CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

Advanced Hospital management system is software intended to atomize the functionalities of a hospital which are done currently through paper works. The main Feature in this project is intended to fix an appointment with the doctor by the patient. Our project Advanced Hospital Management System include registration of patients, storing their details into the system and also computerized billing. The system will be helpful for the employees in the hospital to manage the working smoothly and efficiently. It will also helpful for the patients who visiting the hospital. This system will helpful to avoid the delay caused in the existing system and also provide an efficient method for providing the details about the patient such as the date at which he admitted, ward he admitted etc. That is it will give a complete details about the patient.

This system comprises mainly five user’s admin, receptionist, head nurse, pharmacist and patient. This part of the system contains admin, receptionist, nurse, pharmacist, patient login. The admin has the full control of the system. He can add and remove doctors and staff. And can also view the patient’s details. The newly registered users can login to the system only after the approval of admin. The receptionist does the op registration, token generation, conversion of op to ip, generation of discharge cards. The head nurse can manage the next login. The Head nurse allots room for inpatients. And can also change the room. The pharmacist does the medicine billing and controls the medicine store details.

1.2 PROJECT SPECIFICATION

Hospital is the essential part of our lives, providing best medical facilities to peoples suffering from various ailments. It is necessary for the hospitals to keep track of its day-to-day activities and records of its patients, doctors, staff, ward etc. that keep the hospital running smoothly and successfully.

- The system is desired to handle the activities of the Hospital
- Patients can fix online appointment with the doctor
- The system will have the ability to manage

  -Registration of the Patients.

-Registration of Doctors and other staffs.

-Online Booking.

-Automatic billing.

-Online allocation of wards and rooms for IP

CHAPTER 2

SYSTEM STUDY

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minutes detail and analysed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analysing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2 EXISTING SYSTEM

The existing system is a manual processing system. In most government and small hospitals records are held manually. Patient details are kept as written records in register. It is difficult to search for patient details in the existing system. Also there is no queue order in consulting doctor. Doctors make prescription to patients in handwritten slips. For generating any report they need to refer through previous records of patients. It will be more difficult if they need to know the number of patients admitted with a specific disease in certain time period. They would have to check each and every records within the given time period. After few days of discharge, if a person needs his discharge card for any purpose then the authority have to check that persons previous records to give his discharge card.

2.3 DRAWBACKS OF EXISTING SYSTEM

- Accuracy not achieved.
- Time consuming.
- A lot of paper works
- There is chance of losing record
- No direct role for the higher officials

To avoid all these limitations and make the system working more accurately it needs to be computerized.

2.4 PROPOSED SYSTEM

The proposed system handles all the difficulties of the existing system. The Hospital management system software is user-friendly software. The main objectives of the system are which shows and helps you to collect most of the information about Hospitality and Medical Services. The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations.

The proposed system includes dedicated user interface for carrying out operations for operator, nurse/doctor, admin and record librarian. This system is very user friendly and fast in recording, retrieving and storing data and data is maintained efficiently by automating it into computerized system. Moreover, the graphical user interface provided in the proposed system makes user to interact with the system easily. The main objectives of the proposed system can be enumerated as follows:

- Patients are easily allocated to the doctors.
- Doctors Search is possible.
- Today’s patient list helps doctors to search their patients.
- Patients can fix online appointment with the doctor.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Easier
- More Efficient
- Record Keeping is easier
- Security of data.
- Administrator controls the entire system.
- Minimize manual data entry.
- Minimum time required

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provide the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features:

3.1.1 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The system is user friendly, so there is no cost to spend for training users. The advantage of the system includes low cost. So it is economically feasible

3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using ASP.NET in front end and SQL server in back end, the project is technically feasible for development.

3.1.3 Behavioural Feasibility

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor	:	Pentium IV/AMD Dual core
RAM	:	1 GB
Hard disk	:	500 GB

3.2.2 Software Specification

Front End	:	ASP.NET
Backend	:	SQL SERVER
Client on PC	:	Windows 7 and above.
Technologies used	:	JS, HTML5, AJAX, J Query

3.3 SOFTWARE DESCRIPTION

3.3.1 ASP.Net

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE,.NET Framework, .NET Compact Framework and Microsoft Silverlight. Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

.NET is an object-oriented programming (OOP) model introduced to help developers create Internet-based distributed systems. It provides a platform-independent framework that enables developers to quickly build, deploy, and manage Web-based applications, smart client applications, and XML Web services applications. The platform-independence feature enables businesses to quickly integrate their systems, information, and devices, thereby helping users collaborate and communicate effectively. **ASP.NET** is a server side scripting technology that enables scripts (embedded in web pages) to be executed by an Internet server.ASP.NET is a unified Web development model that includes the services necessary for us to build enterprise-class Web applications with a minimum of coding. ASP.NET is part of the .NET Framework, and when coding ASP.NET applications we have access to classes in the .NET Framework. You can code our applications in any language compatible with the common language runtime (CLR), including Microsoft Visual Basic and C#. These languages enable us to develop ASP.NET applications that benefit from the common language runtime, type safety, inheritance, and so on.

Working of ASP.NET:

1. When a browser requests an HTML file, the server returns the file

2. When a browser requests an ASP.NET file, IIS passes the request to the ASP.NET engine on the server

3. The ASP.NET engine reads the file, line by line, and executes the scripts in the file

4. Finally, the ASP.NET file is returned to the browser as plain HTML

ASP.NET File

1. An ASP.NET file can contain HTML, XML, and scripts

2. Scripts in an ASP.NET file are executed on the server

3. An ASP.NET file has the file extension ".aspx"

Comparison of ASP.NET with other programming languages: ASP.NET framework also provides big number of built-in classes. However, PHP provides limited built-in classes. PHP application development is mostly done in Dreamweaver. Dreamweaver application development environment provides limited functionalities and features. ASP.NET application also provides better error trace than PHP application. Thus, application development is easier and faster using ASP.NET than PHP.

FEATURES

Some of the main features of ASP.NET are:

1. Multiple Language Support

ASP.NET supports multiple programming languages for web application development. Applications using ASP.NET can be developed using over 20 languages. Microsoft has the compilers for Visual Basic, MS Visual C#, MS Visual C++, and MS Jscript. .NET compilers for COBOL, Pascal, Perl and Smalltalk, among others, can also be used to develop the Web application. For example, a programmer may write code for one module in Visual Basic.NET and code for another module in Jscript and Visual C#.

2. Code is compiled

One of the most important features of ASP.NET is that the code is compiled. Code compilation means that the programmatic instructions are converted to the machine language. In ASP.NET, however, code is not compiled to the machine language directly. It is compiled to an intermediate language called Microsoft Intermediate Language (MSIL or IL). This code is further compiled to machine language by the JIT compiler (just-in-time compiler). The JIT compiler compiles each portion of code as it is called, instead of compiling the complete application at one shot. Due to this, application start-up time is less. The compiled code is stored till the application exits and hence does not have to be recompiled each time the compiled portion of the code gets called. The compilation of code greatly improves Web applications performance.

3. Classes and Namespaces

ASP.NET is an object oriented technology and includes a collection of useful classes and namespaces. For example, the Classes HtmlTextBox, HtmlLabel and HtmlForm are included within the System. Web.UI.HtmlContrl namespace.

4. Server Controls

A set of server controls is provided in ASP.NET. These controls provide various properties, methods and events for simplifying the making of powerful web applications.

HTML

HTML, which stands for Hyper Text Mark-up Language, is the predominant mark-up language for web pages. HTML is the basic building-blocks of web pages. A mark-up language is a set of mark-up tags, and HTML uses mark-up tags to describe web pages. The purpose of a web browser is to read HTML documents and compose them into visual or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. It can embed scripts in languages such as JavaScript which affect the behaviour of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the appearance and layout of text and other material. The W3C, maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicitly presentational HTML mark up.

3.3.2 Structured Query Language (SQL)

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 (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time). SQL Server 2012 (formerly codenamed "Katmai") was released on August 6, 2012 and aims to make data management self-tuning, self-organizing, and self-maintaining with the development of *SQL Server Always On* technologies, to provide near-zero downtime. SQL Server 2012 also includes support for structured and semi-structured data, including digital media formats for pictures, audio, video and other multimedia data. In current versions, such multimedia data can be stored as BLOBs (binary large objects), but they are generic bit streams. Intrinsic awareness of multimedia data will allow specialized functions to be performed on them. SQL Server 2012 can be a data storage backend for different varieties of data: XML, email, time/calendar, file, and document, spatial, etc. as well as perform search, query, analysis, sharing, and synchronization across all data types. SQL Server 2012 also natively supports hierarchical data, and includes T-SQL constructs to directly deal with them, without using recursive queries. SQL Server 2012 R2 adds certain features to SQL Server 2012 including a master data management system branded as Master Data Services, a central management of master data entities and hierarchies. Also Multi Server Management, a centralized console to manage multiple SQL Server 2012 instances and services including relational databases, Reporting Services, Analysis Services & Integration Services.

Microsoft SQL Server 2012 is a set of components that work together to meet the data storage and analysis needs of the largest Web sites and enterprise data processing systems. SQL Server is a relational database management system for distributed Client-Server computing. Like all other database management systems, it provides the following features:

Features

- A variety of user interfaces
- Physical data independence
- Logical data independence
- Query optimization
- Data integrity
- Concurrency control
- Backup and recovery
- Security and authorization

Query Analyser

Query Analyzer is another tool with SQL server, which extends the capabilities of ANSI standard SQL. It is an application that recognizes and executes SQL commands and specialized T_SQL commands that can be used to create database objects using SQL commands. We can use query analyzer commands to

- Enter, Edit, Store, and Retrieve and Run SQL commands.
- Format, perform calculations on, store, and print query results.
- List column definitions for any table.
- Access any copy of data between SQL databases.
- We can create tables and insert data or alter it or we can delete data using SQL commands.

Features of SQL Server

Microsoft SQL Server features include:

- **Internet Integration**

The SQL Server 2012 database engine includes integrated XM support. It has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites.

- **Scalability and Availability**

The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows 98 through large, multiprocessor servers running Microsoft Windows 2005 Data centre Edition. SQL Server 2012 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.

- **Enterprise-Level Database Features**

The SQL Server 2012 relational database engine supports the features required to support demanding data processing environments .The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database.

- **Ease of installation, deployment, and use**

SQL Server 2012 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing and using SQL Server across several sites. SQL Server 2012 also supports a standards- base programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems.

- **Data Warehousing**

SQL Server 2012 includes tools for extracting and analysing summary data for online analytical processing .SQL Server also includes tools for visually designing databases and analysing data using English-based questions.

Advantages of SQL Server 2012 as a Database Server

Microsoft SQL Server 2012 is capable of supplying the database services needed by extremely large systems. Large servers may have thousands of users connected to an instance of SQL Server 2012 at the same time .SQL Server 2012 has full protection for these environments, with safeguards that prevent problems, such as having multiple users trying to update the same piece of data at the same time .SQL Server 2012 also allocates the available resources effectively, such as memory, network bandwidth, and disk I/O, among the multiple users. Extremely large Internet sites can partition their data across multiple servers, spreading the processing load across many computers, and allowing the site to serve thousands of concurrent users. Multiple instances of SQL Server 2012 can be run on a single computer.

CHAPTER 4

SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term "design" is defined as "the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 ARCHITECTURAL DESIGN

This section describes the components of the Advanced Hospital Management System.

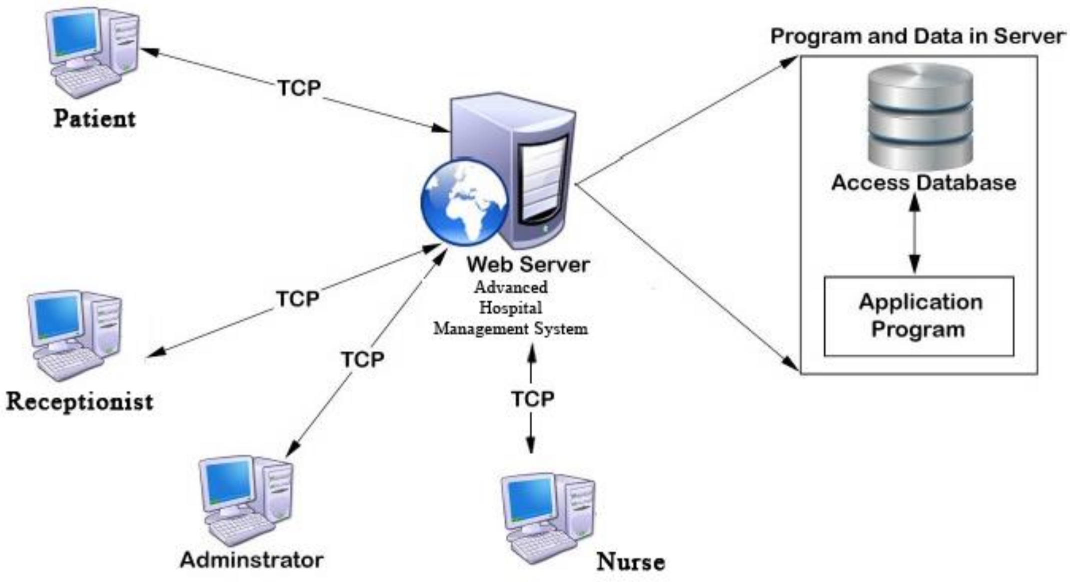


Figure 1: Architectural Design

The user, admin, nurse, receptionist access Advanced hospital management System through internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System's application program process the user's request and provides the required services by taking data from the system database.

4.3 MODULE DESIGN

Admin Module

The admin module includes the approval for staff. He can have the proficiency to add the doctor's information, add new wards, rooms, searching, report generation and also have the permission to approve the registered user.

Manage State , District, Ward, Rooms	Manage Doctor Specialisations and Staff Details
Approve/Reject the registered users	Report of IP and OP

Receptionist / Pharmacist.

This module includes the op registration, conversion of op to ip, billing, adding and updating medicine details and discharge card generation.

Manage op registration	Confirm the user appointments
OP to IP Conversion, Transfer ward	Billing and discharge

Nurse

Each ward has a head nurse. When the doctor refers a patient to admit in a particular ward then the notification will be pass to the corresponding head nurse by the receptionist. After receiving the notification the nurse allocates room to the patient.

Room Allocation of patients	Ward transfer

Patient.

New user can register through online. After admin approves the users can view the details of doctors and they can take online appointment with the doctor through online.

The user can register and then login	The user can update their profiles
Can book online appointment to the Doctor	Can view Doctor’s details
Will get hospital approval by Email	

4.4 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It’s a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

The purpose of the design is to create architecture for the evolving implementation and to establish the common tactical policies that must be used by desperate elements of the system. We begin the design process as soon as we have some reasonably completed model of the behaviour of the system. It is important to avoid premature designs, wherein develop designs before analysis reaches closer. It is important to avoid delayed designing where in the organization crashes while trying to complete an unachievable analysis model.

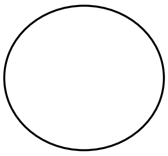
Throughout the project, the context flow diagrams, data flow diagrams and flow charts have been extensively used to achieve the successful design of the system. In our opinion, “efficient design of the data flow and context flow diagrams helps to design the system successfully without much major flaws within the scheduled time”. This is the most complicated part in a project. In the designing process, our project took more than the activities in the software life cycle. If we design a system efficiently with all the future enhancements, the project will never become junk and it will be operational.

