }
set
            {
                SetPropertyValue("MaceratedStillBirth", ref _MaceratedStillBirth, value);
            }
        }
        public string  TreatmentCode
        {
get
{
                return _TreatmentCode;
            }
set
```

```
            {
                SetPropertyValue("TreatmentCode", ref _TreatmentCode, value);
            }
        }


        public string  AttendantNurse
        {
get
{
                return _AttendantNurse;
            }
set
            {
                SetPropertyValue("AttendantNurse", ref _AttendantNurse, value);
            }
        }          public string
BookingType
        {
get
{
                return _BookingType;
            }
set
            {
                SetPropertyValue("BookingType", ref _BookingType, value);
}
        }
        public string  Episiotomy
        {
get
{
                return _Episiotomy;
            }
            set
            {
                SetPropertyValue("Episiotomy", ref _Episiotomy, value);
            }
        }
        public string  Complications
        {
get
{
                return _Complications;
            }
set
            {
                SetPropertyValue("Complications", ref _Complications, value);
            }
        }
        public string  ChestCircumference
        {
get
{
                return _ChestCircumference;
            }
set
            {
                SetPropertyValue("ChestCircumference", ref _ChestCircumference, value);
            }
        }
```

```
        public DateTime  DeliveryDate
        {
get
{
                return _DeliveryDate;
            }
set
            {
                SetPropertyValue("DeliveryDate", ref _DeliveryDate, value);
}
        }

        [Size(500)]
public string Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
}         }
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get {
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }

        }
    } }
```

Pharmacy Prescriptions

```
using System; using System.Linq; using
System.Text; using DevExpress.Xpo;
using DevExpress.ExpressApp; using
System.ComponentModel; using
DevExpress.ExpressApp.DC; using
DevExpress.Data.Filtering; using
DevExpress.Persistent.Base; using
System.Collections.Generic; using
DevExpress.ExpressApp.Model; using
DevExpress.Persistent.BaseImpl; using
DevExpress.Persistent.Validation;
using HIMS.Module.SetUp;

namespace HIMS.Module.Pharmacies
{

    [DefaultClassOptions]
[NavigationItem("Pharmacies")]     public
class Prescriptions : BaseObject
    {
        public Prescriptions(Session session)
            : base(session)
        {
```

```
        }          public override void
AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...
        private Pharmacist _Pharmacist;
private PatientName _PatientName;
private string _SheetCode;
private string _TotalAmount;
private string _Quantity;         private
string _Amount;
        private Duration _Duration;
        private Doctor _Doctor;
private Dosage _Dosage;         private
string _Description;         private
DrugCategory _DrugCategory;         private
string _Notes;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;

        public PatientName PatientName
        {
get
{
                return _PatientName;
            }
set
            {
                SetPropertyValue("PatientName", ref _PatientName, value);
            }
        }
        public DrugCategory DrugCategory
        {
get
{
                return _DrugCategory;
            }
set
            {
                SetPropertyValue("DrugCategory", ref _DrugCategory, value);
            }
        }

        public Pharmacist  Pharmacist
        {
get
{
                return _Pharmacist;
            }
set
            {
                SetPropertyValue("Pharmacist", ref _Pharmacist, value);
            }
        }
        public string  Description
```

```
        {
get
{
                return _Description;
            }
set
            {
                SetPropertyValue("Description", ref _Description, value);
}
        }

        public Dosage  Dosage
        {
get
{
                return _Dosage;
            }
set
            {
                SetPropertyValue("Dosage", ref _Dosage, value);
            }
        }
        public Doctor  Doctor
        {
get
{
                return _Doctor;
            }
set
            {
                SetPropertyValue("Doctor", ref _Doctor, value);
            }
        }
        public Duration  Duration
        {
get
{
                return _Duration;
            }
set
            {
                SetPropertyValue("Duration", ref _Duration, value);
            }
        }
        public string  Amount
        {
get
{
                return _Amount;
            }
set
            {
                SetPropertyValue("Amount", ref _Amount, value);
            }
        }
        public string  Quantity
        {
get
{
                return _Quantity;
```

```
            }
set
            {
                SetPropertyValue("Quantity", ref _Quantity, value);
            }
        }
        public string  TotalAmount
        {
get
{
            return _TotalAmount;
            }
set
            {
                SetPropertyValue("TotalAmount", ref _TotalAmount, value);
            }
        }
        public string  SheetCode
        {
get
{
            return _SheetCode;
            }
set
            {
                SetPropertyValue("SheetCode", ref _SheetCode, value);
            }
        }
[Size(500)]         public
string Notes
        {
get
{
            return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
}
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
            if (_auditTrail == null)
                _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
            return _auditTrail;
            }

        }

    } }
```

Stocks

```
using System; using
System.Linq; using
System.Text; using
DevExpress.Xpo;
```

```
using
DevExpress.ExpressA
pp; using
System.ComponentMod
el; using
DevExpress.ExpressA
pp.DC; using
DevExpress.Data.Fil
tering; using
DevExpress.Persiste
nt.Base; using
System.Collections.
Generic; using
DevExpress.ExpressA
pp.Model; using
DevExpress.Persiste
nt.BaseImpl; using
DevExpress.Persiste
nt.Validation;
using
HIMS.Module.SetUp;

namespace HIMS.Module.Pharmacies
{

    [DefaultClassOptions]
[NavigationItem("Pharmacies")]     public
class Stocks : BaseObject
    {
        public Stocks(Session session)
            : base(session)
        {