The data flow diagrams were first developed by Larry Constantine as way for expressing system requirements in graphical form. A data flow diagram also known as “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. It functionally decomposes the requirement specification down to the lowest level. DFD depicts the information flow, the transformation flow and the transformations that are applied as data move from input to output. Data Flow Diagram is quite effective, especially when the required design is unclear and the user and analyst need a notational language for communication. It is used to model the system components such as the system process, the data used by the process, any external entities that interact with the system and information flows in the system.

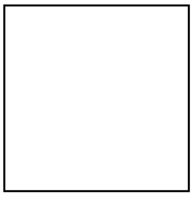
Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Main symbols used in the data flow diagram are:

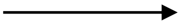
1. Circle represents a process that transforms incoming data flows in to outgoing data flows.



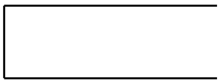
2. A square defines a source and destination of system data.



3. Arrow identifies data in motion.



4. An open rectangle defines a data store, data at rest or temporary repository of data.



Steps to Construct Data Flow Diagrams:-

Four steps are commonly used to construct a DFD:

- Process should be named and numbered for easy reference. Each name should be representative of the process.
- The destination of flow is from top to bottom and from left to right.
- When a process is exploded in to lower level details they are numbered.
- The names of data stores, sources and destinations are written in capital letters.

Rules for constructing a Data Flow Diagram

- Arrows should not cross each other.
- Squares, circles and files must bear names.
- Decomposed data flow squares and circles can have same names.
- Draw all data flow around the outside of the diagram.

Data Flow Diagrams of Advanced hospital management system

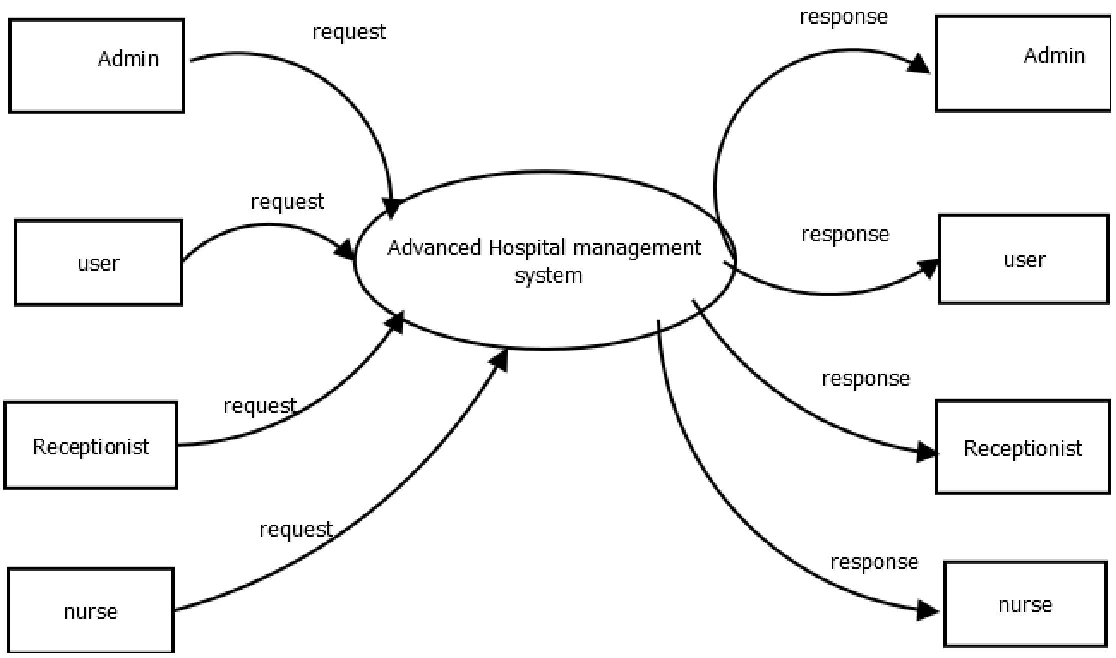


Figure 2

Admin Level-1

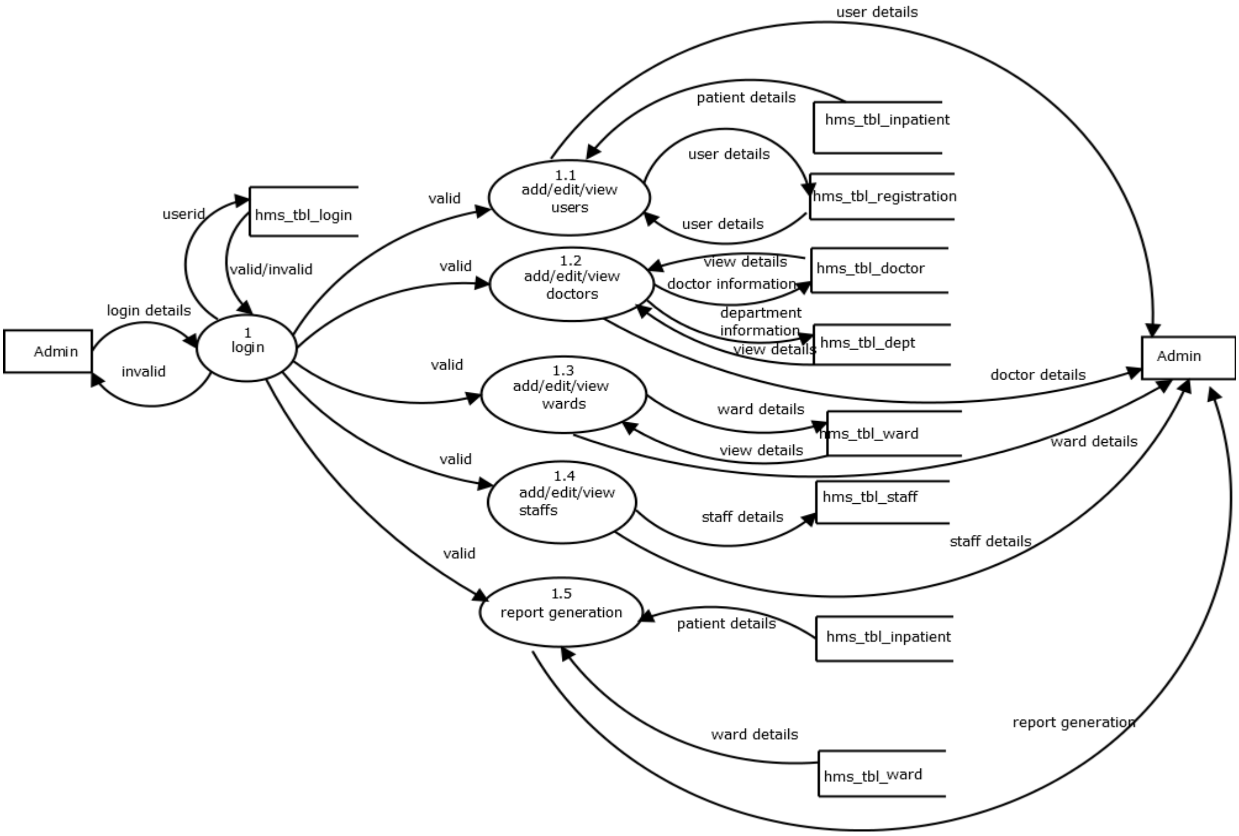


Figure 3

Receptionist Level-1

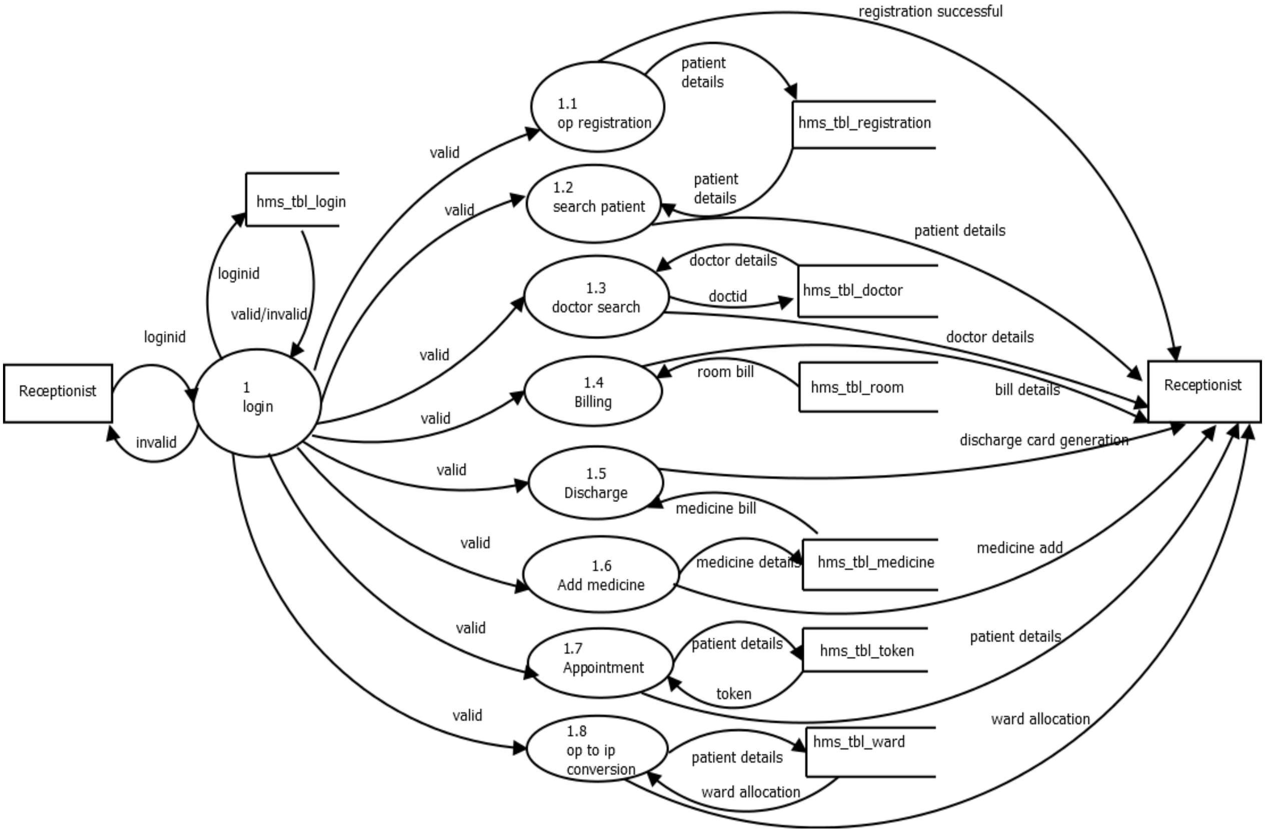


Figure 4

Nurse Level-1

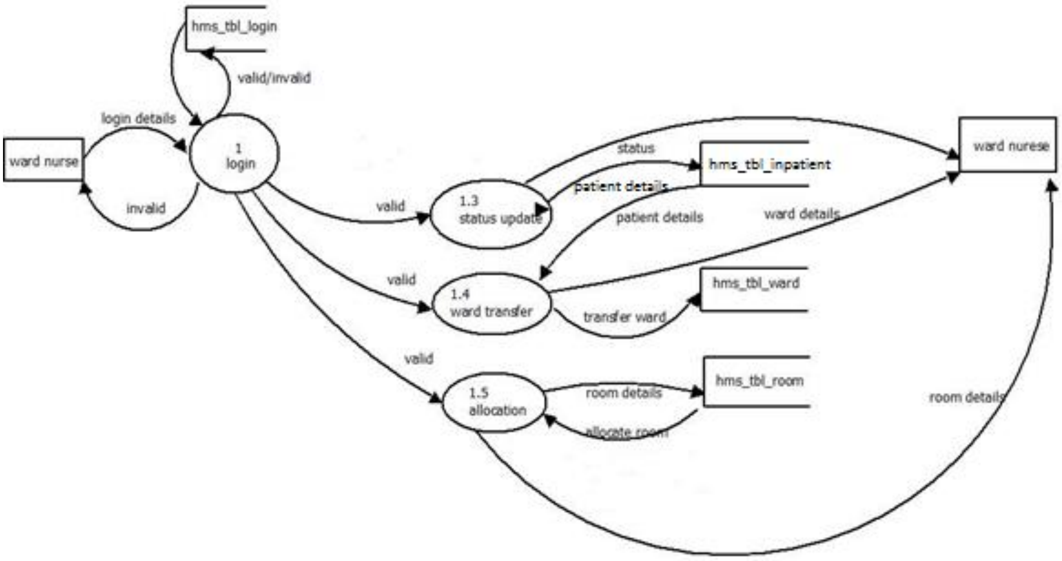


Figure 5

Patient Level-1

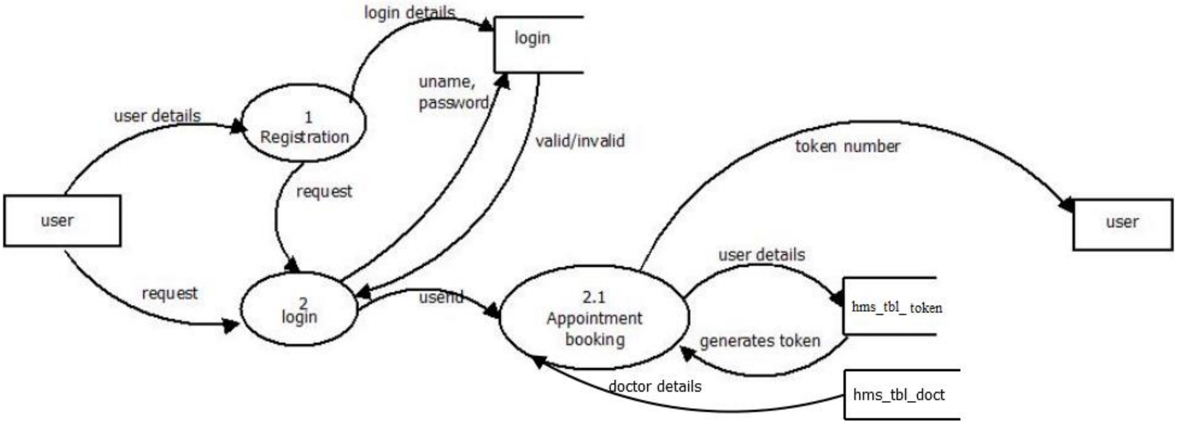


Figure 6

4.5 USER INTERFACE DESIGN

- **Login Form**

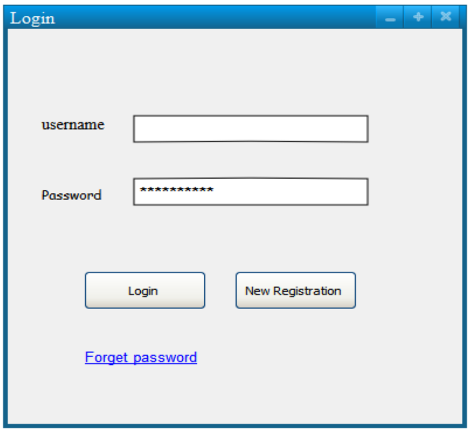


Figure 7. Login form

- **Registration**

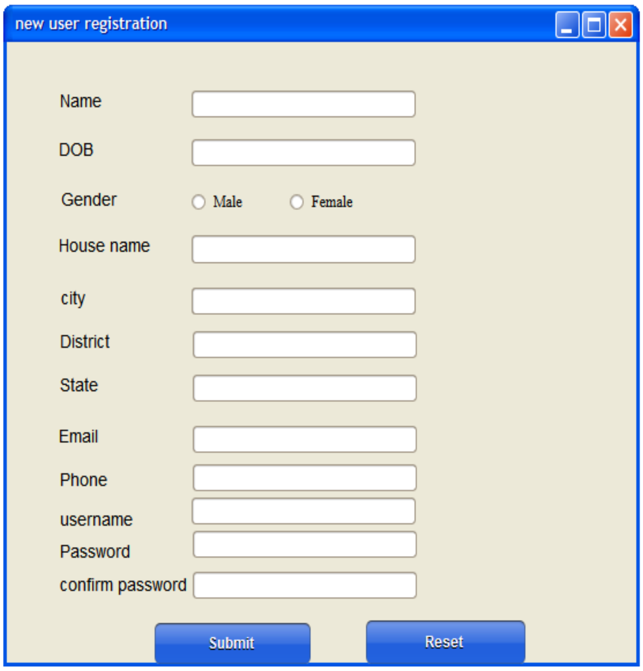


Figure 8. Registration form

- **Appointment**

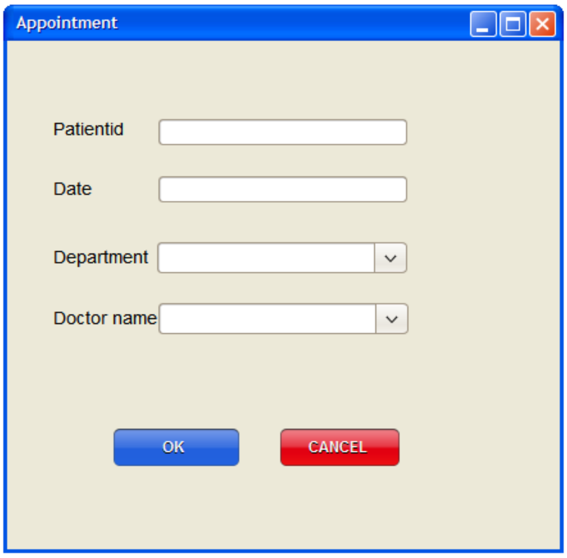


Figure 9. Appointment booking

- **Op Registration**

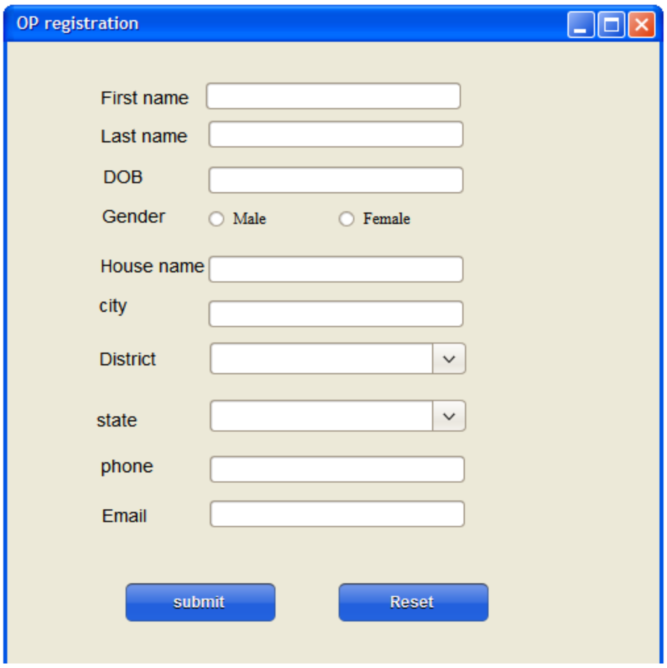
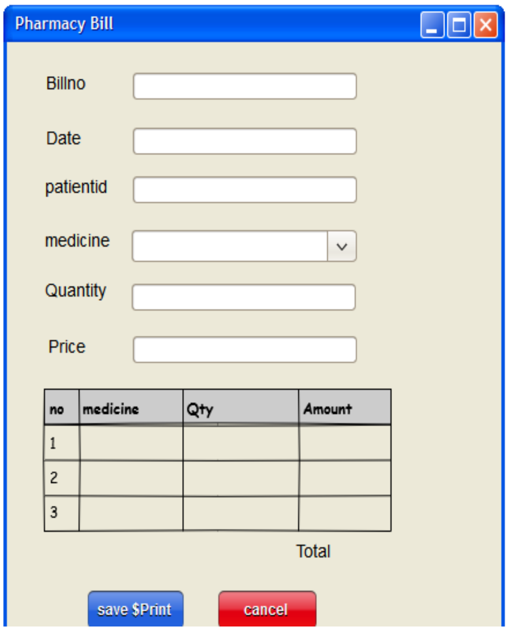


Figure 10. Op registration

- **Pharmacy Bill**



Pharmacy Bill

Billno

Date

patientid

medicine

Quantity

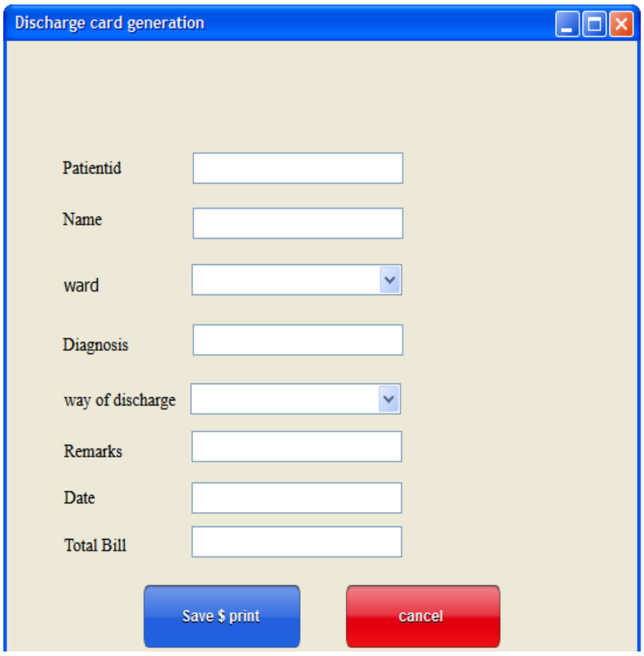
Price

no	medicine	Qty	Amount
1			
2			
3			

Total

Figure 11.Pharmacy bill

- **Discharge Sheet**



Discharge card generation

Patientid

Name

ward

Diagnosis

way of discharge

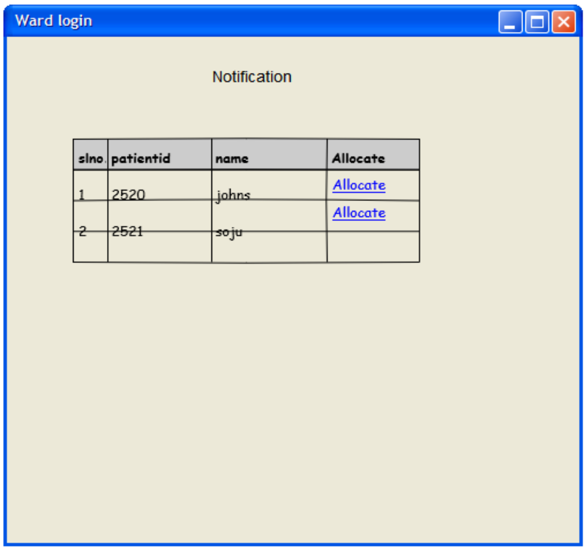
Remarks

Date

Total Bill

Figure 12.Discharge sheet

- ## Nurse - Notification



slno	patientid	name	Allocate
1	2520	johns	Allocate
2	2521	soju	Allocate

Figure 13.notification

- ## Room Allocation

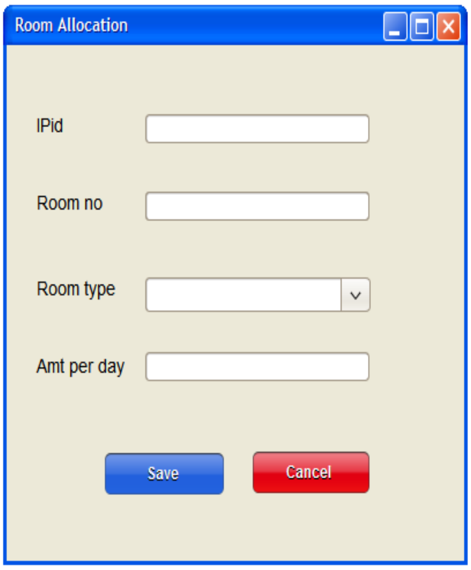


Figure 14.Room allocation

Admin Login

- ### Add staff

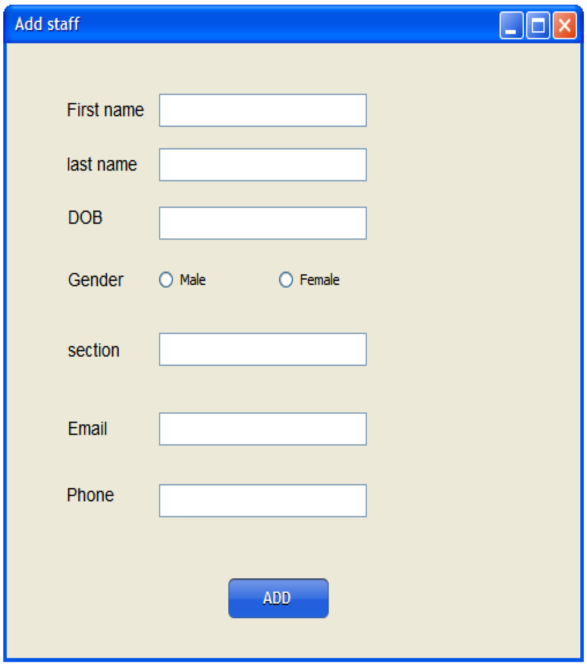


Figure 15.Add staff

- ### Add Doctor

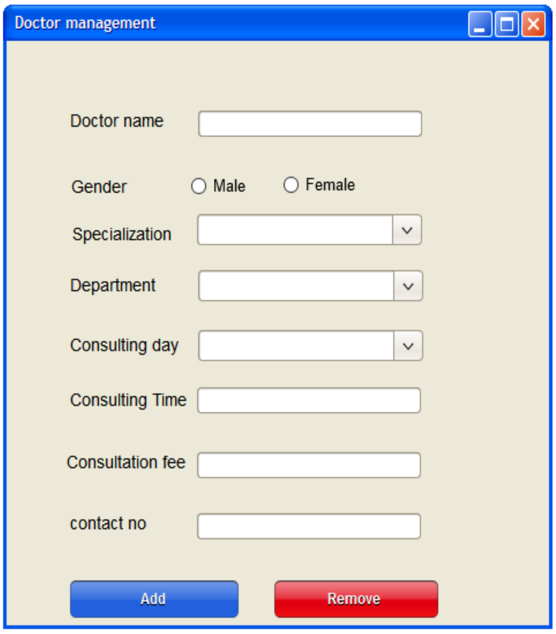
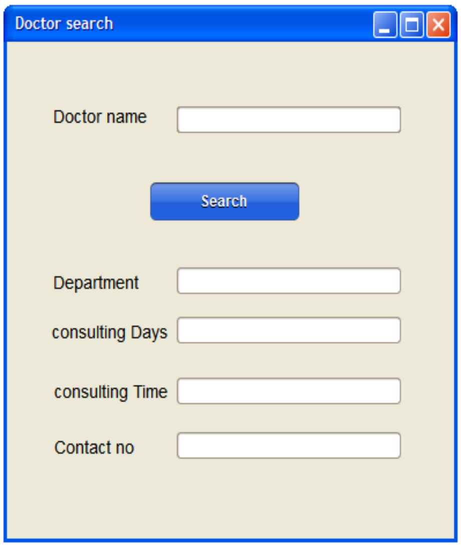


Figure 16. Add doctor

- ## Search Doctor



Doctor search

Doctor name

Department

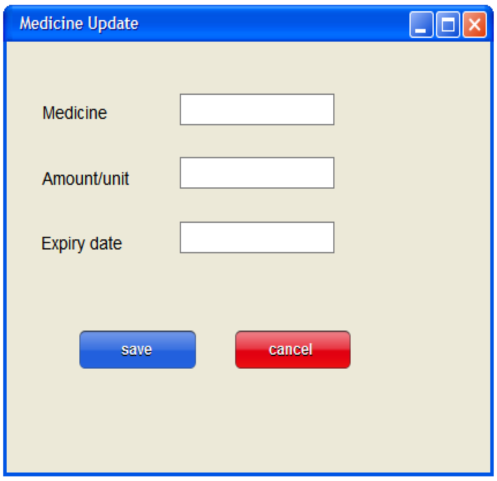
consulting Days

consulting Time

Contact no

Figure17. Search doctor

- ## Medicine Update



Medicine Update

Medicine

Amount/unit

Expiry date

Figure 18.Medicine update

4.6. Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

4.6.1 Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

4.6.2 Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity. Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies. Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table. There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form.

As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute. In other words 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be donor by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non-atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key. A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

Table no: 1 hms_tbl_registration

Primary Key: Patientid

Foreign Key :Lid

Field Name	Data Type	Size	Description
Patientid	Integer	2	Patient id
Lid	Integer	2	Login id
Fname	Varchar	20	First name
Lname	Varchar	20	Last name
Gender	Varchar	20	Gender
Dob	Date		Date of birth
Mobno	Integer	20	Contact number
Hname	Varchar	20	House name
Email	Varchar	20	Email id
City	Varchar	20	City name
Distid	Varchar	20	District id
Sid	Varchar	20	State id
Status	Integer	1	Status

Table No: 2

hms_tbl_login

Primary Key: Lid

Field Name	Data type	Size	Description
Lid	Integer	2	Login id
Uname	Varchar	10	Username
Pwd	Varchar	10	Password
Type	Integer	2	User type
Status	Integer	1	Status

Table No: 3

hms_tbl_dept

Primary Key: Deptid

Field Name	Data type	Size	Description
Deptid	Integer	2	Department id
Deptname	Varchar	50	Department name
Status	Integer	1	Status

Table No: 4

hms_tbl_doctor

Primary Key:Doctid

Foreign Key:Deptid

Field Name	Data Type	Size	Description
Doctid	Integer	2	Doctor id
Deptid	Integer	2	Department id
Fname	Varchar	50	First name
Lname	Varchar	50	Last name
Gender	Varchar	10	Gender
Conslt_day	Varchar	50	Consulting day
Conslt_time	Varchar	50	Consulting time
Mob	Integer	2	Contact number
Fee	Integer	8	Consultation fee
Status	Integer	1	Status

Table No: 5

hms_tbl_medicine

Primary Key: Mid

Field Name	Data type	Size	Description
Mid	Integer	2	Medicine id
Mname	Varchar	100	Medicine name
Cname	Varchar	100	Company name
Price	Integer	5	Price
Mdate	Varchar	100	Manufacture date
Exp_date	Varchar	30	Expiry date
Rack	Varchar	30	Rack
Rrow	Varchar	30	Specifies the row
Rcolumn	Varchar	30	Specifies the column
Remark	Varchar	30	Remarks
Status	Integer	1	Status

Table No: 6

hms_tbl_bill

Primary Key: Id

Foreign Key:Patientid

Field Name	Data Type	Size	Description
Id	Integer	2	Id
Patientid	Integer	2	Patient id
Itemname	Varchar	50	Item name
Qty	Varchar	50	Quantity
Price	Varchar	50	Price
Pharmacy	Integer	2	Pharmacyid
Status	Integer	1	Status

Table No: 7

hms_tbl_ward

Primary Key:Wid

Foreign Key:Deptid

Field Name	Data Type	Size	Description
Wid	Integer	2	Ward id
Deptid	Integer	2	Department id
Wname	Varchar	20	Ward name
Totroom	Integer	6	Total room
Status	Integer	1	Status

Table No 8

hms_tbl_staff

Primary Key:Staffid

Foreign key: Lid

Field Name	Data Type	Size	Description
Staffid	Integer	2	Staff id
Lid	Integer	2	Login id
Wid	Integer	2	Ward id
Deptid	Integer	2	Department id
Firstname	Varchar	20	First name
Lastname	Varchar	20	Last name
Dob	Date		Date of birth
Gender	Varchar	10	Gender
Email	Varchar	20	Email id
Stype	Varchar	20	Staff type
Phone	Integer	10	Contact number
Status	Integer	1	Status

Table No 9

hms_tbl_token.