        }
        public override void AfterConstruction()
        {
            base.AfterConstruction();

        }

        // Fields...         private
DateTime _DateOfCollection;         private
DateTime _ExpiryDate;         private
DateTime _ManufactureDate;         private
DateTime _StockDate;         private Status
_Status;         private string
_DirectionOfUse;         private string
_Constituents;         private Dosage
_Dosage;         private SupplierCode
_SupplierCode;         private string
_CollectedBy;         private string
_PharmacyCode;         private string
_NAFDACRegNo;         private string
_BatchNo;         private string
_DiscountComputed;         private string
_DiscountPercent;         private string
_DiscountedAmount;         private string
_SellingPrice;         private string
_Quantity;         private string
_CostPrice;         private string _Notes;
```

```
private DrugCategory _DrugCategory;
private DrugName _DrugName;

        private XPCollection<AuditDataItemPersistent> _auditTrail = null;
public DrugName DrugName
        {
get
{
                return _DrugName;
            }
set
            {
                SetPropertyValue("DrugName", ref _DrugName, value);
            }
        }
        public DrugCategory DrugCategory
        {
get
{
                return _DrugCategory;
            }
set
            {
                SetPropertyValue("DrugCategory", ref _DrugCategory, value);
            }
        }
[Size(500)]         public
string  Notes
        {
get
{
                return _Notes;
            }
set
            {
                SetPropertyValue("Notes", ref _Notes, value);
            }
        }
        public string  CostPrice
        {
get
{
                return _CostPrice;
            }
set
            {
                SetPropertyValue("CostPrice", ref _CostPrice, value);
            }
        }
        public string  Quantity
        {
get
{
                return _Quantity;
            }
set
            {
                SetPropertyValue("Quantity", ref _Quantity, value);
            }
        }
        public string  SellingPrice
```

```
        {
get
{
                return _SellingPrice;
            }
set
            {
                SetPropertyValue("SellingPrice", ref _SellingPrice, value);
            }
        }
        public string  DiscountedAmount
        {
get
{
                return _DiscountedAmount;
            }
set
            {
                SetPropertyValue("DiscountedAmount", ref _DiscountedAmount, value);
            }
        }
        public string  DiscountPercent
        {
get
{
                return _DiscountPercent;
            }
set
            {
                SetPropertyValue("DiscountPercent", ref _DiscountPercent, value);
            }
        }
        public string  DiscountComputed
        {
get
{
                return _DiscountComputed;
            }
set
            {
                SetPropertyValue("DiscountComputed", ref _DiscountComputed, value);
            }
        }

        public DateTime  DateOfCollection
        {
              get
       {
                     return _DateOfCollection;
              }
              set
              {
                SetPropertyValue("DateOfCollection", ref _DateOfCollection, value);
              }
        }
        public string  BatchNo
        {
get
{
                return _BatchNo;
```

```
            }
set
            {
                SetPropertyValue("BatchNo", ref _BatchNo, value);
            }
        }
        public string  NAFDACRegNo
        {
get
{
                return _NAFDACRegNo;
            }
set
            {
                SetPropertyValue("NAFDACRegNo", ref _NAFDACRegNo, value);
            }
        }
        public string  PharmacyCode
        {
get
{
                return _PharmacyCode;
            }
set
            {
                SetPropertyValue("PharmacyCode", ref _PharmacyCode, value);
            }
        }

        public DateTime  ExpiryDate
        {
               get           {
        return _ExpiryDate;
               }
 set
              {
                SetPropertyValue("ExpiryDate", ref _ExpiryDate, value);
              }
        }
        public string CollectedBy
        {
get
{
                return _CollectedBy;
            }
set
            {
                SetPropertyValue("CollectedBy", ref _CollectedBy, value);
            }
        }
        public SupplierCode  SupplierCode
        {
get
{
                return _SupplierCode;
            }
set
            {
                SetPropertyValue("SupplierCode", ref _SupplierCode, value);
            }
        }
```

```
        public DateTime  ManufactureDate
        {
               get
 {
                     return _ManufactureDate;
               }
 set
              {
                SetPropertyValue("ManufactureDate", ref _ManufactureDate, value);
              }
        }
        public Dosage  Dosage
        {
get
{
                return _Dosage;
            }
set
            {
                SetPropertyValue("Dosage", ref _Dosage, value);
            }
        }
        public string  DirectionOfUse
        {
get
{
                return _DirectionOfUse;
            }
set
            {
                SetPropertyValue("DirectionOfUse", ref _DirectionOfUse, value);
}
        }

        public Status Status
        {
get
{
                return _Status;
            }
set
            {
                SetPropertyValue("Status", ref _Status, value);
            }
        }
        public string  Constituents
        {
get
{
                return _Constituents;
            }
set
            {
                SetPropertyValue("Constituents", ref _Constituents, value);
            }
        }

        public DateTime  StockDate
        {
```

```
                get
 {
                      return _StockDate;
                }
 set
               {
                 SetPropertyValue("StockDate", ref _StockDate, value);
               }
        }
        public XPCollection<AuditDataItemPersistent> AuditTrail
        {
get
{
                if (_auditTrail == null)
                    _auditTrail = AuditedObjectWeakReference.GetAuditTrail(Session, this);
                return _auditTrail;
            }

        }
        }

    }
```