Primary Key:Id

Foreign key: Patientid

Field Name	Data Type	Size	Description
Id	Integer	2	Id
Patientid	Integer	2	Patient id
Doctor	Integer	2	doctor id
Date	Varchar	50	Token date
Status	Integer	1	Status

Table No 10

hms_tbl_room

Primary Key:Roomid

Foreign Key:Wid,Patientid

Field Name	Data Type	Size	Description
Roomid	Integer	2	Room id
Wid	Integer	20	Ward id
Roomname.	Varchar	20	Room no.
Patientid	Integer	2	Patient id
Date	Varchar	50	Admit date
Rent	Varchar	50	Room rent
Status	Integer	1	Status

Table No: 11

hms_tbl_state

primary key:Sid

Field Name	Data Type	Size	Description
Sid	Integer	2	State id
Sname	Varchar	100	State name
Status	Integer	1	Status

Table No: 12

hms_tbl_district

primary key: Distid

foregin key :Sid

Field Name	Data Type	Size	Description
Distid	Integer	2	id
Dname	Varchar	50	District name
Sid	Integer	2	State id
Status	Integer	1	Status

Table No: 13

hms_tbl_inpatient

Primary Key: Id

Foreign Key: Patientid

Field Name	Data Type	Size	Description
Id	Integer	2	Id
Patientid	Integer	2	Patient id
Ward	Integer	2	Ward id
Room	Integer	2	Room id
Doctor	Integer	2	Doctor id
Status	Integer	1	Status

CHAPTER 5

TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Validation **:** Are we doing the right job?

Verification **:** Are we doing the job right?

Software testing should not be confused with debugging. Debugging is the process of analysing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software, a methodical approach to software testing is a much more thorough means for identifying bugs. Debugging is therefore an activity which supports testing, but cannot replace testing.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis looks at the behaviour of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

5.2 TEST PLAN

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- ❖ Unit testing
- ❖ Integration Testing
- ❖ Data validation Testing
- ❖ Output Testing

5.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white-box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur. Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

Unit testing was done in Sell-Soft System by treating each module as separate entity and testing each one of them with a wide spectrum of test inputs. Some flaws in the internal logic of the modules were found and were rectified. After coding each module is tested and run individually. All unnecessary code were removed and ensured that all modules are working, and gives the expected result.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop.

After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

5.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

5.3 TEST CASES

5.3.1 Test Cases 1

Project Name:Advanced Hospital Management System	
Advanced Hospital Management System Test Case 1	
Test Case ID:Test Case 1	Test Designed by: Shijo K Jacob
Test Priority(Low/Medium/High):Medium	Test Designed date:20/10/2016
Module Name: Login	Test Executed by: Ms. Teenu V Therese
Test Title: Verify login with valid username and Password	Test Execution date:20/10/2016
Description: Test the Login Page	
Pre-conditions: User has Valid username and Password	
Dependencies	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Note
1	Navigation to login page		Login page for users	Login page for users	Pass	
2	Provide Valid username	Username admin@gmail.com	User should be able to login	Logged in and the User is navigated to Dashboard with Records	Pass	
3	Provide valid password	Password: admin				
4	Click on login Button					

Post conditions:
User is validated with database and successfully login to account. The account session details are logged in database

5.3.2 Test Cases 2

Project Name: Advanced Hospital Management System	
Advanced Hospital Management System Test Case 2	
Test Case ID:Test Case 2	Test Designed by: Shijo K Jacob
Test Priority(Low/Medium/High):Medium	Test Designed date:20/10/2016
Module Name: Doctor Management	Test Executed by: Teenu V Therese
Test Title: Adding the details of the doctor	Test Execution date:20/10/2016
Description: Hospial Doctor Adding	
Pre-conditions: Doctor data should be correct	
Dependencies	

Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Note
1	Navigation to Doctor Adding Page		Add and edit doctors page	Add and edit doctors page	Pass	
2	Provide Valid data	Doctor Details	Successfully added message should be shown	Message shown and redirected to	Pass	
3	Click on Save Button	Validations				

Post conditions:
User is validated with database and other validations.

5.3.3 Test Cases 3

Project Name: Advanced Hospital Management System						
Advanced Hospital Management System Test Case 3						
Test Case ID:Test Case 3				Test Designed by: Shijo K Jacob		
Test Priority(Low/Medium/High):Medium				Test Designed date:20/10/2016		
Module Name: Advanced Hospital Management System				Test Executed by: Teenu V Therese		
Test Title: Registering user				Test Execution date:20/10/2016		
Description: Test the user Registration						
Pre-conditions: User Registration data should be correct						
Dependencies						
Step	Test Steps	Test Data	Expected Result	Actual Result	Status(Pass/Fail)	Note
1	Navigation to user l Registration Page		User registration page	User registration page	Pass	
2	Provide Valid data	User Registration Data	Successfully added message should be shown	Message shown and redirected to	Pass	
3	Click on Save Button	Validations				
Post conditions: User Registration data adding is validated with database and other validations.						

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

Implementation is the stage of the project where the theoretical design is tuned into a working system. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover phase.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won’t take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy..

6.2.3 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Most of the hospitals in Kerala are following old paper system to record patient details, which is difficult for them for report generation and viewing previous patient records. This project is designed in such a way that it is suitable for every hospital across India. Hospital is the essential part of our lives, providing best medical facilities to peoples suffering from various ailments. It is necessary for the hospitals to computerize the whole working for running smoothly and successfully. We can also extend this software to another area in the hospital like hospital office automation, asset tracking, various certificate generations, etc.

The successful completion of the system result in:-

- User-friendliness.
- Fast data processing.
- Generation of reports in neat format.
- Protection of the system from unauthorized user’s access.

In abstract, this system is perceived to the user as a reliable and user-friendly manner.

7.2 FUTURE SCOPE

We can also extend this software to another areas in the hospital like hospital office automation, asset tracking, various certificate generations, etc.

CHAPTER 8

BIBLIOGRAPHY

BOOKS/REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “So*ftware engineering*: a precise approach”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.htm

CHAPTER 9

APPENDIX

9.1 Sample Code

1.LOGIN

***@since File available since Release 1.0**

***@created on 09-09-2016**

***@last edited on oct-10-2016**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Web.Security;
using myhospital;
using System.Configuration;

namespace MyHospital
{
    public partial class _Default : Page
    {
        Class1 obj = new Class1();
        protected void Page_Load(object sender, EventArgs e)
        {

        }
        public static string EncodePasswordToBase64(string password)
        {
            try
            {
                byte[] encData_byte = new byte[password.Length];
                encData_byte = System.Text.Encoding.UTF8.GetBytes(password);
                string encodedData = Convert.ToBase64String(encData_byte);
                return encodedData;
            }
            catch (Exception ex)
            {
                throw new Exception("Error in base64Encode" + ex.Message);
            }
```

```
}
protected void btnlogin_Click(object sender, EventArgs e)
{
   login_check();
}
public void login_check()
{
   obj.cmd.Parameters.Clear();
   DataTable dt = new DataTable();
   string user = txtuname.Text;
   string pass = EncodePasswordToBase64(txtpwd.Text);
   obj.cmd.Parameters.AddWithValue("@username", user);
   obj.cmd.Parameters.AddWithValue("@password", pass);
   dt = obj.gettable("login_check");




   int c = dt.Rows.Count;
   if (c == 1)
   {
      int id = Convert.ToInt32(dt.Rows[0].ItemArray[0]);
      int des = Convert.ToInt32(dt.Rows[0].ItemArray[1]);
      if (des == 1)
      {
         Session["id"] = id;
         Response.Redirect("area/admin/adminhome.aspx");

      }
      else if (des == 2)
      {
         Session["id"] = id;
         Response.Redirect("area/patient/patienthome.aspx");

      }
      else if (des == 3)
      {
         Session["id"] = id;
               Response.Redirect("area/receptionist/receptionistHome.aspx");

      }
      else if (des == 4)
      {
         Session["id"] = id;
         Response.Redirect("area/Pharmacy/phhome.aspx");

      }
      else if (des == 5)
      {
```

```
                    Session["id"] = id;
                    Response.Redirect("area/nurse/nursehome.aspx");

                }
            }
            else
                lbl_msg.Text = "Login Failed..!!";

        }


    }
}
```

Stored procedure

```
CREATE procedure [dbo].[login_check]
@username varchar(50),
@password varchar(50)
as
begin
set nocount on;
select lid,type from hms_tbl_login where status=1 and uname=@username and
pwd=@password
end
```

2.New user Registration

***@since**	**File available since Release 1.0**
***@created on**	**09-09-2016**
***@last edited on**	**oct-10-2016**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Security.Cryptography;
using System.Web.Security;
using System.IO;
using System.Net.Mail;
using System.Net;
using myhospital;

namespace MyHospital.Area.Shared
{
   public partial class NewReg : System.Web.UI.Page
   {
      Class1 obj = new Class1();
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            viewstates();


         }
      }
      public static string EncodePasswordToBase64(string password)
      {
         try
         {
            byte[] encData_byte = new byte[password.Length];
            encData_byte = System.Text.Encoding.UTF8.GetBytes(password);
            string encodedData = Convert.ToBase64String(encData_byte);
            return encodedData;
         }
         catch (Exception ex)
         {
            throw new Exception("Error in base64Encode" + ex.Message);
         }
      }

      public void viewstates()
      {
         obj.cmd.Parameters.Clear();
         DataTable dt = new DataTable();
         dt = obj.gettable("getState");
         DropDownList1.DataSource = dt;
         DropDownList1.DataTextField = "sname";
         DropDownList1.DataValueField = "sid";
         DropDownList1.DataBind();
         DropDownList1.Items.Insert(0, "---Select---");
```

```
    }

    public void viewDistrict()
    {
        if (DropDownList2.SelectedIndex != 0)
        {



            obj.cmd.Parameters.Clear();
            DataTable dt = new DataTable();
            obj.cmd.Parameters.AddWithValue("@stateid",
Convert.ToInt32(DropDownList1.SelectedItem.Value));
            dt = obj.gettable("getDistrict");
            DropDownList2.DataSource = dt;
            DropDownList2.DataTextField = "dname";
            DropDownList2.DataValueField = "distid";
            DropDownList2.DataBind();
            DropDownList2.Items.Insert(0, "---Select---");
        }
        else
        {
            DropDownList2.Items.Clear();
            DropDownList2.Items.Insert(0, "---Select---");
        }

    }


    protected void btn_save_Click(object sender, EventArgs e)
    {



        string s = " ";
        obj.cmd.Parameters.Clear();

        obj.cmd.Parameters.AddWithValue("@fname", txtfname.Text);
        obj.cmd.Parameters.AddWithValue("@lname", txtlname.Text)
        obj.cmd.Parameters.AddWithValue("@gender", rdogender.Text);
        obj.cmd.Parameters.AddWithValue("@dob", txtdob.Text);
        obj.cmd.Parameters.AddWithValue("@hname", txthname.Text);
        obj.cmd.Parameters.AddWithValue("@city", txtcity.Text);
        obj.cmd.Parameters.AddWithValue("@phno", txtphn.Text);
        obj.cmd.Parameters.AddWithValue("@uname", txtmail.Text);
        obj.cmd.Parameters.AddWithValue("@pwd",
EncodePasswordToBase64(txtpwd.Text));
```

```
            obj.cmd.Parameters.AddWithValue("@sid",
Convert.ToInt32(DropDownList1.SelectedItem.Value));
            obj.cmd.Parameters.AddWithValue("@distid",
Convert.ToInt32(DropDownList2.SelectedItem.Value));
            s = obj.getvalue("patients_insertion1").ToString();

            ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + s +
"');", true);
            btn_save.Enabled = false;

                        MailMessage MyMailMessage = new
MailMessage("shijokjacob@gmail.com", txtmail.Text, "subject ", "Hi,This is Your
Login Details.Username: '" + txtmail.Text + "'and Password:'" + txtpwd.Text + "'
Login Here http://localhost:61461/login");
            MyMailMessage.IsBodyHtml = false;
            NetworkCredential mailAuthentication = new
NetworkCredential("shijokjacob@gmail.com", "9656970539");
            SmtpClient mailClient = new SmtpClient("smtp.gmail.com", 587);
            mailClient.EnableSsl = true;
            mailClient.UseDefaultCredentials = false;
            mailClient.Credentials = mailAuthentication;
            mailClient.Send(MyMailMessage);
            Response.Write("E-mail sent!");
                    }

        protected void DropDownList3_SelectedIndexChanged(object sender,
EventArgs e)
        {

        }

        protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
        {
            viewDistrict();
        }




        protected void txtmail_TextChanged(object sender, EventArgs e)
        {
            string uName = txtmail.Text;
            obj.cmd.Parameters.AddWithValue("@uName", uName);
            string s = obj.getvalue("pro_slctuname").ToString();
            if (s == "1")
            {
                lbl_available.ForeColor = System.Drawing.Color.Red;
                lbl_available.Text = "username exists";
                btn_save.Enabled = false;

            }
            else
            {
                lbl_available.ForeColor = System.Drawing.Color.Green;
                lbl_available.Text = "username available";
                btn_save.Enabled = true;
                if (lbl_available.Text == "username available")
                {
                    btn_save.Visible = true;
                }
                else
                {
                    btn_save.Visible = false;
                }
            }

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            Response.Redirect("login.aspx");
        }

        protected void btn_cancel_Click(object sender, EventArgs e)
        {
            txtfname.Text = "";
            txtlname.Text = "";
            rdogender.Text = "";
            txtdob.Text = "";
            txthname.Text = "";
            txtcity.Text = "";
            txtmail.Text = "";
            txtpwd.Text = "";
            txtphn.Text = "";
        }

    }
}
```

Stored procedure

```
1.CREATE PROCEDURE [dbo].[patients_insertion1]
            @fname varchar(50),
            @lname varchar(50),
            @gender varchar(50),
            @dob varchar(50),
            @sid int,
            @distid int,
            @hname varchar(50),
            @city varchar(50),
            @phno varchar(50),
            @uname varchar(50),
            @pwd varchar(50)
AS
BEGIN
declare @msg varchar(50),
@cnt int
            SET NOCOUNT ON;


            insert into hms_tbl_login(uname,pwd,type) values (@uname,@pwd,2)
            select @cnt=max(lid) from hms_tbl_login;
            insert into
hms_tbl_registration(fname,lname,gender,dob,address,city,email,phone,lid,sid,distid)
values(@fname,@lname,@gender,@dob,@hname,@city,@uname,@phno,@cnt,@si
d,@distid)
            set @msg='Details Inserted Succesfully'
            select @msg;
            end
```

```
2. CREATE PROCEDURE [dbo].[pro_slctuname]
@uName varchar(100)
as
begin
declare @cnt int
set nocount on;
select @cnt=COUNT(*)from hms_tbl_login where uname=@uName and status=1
end
select @cnt
```

3.Adding new staff

***@since File available since Release 1.0**

***@created on 09-09-2016**

***@last edited on oct-10-2016**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Security.Cryptography;
using System.Web.Security;
using System.IO;
using System.Net.Mail;
using System.Net;
using myhospital;

namespace MyHospital.Area.Admin
{
   public partial class staffmgmnt : System.Web.UI.Page
   {
      Class1 obj = new Class1();
      protected void Page_Load(object sender, EventArgs e)
      {
         var vsession = Session["id"];
         if (vsession == null)
         {
            Response.Redirect("/login.aspx");
         }
         if (!IsPostBack)
         {
            viewdept();
            //viewDistrict();
            //txtuname.Text = "";
            //txtpwd.Text = "";

         }
      }
      public static string EncodePasswordToBase64(string password)
      {
         try
         {
            byte[] encData_byte = new byte[password.Length];
            encData_byte = System.Text.Encoding.UTF8.GetBytes(password);
            string encodedData = Convert.ToBase64String(encData_byte);
            return encodedData;
         }
         catch (Exception ex)
         {
            throw new Exception("Error in base64Encode" + ex.Message);
         }
      }
```

```
    public void viewdept()
    {
        obj.cmd.Parameters.Clear();
        DataTable dt = new DataTable();
        dt = obj.gettable("getdept");
        DropDownList1.DataSource = dt;
        DropDownList1.DataTextField = "deptname";
        DropDownList1.DataValueField = "deptid";
        DropDownList1.DataBind();
        DropDownList1.Items.Insert(0, "---Select---");


    }
    public void viewward()
    {
        if (DropDownList2.SelectedIndex != 0)
        {



            obj.cmd.Parameters.Clear();
            DataTable dt = new DataTable();
            obj.cmd.Parameters.AddWithValue("@deptid",
Convert.ToInt32(DropDownList1.SelectedItem.Value));
            dt = obj.gettable("getward");
            DropDownList2.DataSource = dt;
            DropDownList2.DataTextField = "wname";
            DropDownList2.DataValueField = "wid";
            DropDownList2.DataBind();
            DropDownList2.Items.Insert(0, "---Select---");
        }
        else
        {
            DropDownList2.Items.Clear();
            DropDownList2.Items.Insert(0, "---Select---");
        }

    }

    protected void DropDownList1_SelectedIndexChanged(object sender,
EventArgs e)
    {
        viewward();
    }

    protected void btnsubmit_Click1(object sender, EventArgs e)
    {
            string s = " ";
            obj.cmd.Parameters.Clear();
            obj.cmd.Parameters.AddWithValue("@fname", txtfname.Text);
            obj.cmd.Parameters.AddWithValue("@lname", txtlname.Text);
            obj.cmd.Parameters.AddWithValue("@gender", rbgender.Text);
            obj.cmd.Parameters.AddWithValue("@dob", txtdob.Text);
            obj.cmd.Parameters.AddWithValue("@phno", txtphn.Text);
            obj.cmd.Parameters.AddWithValue("@uname", txtmail.Text);
            obj.cmd.Parameters.AddWithValue("@pwd",
EncodePasswordToBase64(txtpwd.Text));
            obj.cmd.Parameters.AddWithValue("@stype",
Convert.ToInt32(DropDownList3.SelectedItem.Value));
            obj.cmd.Parameters.AddWithValue("@deptid",
Convert.ToInt32(DropDownList1.SelectedItem.Value));
            obj.cmd.Parameters.AddWithValue("@wid",
Convert.ToInt32(DropDownList2.SelectedItem.Value));
            s = obj.getvalue("staffinsert").ToString();
            ClientScript.RegisterStartupScript(this.GetType(), "myalert", "alert('" + s +
"');", true);

 MailMessage MyMailMessage = new MailMessage("shijokjacob@gmail.com",
txtmail.Text, "subject ", "Hi,This is Your Login Details.Username: '" + txtmail.Text +
"'and Password:'" + txtpwd.Text + "' Login Here http://localhost:61461/login");
            MyMailMessage.IsBodyHtml = false;
            NetworkCredential mailAuthentication = new
NetworkCredential("shijokjacob@gmail.com", "9656970539");
            SmtpClient mailClient = new SmtpClient("smtp.gmail.com", 587);
            mailClient.EnableSsl = true;
            mailClient.UseDefaultCredentials = false;
            mailClient.Credentials = mailAuthentication;
            mailClient.Send(MyMailMessage);
            Response.Write("E-mail sent!");
        }

        protected void txtmail_TextChanged(object sender, EventArgs e)
        {
            string uName = txtmail.Text;
            obj.cmd.Parameters.AddWithValue("@uName", uName);
            string s = obj.getvalue("pro_slctuname").ToString();
            if (s == "1")
            {
                lbl_available.ForeColor = System.Drawing.Color.Red;
                lbl_available.Text = "username exists";
                btnsubmit.Enabled = false;

            }

            else
            {
```

```
            lbl_available.ForeColor = System.Drawing.Color.Green;
            lbl_available.Text = "username available";
            btnsubmit.Enabled = true;
            if (lbl_available.Text == "username available")
            {
                btnsubmit.Visible = true;
            }
            else
            {
                btnsubmit.Visible = false;
            }
        }
    }



    protected void btn_cancel_Click(object sender, EventArgs e)
    {
        Response.Redirect("staffmgmnt.aspx");
    }



  }
}
```

Stored procedure

1. CREATE PROCEDURE [dbo].[staffinsert]

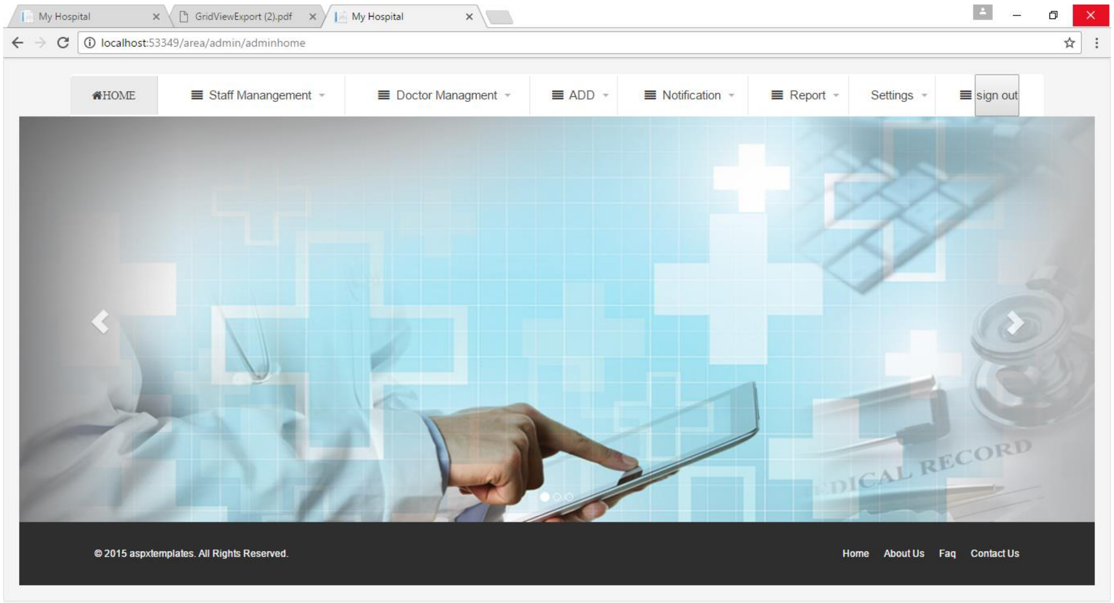
```
        @fname varchar(50),
        @lname varchar(50),
        @stype varchar(50),
        @dob varchar(50),
        @gender varchar(50),
        @uname varchar(50),
        @pwd varchar(50),
        @phno varchar(50),
        @deptid int,
        @wid int
AS
BEGIN
        declare @msg varchar(50),
        @cnt int,@cnt1 int
        SET NOCOUNT ON;
        select @cnt=COUNT(email) from hms_tbl_staff where email=@uname and
status=1
```

```
        select @cnt1=COUNT(stype) from hms_tbl_staff where stype=5 and
wid=@wid and deptid=@deptid and status=1
        if @cnt=0 and @cnt1=0
        begin
        insert into hms_tbl_login(uname,pwd,type,status) values
(@uname,@pwd,@stype,1)
        select @cnt=max(lid) from hms_tbl_login;
        insert into
hms_tbl_staff(fname,lname,stype,gender,dob,email,phone,deptid,wid,lid,status)values
(@fname,@lname,@stype,@gender,@dob,@uname,@phno,@deptid,@wid,@cnt,1)
        set @msg='inserted succefully'
        end
        else
        begin
        set @msg='Staff already exist'
        end
        SELECT @msg
END
```

```
2. CREATE PROCEDURE [dbo].[pro_slctuname]
@uName varchar(100)
as
begin
declare @cnt int
set nocount on;
select @cnt=COUNT(*)from hms_tbl_login where uname=@uName and status=1
end
select @cnt
```

9.2 Screen Shots

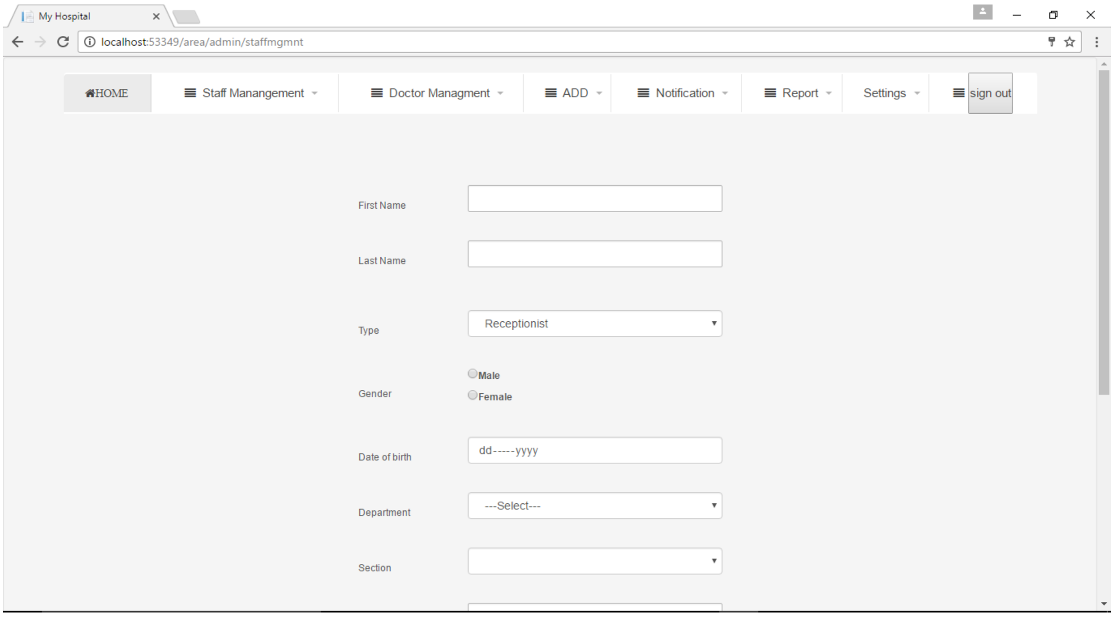
- Admin Home



© 2015 aspxtemplates. All Rights Reserved.

Figure 19. Admin home

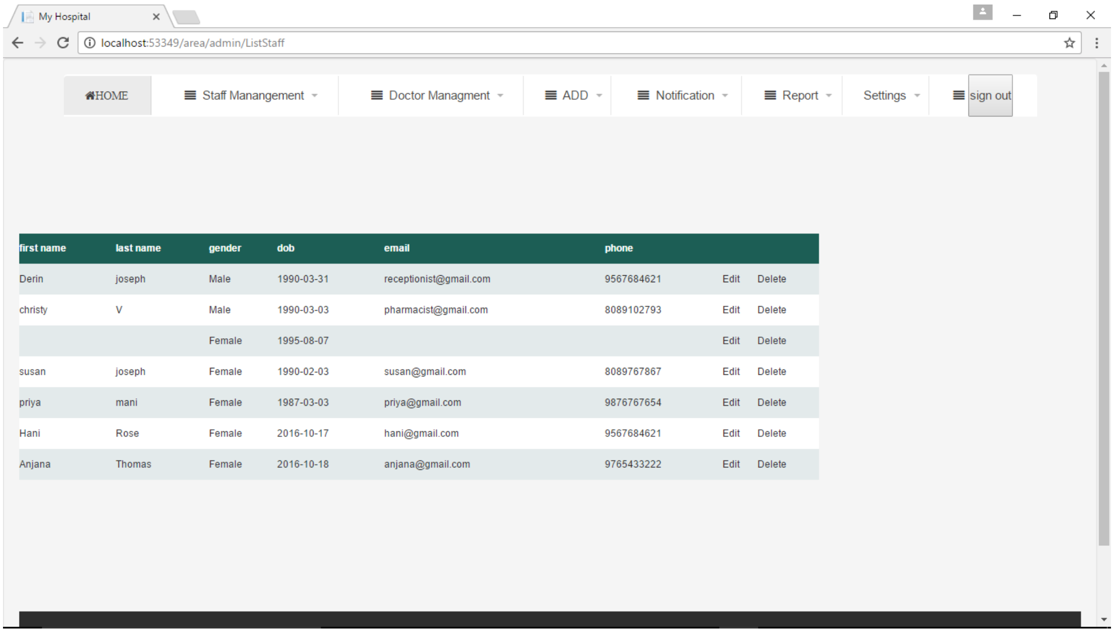
- Add Staff



First Name	
Last Name	
Type	Receptionist ▾
Gender	○ Male ○ Female
Date of birth	dd-----yyyy
Department	---Select--- ▾
Section	▾

Figure 20. Add staff

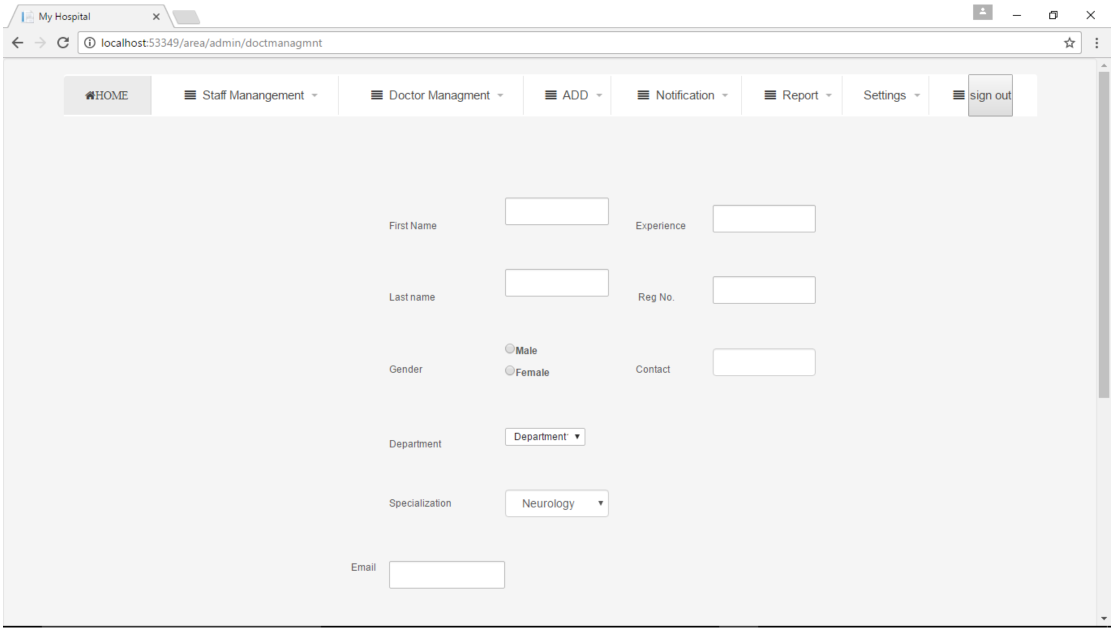
- View and edit staff



first name	last name	gender	dob	email	phone		
Derin	joseph	Male	1990-03-31	receptionist@gmail.com	9567684621	Edit	Delete
christy	V	Male	1990-03-03	pharmacist@gmail.com	8089102793	Edit	Delete
		Female	1995-08-07			Edit	Delete
susan	joseph	Female	1990-02-03	susan@gmail.com	8089767867	Edit	Delete
priya	mani	Female	1987-03-03	priya@gmail.com	9876767654	Edit	Delete
Hani	Rose	Female	2016-10-17	hani@gmail.com	9567684621	Edit	Delete
Anjana	Thomas	Female	2016-10-18	anjana@gmail.com	9765433222	Edit	Delete

Figure 21.View and edit staff

- Add Doctor



First Name

Experience

Last name

Reg No.

Gender ○Male ○Female

Contact

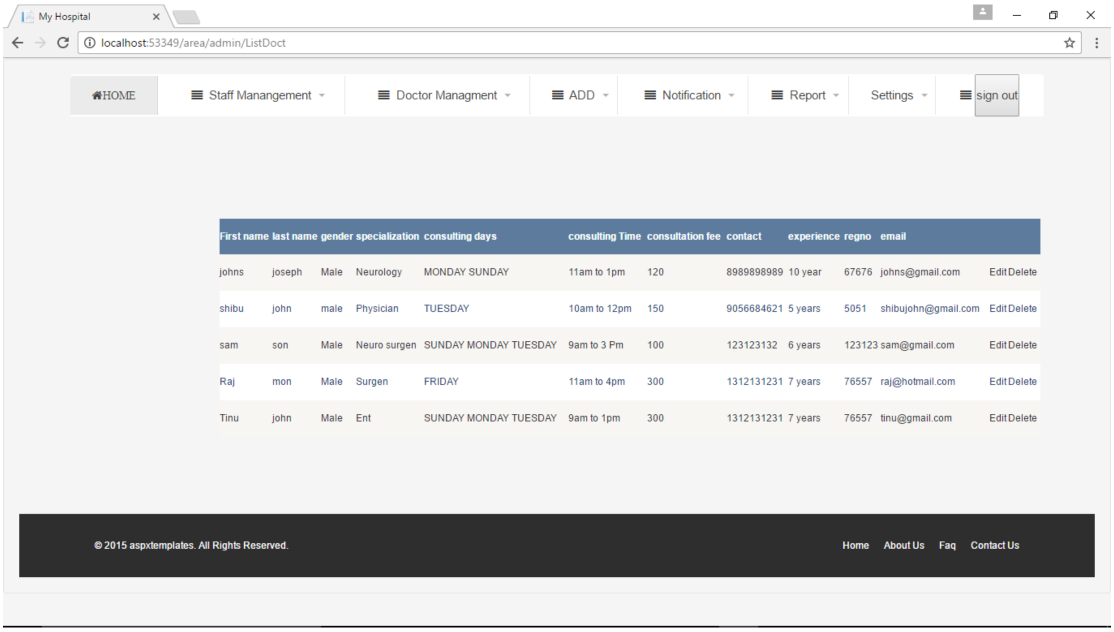
Department Department ▾

Specialization Neurology ▾

Email

Figure 22.Add Doctor

- View and edit Doctor



First name	last name	gender	specialization	consulting days	consulting Time	consultation fee	contact	experience	regno	email	
johns	joseph	Male	Neurology	MONDAY SUNDAY	11am to 1pm	120	8989898989	10 year	67676	johns@gmail.com	Edit Delete
shibu	john	male	Physician	TUESDAY	10am to 12pm	150	9056684621	5 years	5051	shibujohn@gmail.com	Edit Delete
sam	son	Male	Neuro surgen	SUNDAY MONDAY TUESDAY	9am to 3 Pm	100	123123132	6 years	123123	sam@gmail.com	Edit Delete
Raj	mon	Male	Surgen	FRIDAY	11am to 4pm	300	1312131231	7 years	76557	raj@hotmail.com	Edit Delete
Tinu	john	Male	Ent	SUNDAY MONDAY TUESDAY	9am to 1pm	300	1312131231	7 years	76557	tinu@gmail.com	Edit Delete

© 2015 aspxtemplates. All Rights Reserved.

Figure 23. View and edit Doctor

- Add Department, wards, rooms

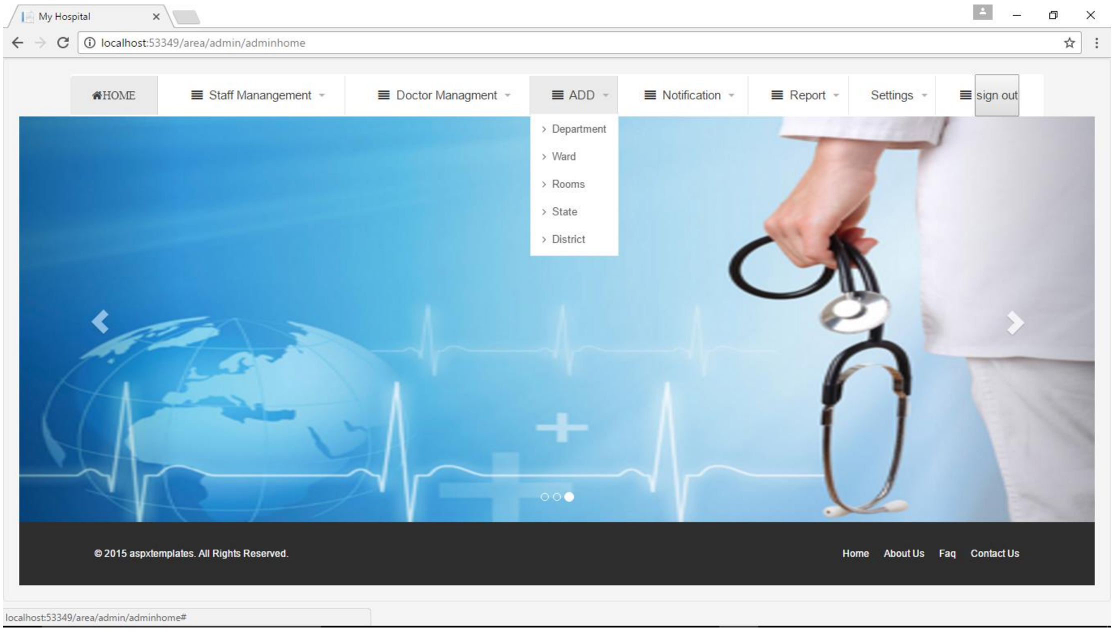


Figure 24. Add department

- New user notification for Approval/Rejection

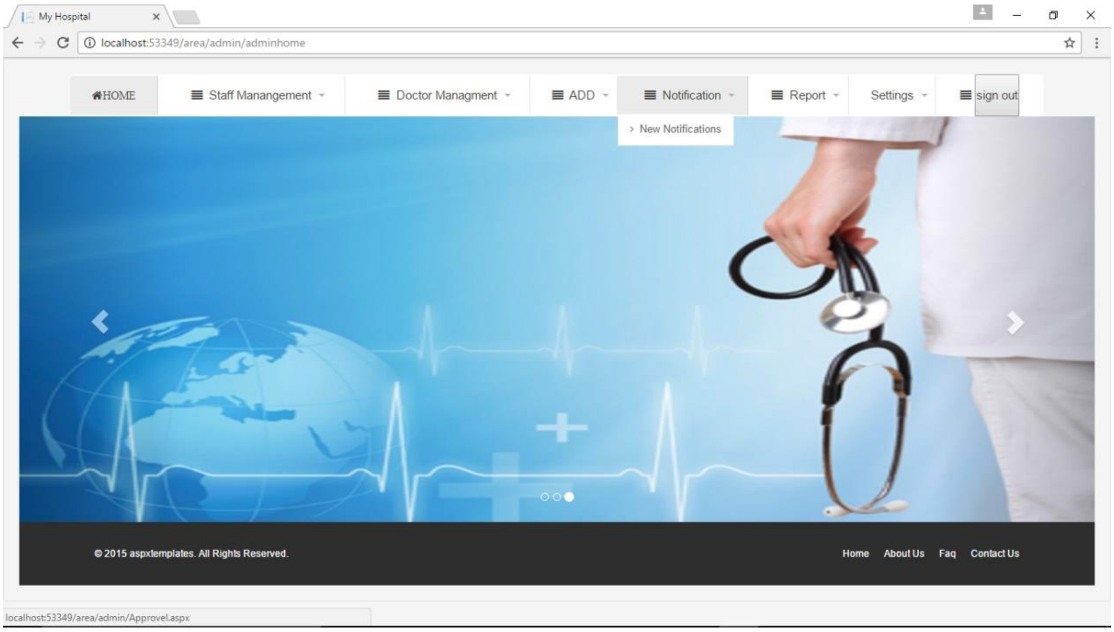
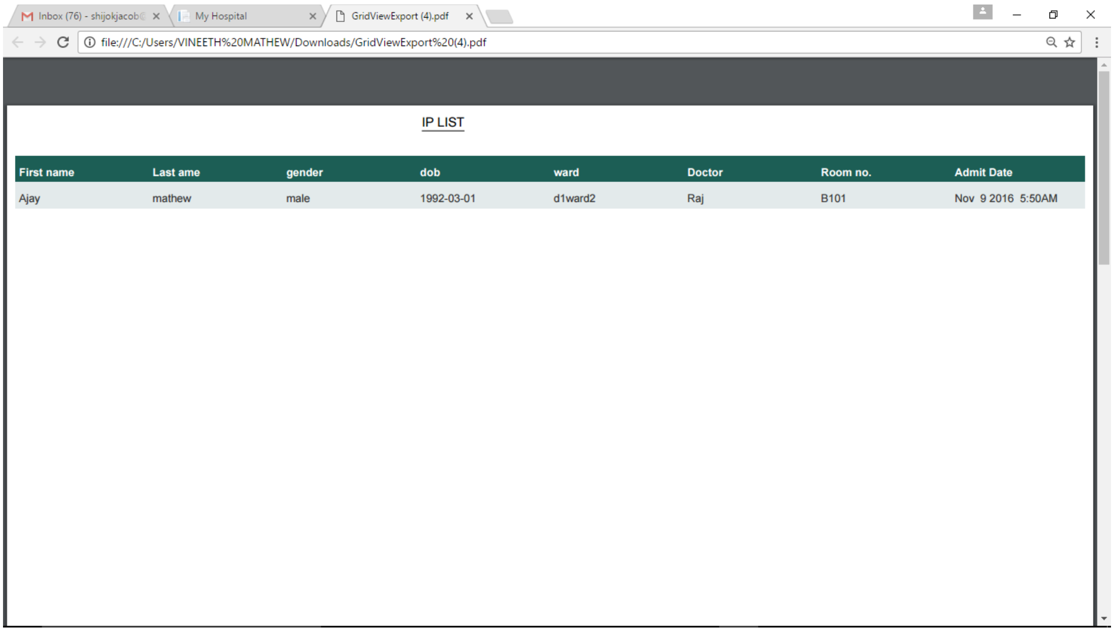


Figure 25. Approval/Rejection

- Report generation

Figure 26.Ip report

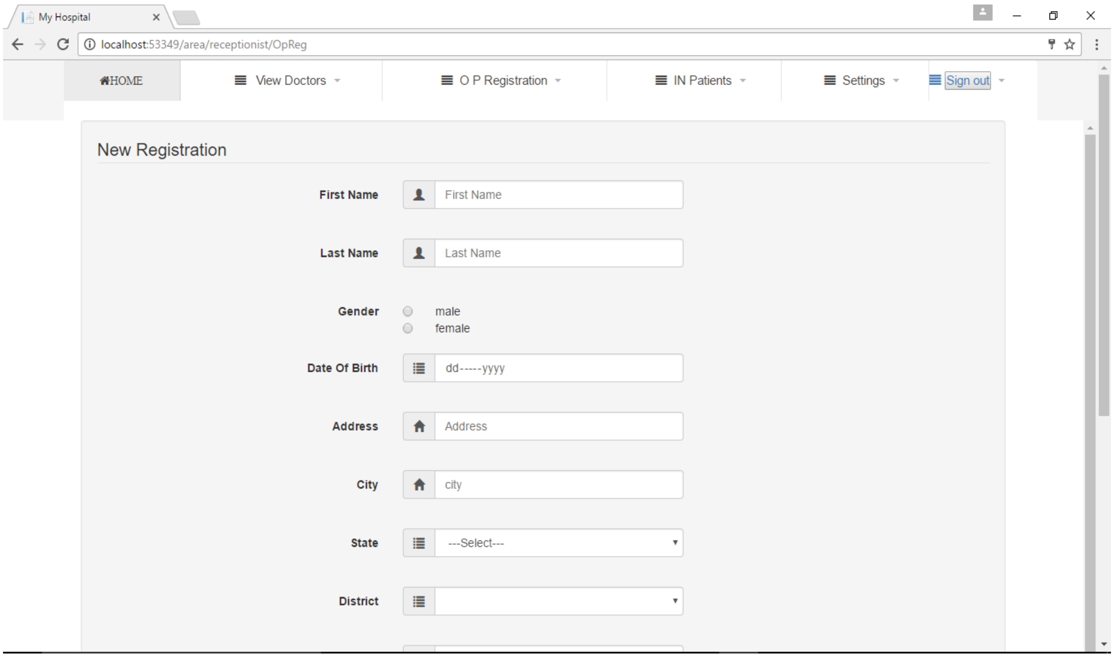


Inbox (76) - shijokjacob@
My Hospital
GridViewExport (4).pdf
file:///C:/Users/VINEETH%20MATHEW/Downloads/GridViewExport%20(4).pdf

IP LIST

First name	Last ame	gender	dob	ward	Doctor	Room no.	Admit Date
Ajay	mathew	male	1992-03-01	d1ward2	Raj	B101	Nov 9 2016 5:50AM

Receptionist- OP Registration



New Registration

First Name: First Name

Last Name: Last Name

Gender: ○ male ○ female

Date Of Birth: dd-----yyyy

Address: Address

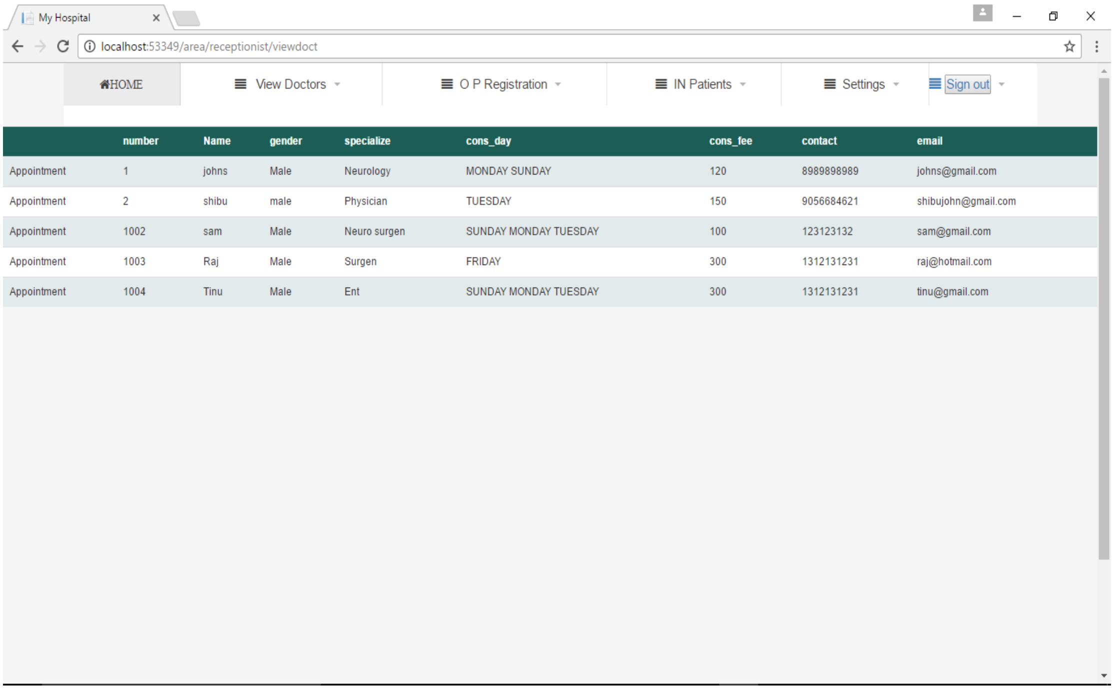
City: city

State: ---Select---

District:

Figure 27. Op registration

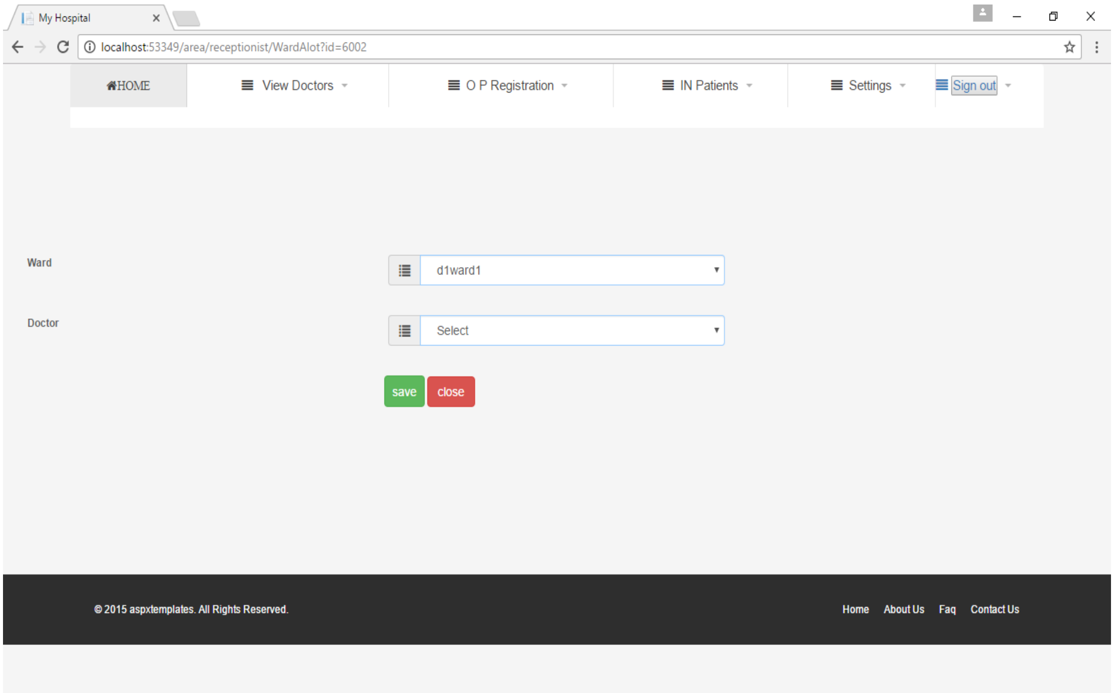
- View Doctors



	number	Name	gender	specialize	cons_day	cons_fee	contact	email
Appointment	1	johns	Male	Neurology	MONDAY SUNDAY	120	8989898989	johns@gmail.com
Appointment	2	shibu	male	Physician	TUESDAY	150	9056684621	shibujohn@gmail.com
Appointment	1002	sam	Male	Neuro surgen	SUNDAY MONDAY TUESDAY	100	123123132	sam@gmail.com
Appointment	1003	Raj	Male	Surgen	FRIDAY	300	1312131231	raj@hotmail.com
Appointment	1004	Tinu	Male	Ent	SUNDAY MONDAY TUESDAY	300	1312131231	tinu@gmail.com

Figure 28. View doctors

- Ward Allocation



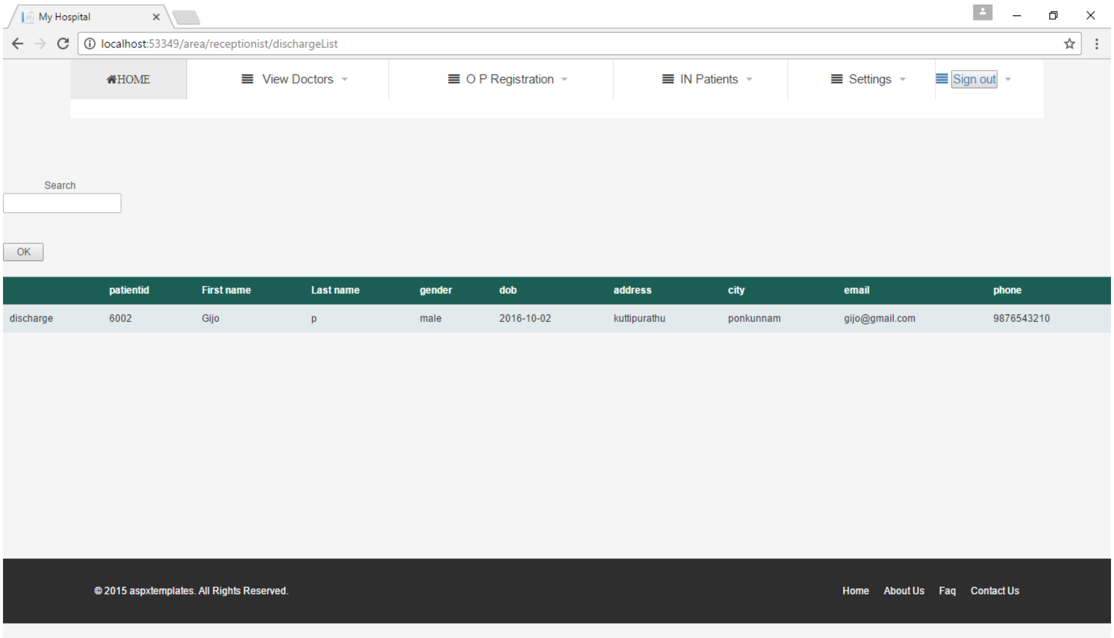
localhost:53349/area/receptionist/WardAlot?id=6002

Ward d1ward1

Doctor Select

Figure 29.Ward allocation

- Search Inpatients



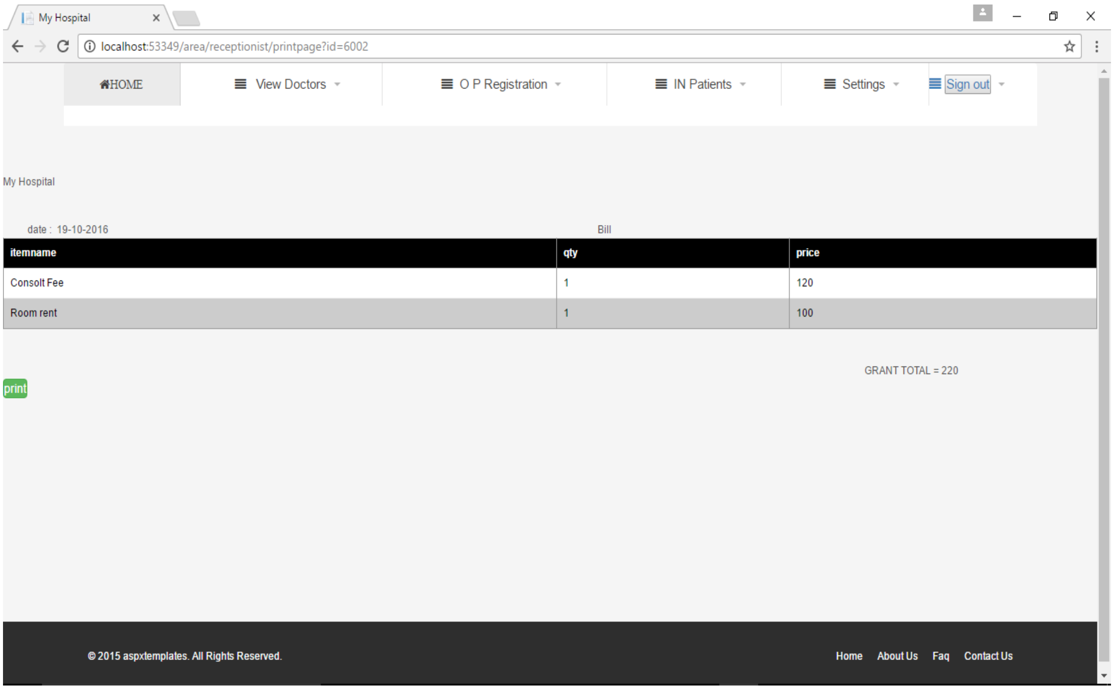
localhost:53349/area/receptionist/dischargeList

Search

	patientid	First name	Last name	gender	dob	address	city	email	phone
discharge	6002	Gijo	p	male	2016-10-02	kuttipurathu	ponkunnam	gijo@gmail.com	9876543210

Figure 30. Search patients

- Discharge card



My Hospital

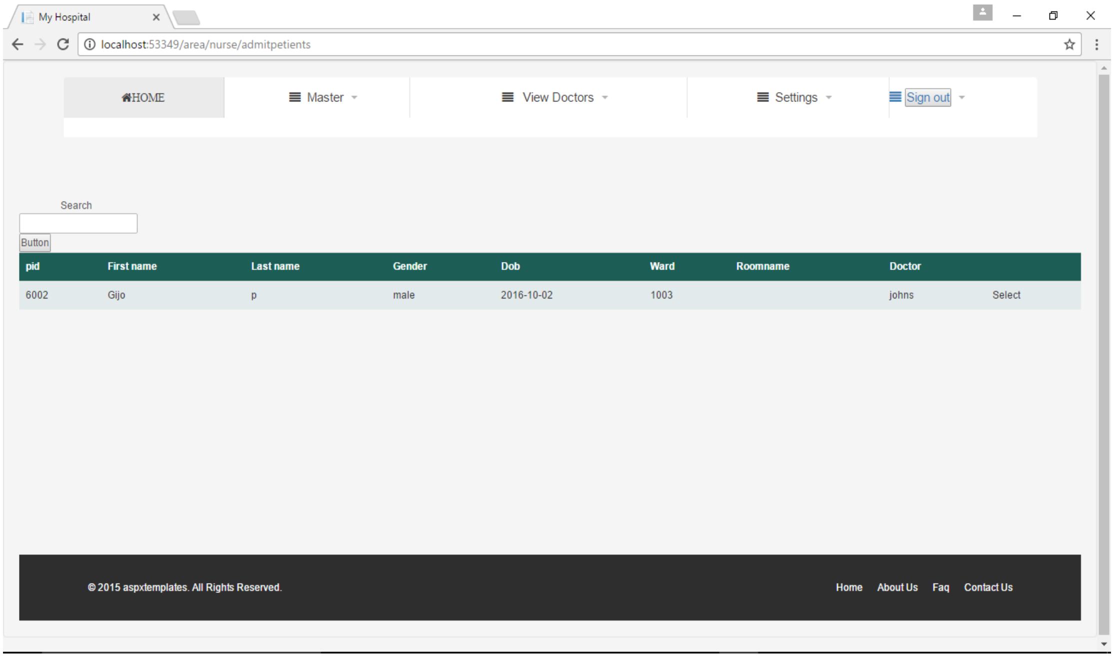
date : 19-10-2016 Bill

itemname	qty	price
Consolt Fee	1	120
Room rent	1	100

GRANT TOTAL = 220

Figure 31.Discharge card

- Nurse – Notification



localhost:53349/area/nurse/admitpetients

Search

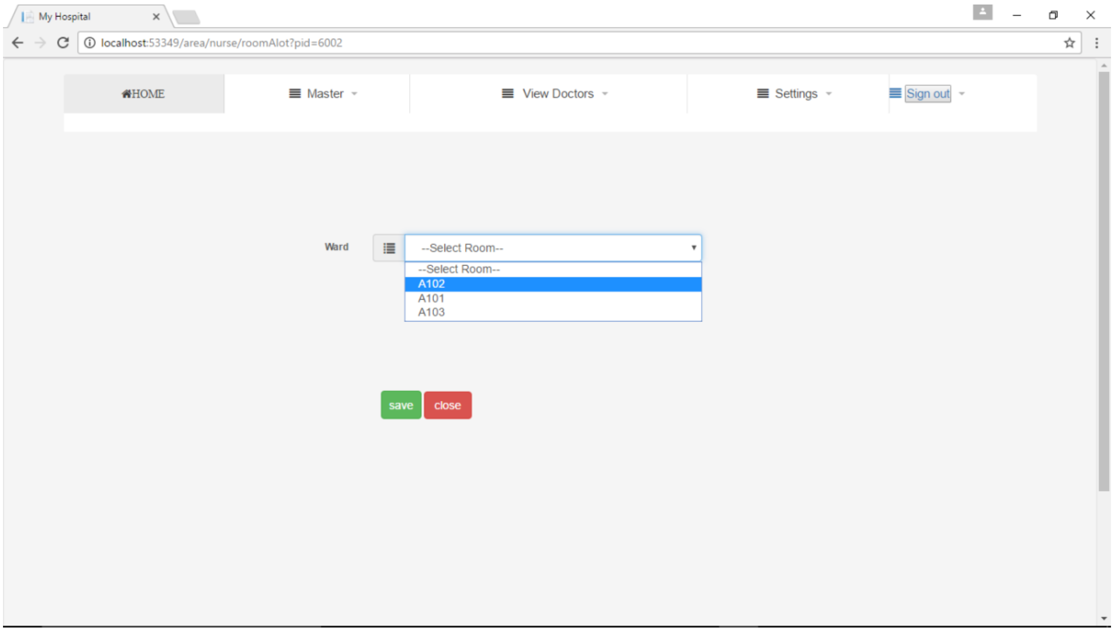
Button

pid	First name	Last name	Gender	Dob	Ward	Roomname	Doctor	
6002	Gijo	p	male	2016-10-02	1003		johns	Select

© 2015 aspxtemplates. All Rights Reserved.

Figure 32. Notification of patients refered to ward

- Room Allocation



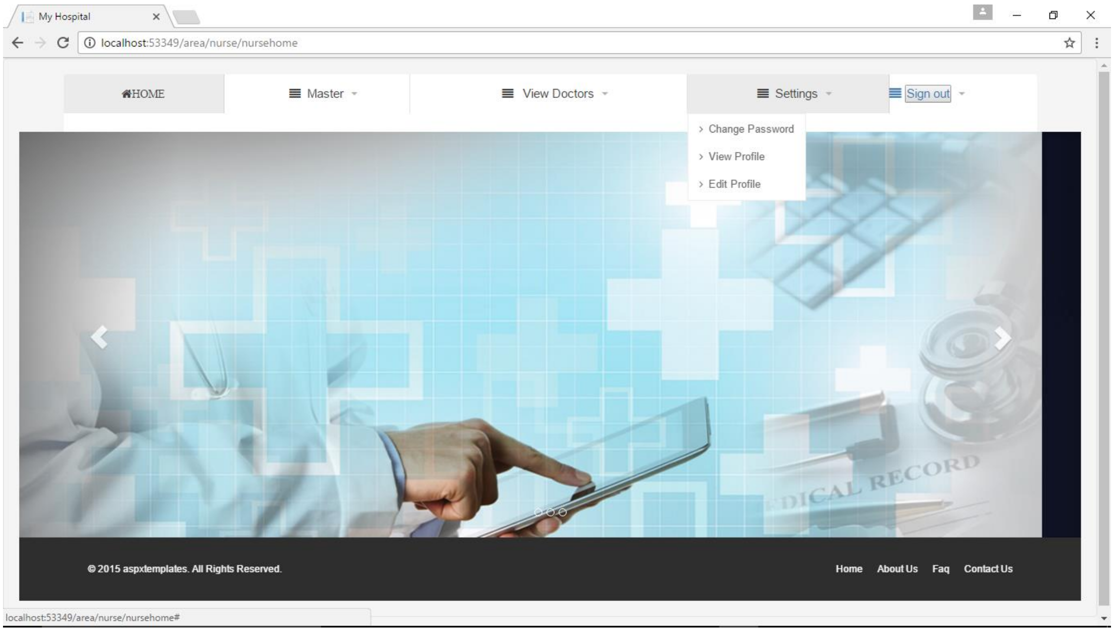
localhost:53349/area/nurse/roomAlot?pid=6002

Ward: --Select Room--

--Select Room--
A102
A101
A103

Figure 33. Room allocation

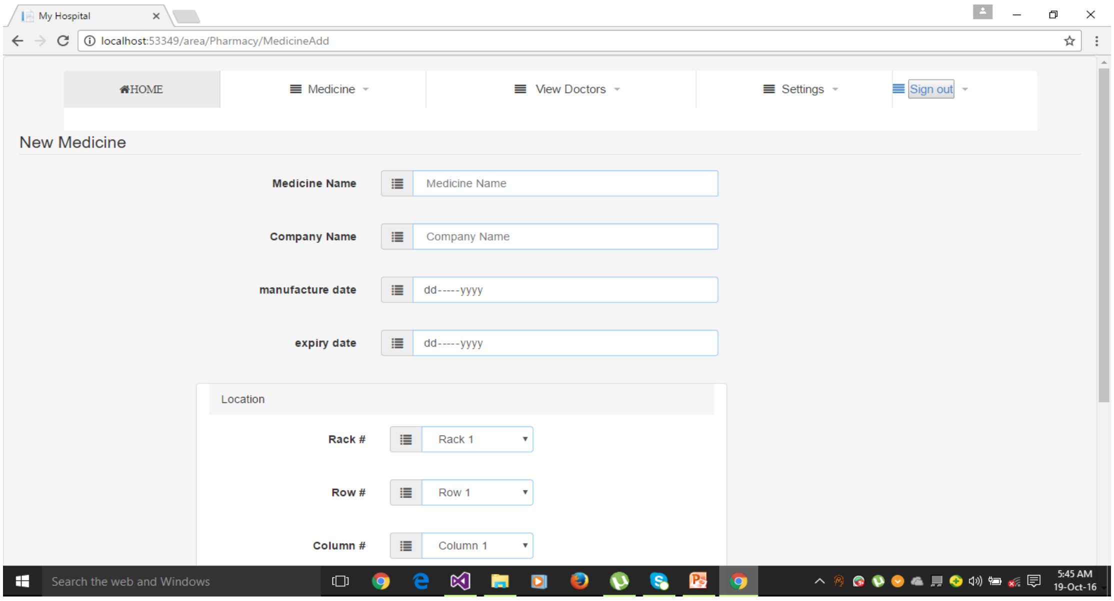
- Profile Settings



© 2015 aspxtemplates. All Rights Reserved.

Figure 34. Profile settings

- Pharmacist – Add medicine


My Hospital
localhost:53349/area/Pharmacy/MedicineAdd
HOME
Medicine
View Doctors
Settings
Sign out
Search the web and Windows
5:45 AM
19-Oct-16

New Medicine

Medicine Name: Medicine Name

Company Name: Company Name

manufacture date: dd-----yyyy

expiry date: dd-----yyyy

Location

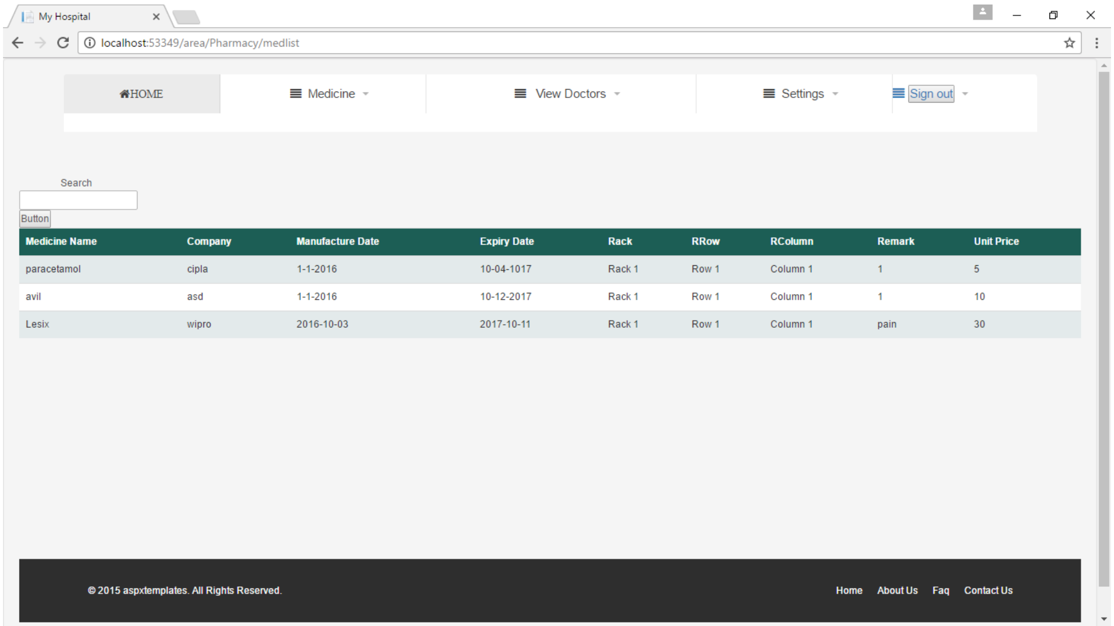
Rack #: Rack 1

Row #: Row 1

Column #: Column 1

Figure 35.Add medicine

- Search medicine



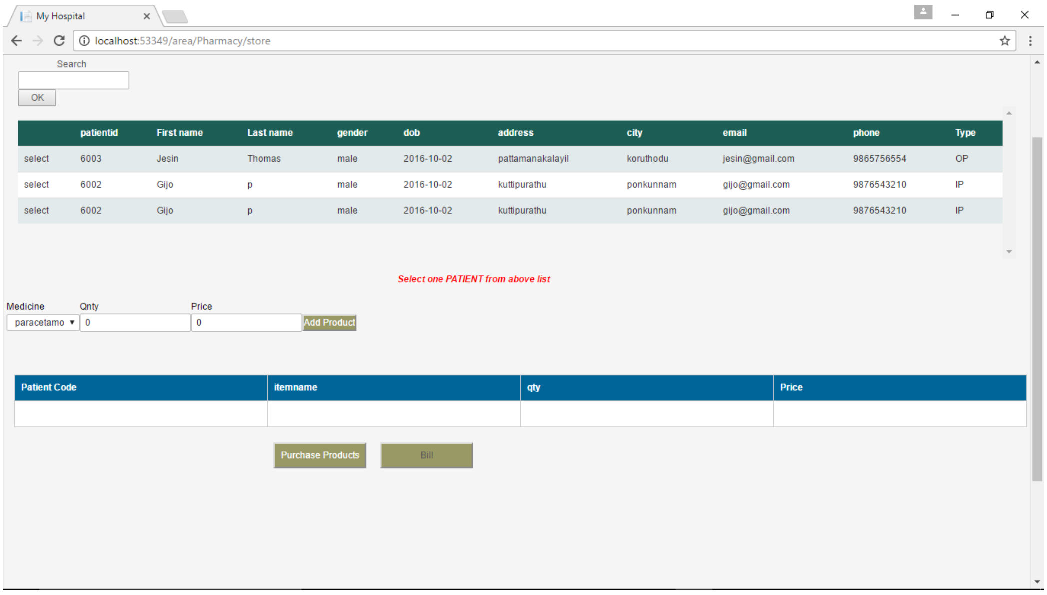
Search

Medicine Name	Company	Manufacture Date	Expiry Date	Rack	RRow	RColumn	Remark	Unit Price
paracetamol	cipla	1-1-2016	10-04-1017	Rack 1	Row 1	Column 1	1	5
avil	asd	1-1-2016	10-12-2017	Rack 1	Row 1	Column 1	1	10
Lesix	wipro	2016-10-03	2017-10-11	Rack 1	Row 1	Column 1	pain	30

© 2015 aspxtemplates. All Rights Reserved.

Figure 36. Search medicine

- Billing



Search

	patientid	First name	Last name	gender	dob	address	city	email	phone	Type
select	6003	Jesin	Thomas	male	2016-10-02	pattamanakalayil	koruthodu	jesin@gmail.com	9865756554	OP
select	6002	Gijo	p	male	2016-10-02	kuttipurathu	ponkunnam	gijo@gmail.com	9876543210	IP
select	6002	Gijo	p	male	2016-10-02	kuttipurathu	ponkunnam	gijo@gmail.com	9876543210	IP

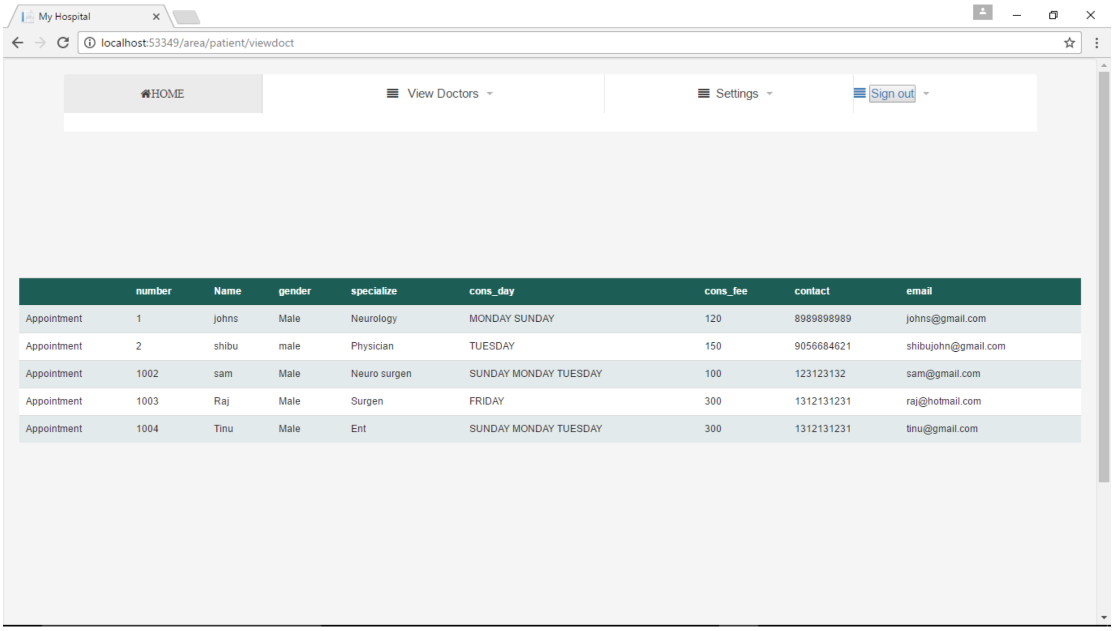
Select one PATIENT from above list

Medicine: paracetamo ▾ Qnty: 0 Price: 0

Patient Code	itemname	qty	Price

Figure 37.Billing

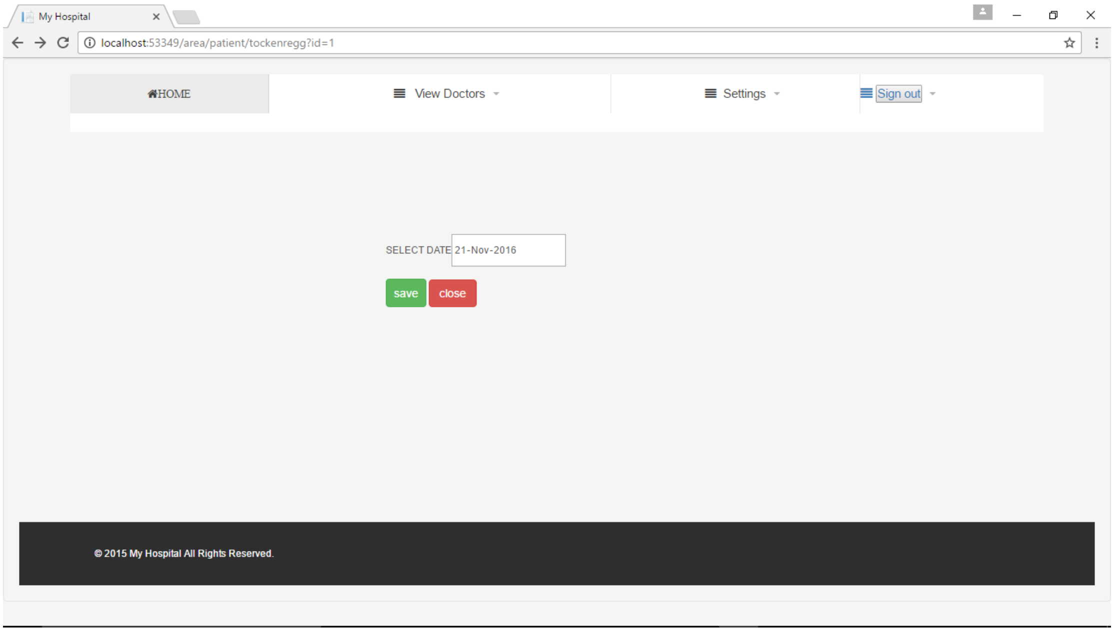
- Patient – View doctor details



	number	Name	gender	specialize	cons_day	cons_fee	contact	email
Appointment	1	johns	Male	Neurology	MONDAY SUNDAY	120	8989898989	johns@gmail.com
Appointment	2	shibu	male	Physician	TUESDAY	150	9056684621	shibujohn@gmail.com
Appointment	1002	sam	Male	Neuro surgen	SUNDAY MONDAY TUESDAY	100	123123132	sam@gmail.com
Appointment	1003	Raj	Male	Surgen	FRIDAY	300	1312131231	raj@hotmail.com
Appointment	1004	Tinu	Male	Ent	SUNDAY MONDAY TUESDAY	300	1312131231	tinu@gmail.com

Figure 38. View doctor details

- Appointment booking



SELECT DATE 21-Nov-2016

© 2015 My Hospital All Rights Reserved.

Figure 39. Appointment booking